Post-hoc calibration without distributional assumptions

Chirag Gupta

August 2023

CMU-ML-23-107

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Aaditya Ramdas (CMU), Chair
Dean Foster (Amazon)
Geoff Gordon (CMU)
Vianney Perchet (CREST, ENSAE)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was partially supported by the Bloomberg Data Science Ph.D. Fellowship and by the National Science Foundation grant number 1713003. All large-scale computations were performed at the Pittsburgh Supercomputing Cluster, on the Bridges and Bridges-2 systems, supported by the XSEDE and ACCESS programs (National Science Foundation grant numbers 1548562, 2138259, 2138286, 2138307, 2137603, 2138296, 1928147).

The views and conclusions contained in this document are those of the author only and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government, or any other entity.

Keywords: machine learning, statistics, forecasting, uncertainty quantification, reliable machine learning, calibration, post-hoc, distribution-free, binning, scaling, online learning, sequential, distribution shift, predictive inference, conformal prediction

To my teachers,
who have shaped me in countless, unseen ways,
and to my parents—my first teachers.

Abstract

Machine learning classifiers typically provide scores for the different classes. These scores are supplementary to class predictions and may be crucial for downstream decision-making. However, can they be interpreted as probabilities? Scores produced by a calibrated classifier satisfy such a probabilistic property, informally described as follows. For binary classification with labels 0 and 1, a classifier is calibrated if on the instances where it predicts a score s (in [0,1]), the probability of the true label being 1 equals s.

The primary goal of this thesis is to demonstrate that a miscalibrated classifier can be provably "post-hoc" calibrated using a small set of held-out datapoints, such as a validation dataset. Such calibration can be achieved in two different senses: (a) model calibration of a given classifier for a fixed data-generating distribution; and (b) forecast calibration of a sequence of probabilistic forecasts for an online data stream. These two views have been studied by two largely independent bodies of literature; we draw from and contribute to both. In particular, we derive the first calibration method that uses both model and forecast calibration techniques.

The algorithms we develop come with theoretical guarantees that hold under mild or no assumptions. A majority of our work is in the "distribution-free" setting, where we assume that the data is i.i.d., but make no parametric or smoothness assumptions on the data-generating distribution. We show that using discretized or binned scores is necessary and sufficient to achieve distribution-free calibration (Chapters 3–5). The culminating work of this thesis goes beyond distribution-free by altogether dispensing with the requirement that data is being generated from a distribution. We show that even if the data is "adversarial", calibration can be provably achieved in a practically meaningful manner (Chapters 6 and 7).

Acknowledgments

As I navigated the world of statistical machine learning, I was guided by my advisor, Aaditya Ramdas (Aadi). Aadi taught me not only the tools and techniques of the trade but also the art and discipline of it. My PhD journey, as that of all others, faced many internal and external hurdles. Aadi has been forward-looking and encouraging throughout. I express my sincerest gratitude to Aadi, and to the rest of my thesis committee—Dean Foster, Geoff Gordon, and Vianney Perchet—for their insightful suggestions, constructive feedback, and appreciation of my work.

I thank my coauthors during the PhD—Sivaraman Balakrishnan, Arun Kumar Kuchibhotla, Aleksandr (Sasha) Podkopaev, Aaron Rumack, and Youngseog Chung—for sharing their knowledge, time, and enthuasiasm with me. In 2021, I received the Bloomberg fellowship, which was one of the high points of my PhD. I thank my collaborators at Bloomberg—David Rosenberg and Philipp Meerkamp—for their patience and grit while we tackled a challenging problem. Before starting my PhD, I learnt machine learning from Prateek Jain, Praneeth Netrapalli, Vinay Namboodiri, and Purushottam Kar. Before that, I learnt math from M. Prakash and Bhaskaracharya Pratishthana, in Pune, India. My teachers at Delhi Public School, Pune, were the source of my dreams and ambitions.

None of this would have been possible, of course, without the institution of the Machine Learning Department at Carnegie Mellon University. The department gave me access to motivated and supportive colleagues, a beautiful office (Gates Hillman Center 8011), seminars and reading groups (particularly the Stat-ML Reading Group), and ample social events that lead to long-lasting friendships.

I've had the fortune of finding incredible friends who provided the affirmation and fun times that kept me going. There are too many friends to name individually; the best way to acknowledge them is perhaps to make sure I stay in touch with them and keep reminding them how important and awesome they are. I've also been blessed with the love and support of a tightly connected extended family, who have celebrated my achievements and success as their own.

Like a tasty bite left for the end of the meal, my deepest gratitudes are in this last paragraph. These are of course for my wife, Amrita Singh, and my parents, Priti and Vijay Gupta. My parents have affirmed me with unconditional love. They have hiked a tougher world, so that I can thrive in an easier one. They have been kind, so that I can be kind. I met my wife around 11 years ago as a teenager, taking in all that the world offered. Now that I am in a position to give back, I am guided by beliefs and aspirations that we found together. I am immensely grateful to my family for enabling me to get here.

Contents

Abstract v

Acknowledgments vii

I Introduction 1

1 Background on calibration 2

1.1 Two views: model calibration and forecast calibration 3
1.2 Achieving model calibration using post-hoc methods 7
1.3 Multiclass calibration 12
1.4 Measuring calibration of models or forecasts 15

2 Thesis overview 20

2.1 Distribution-free binary calibration 20
2.2 Distribution-free multiclass calibration 21
2.3 Online post-hoc calibration 22
2.4 An application to regression 22
2.5 A modification to the forecast calibration setup 23
2.6 Nested conformal prediction and data-efficient uncertainty quantification 23
2.7 Open-source contributions 24

II Original contributions (primary) 25

3 Distribution-free binary classification: prediction sets, confidence intervals, and calibration 26

3.1 Introduction 26
3.2 Calibration, confidence intervals and prediction sets 27
3.3 Relating the notions of uncertainty quantification 31
3.4 Achieving distribution-free calibration 35
3.5 Other related work 40
3.6 Conclusion 41
3.7 Broader impact 41

Appendices for Chapter 3 43

4 Distribution-free calibration guarantees for histogram binning without sample splitting 64

4.1 Introduction . . . 64
4.2 Sample split uniform-mass binning is inefficient . . . 69
4.3 Distribution-free analysis of uniform-mass binning without sample splitting . . . 72
4.4 Simulations . . . 77
4.5 Conclusion . . . 79

Appendices for Chapter 4 80

5 Top-label calibration and multiclass-to-binary reductions 93

5.1 Introduction . . . 93
5.2 Modifying confidence calibration to top-label calibration . . . 94
5.3 Calibration algorithms from calibration metrics . . . 99
5.4 Experiments: M2B calibration with histogram binning . . . 103
5.5 Conclusion . . . 105
5.6 Reproducibility . . . 106

Appendices for Chapter 5 107

6 Online Platt scaling with calibeating 139

6.1 Introduction . . . 139
6.2 Online Platt scaling (OPS) . . . 145
6.3 Calibeating the OPS forecaster . . . 147
6.4 Experiments . . . 150
6.5 Online beta scaling with calibeating . . . 153
6.6 Conclusion . . . 154

Appendices for Chapter 6 156

7 Parity Calibration 173

7.1 Introduction . . . 173
7.2 Parity calibration via binary calibration . . . 177
7.3 Real-world case studies . . . 180
7.4 Conclusion . . . 189

Appendices for Chapter 7 190

III Original contributions (related) 200

8 Faster online calibration without randomization: interval forecasts and the power of two choices 201

8.1 Introduction . . . 201
8.2 Motivation for the POTC calibration game . . . 205

8.3 Main results: algorithm and analysis 207
8.4 $\Omega(1/\sqrt{T})$ lower bound for the classical calibration game 210
8.5 Summary 215

Appendices for Chapter 8 **217**

9 Nested conformal prediction and quantile out-of-bag ensemble methods **227**
9.1 Introduction 227
9.2 Split conformal based on nested prediction sets 230
9.3 Cross-conformal and Jackknife+ using nested sets 232
9.4 Extending ensemble based out-of-bag conformal methods using nested sets . . . 236
9.5 QOOB: A novel conformal method using nested sets 238
9.6 Numerical comparisons 240
9.7 Conclusion 249

Appendices for Chapter 9 **250**

10 Conclusion and future work **266**

Bibliography **282**

Part I

Introduction

Chapter 1 introduces the formal goal of calibration. We discuss techniques commonly used to achieve calibration, with a focus on the post-hoc calibration of machine learning classifiers.

Chapter 2 overviews the scientific contributions arising from this thesis.

Chapter 1

Background on calibration

Consider a canonical machine learning (ML) application. A bank that wants to build an ML model for approving credit outsources this task to an ML firm. The firm reverts with a *score-based* classification model; let us refer to this model as $g : \mathcal{X} \to [0, 1]$. $\mathcal{X}$ is the space of some predictive features for a loan/credit seeker and $[0, 1]$ is the space of risk-scores. If we trust g, a higher risk-score implies a higher chance of default on the loan.

The bank uses g to make a decision $\widehat{y} : \mathcal{X} \to \{0 \equiv \text{reject loan}, 1 \equiv \text{approve loan}\}$ by using some decision threshold $t \in [0, 1]$: $\widehat{y}(\cdot) := \mathbb{1}\{g(\cdot) \leqslant t\}$. Say the bank uses $t = 0.1$. Over the next year, the bank finds that almost 95% of the credit seekers who were given a loan have good credit standing and are paying back their loans on time. Thus the model $\mathbb{1}\{g(\cdot) \leqslant 0.1\}$ appears to do its job.

The bank sends the collected data over the year to the ML firm, and asks them to use it to improve g. The firm offers a new model g_{new} and claims that it is much better than g. The bank then deploys the new model with the same threshold, 0.1, $\widehat{y}(\cdot) = \mathbb{1}\{g_{\text{new}}(\cdot) \leqslant 0.1\}$. However, over the next year only 80% of approved loans are paid back on time.

What happened? Perhaps g_{new} is not a better model as the ML firm claimed. However, it turns out that the above situation is possible even if a perfect threshold exists for the new model, say $y = \mathbb{1}\{g_{\text{new}}(\cdot) \leqslant 0.05\}$, which would makes g_{new} an excellent model. What can the ML firm do to ensure that the right threshold is used in downstream applications?

One solution is to ensure that the model produces *calibrated* scores: on the instances where the score is p, the frequency of $Y = 1$ is also p. This allows the bank to decide their threshold independently, based on levels of risk tolerance that are unknown to the ML firm. Understanding calibration and developing methods for building calibrated models is the primary goal of this thesis.

In this chapter, we define the notion of calibration and discuss prior work on making calibrated predictions. Chapters 3–9 discuss methods that achieve calibration. A question we do not discuss in significant depth is, “Why calibration?” While we are not positioned to answer that question conclusively, we discuss it briefly in Chapter 10.

The topic of calibration can be placed under the broader field of "uncertainty quantification" in machine learning, where the goal is to supplement point estimates such as class predictions or regression outcomes with uncertainties. Unfortunately, the term "calibration" has appeared with multiple meanings in this field. Unless explicitly specified, when we say "calibration", we always mean calibration of probabilistic forecasts over categorical events, such as class membership.

1.1 Two views: model calibration and forecast calibration

Calibration has been studied in two setups with limited knowledge crossover. A large focus in this thesis is *model calibration*, where we are interested in learning calibrated ML models (Platt, 1999; Zadrozny and Elkan, 2001; Guo et al., 2017), such as the model g introduced earlier. Whether a given model is calibrated depends on how the data is generated. A natural proposition is to assume that data is being generated from some distribution P and ask if the model is calibrated for this P.

On the other hand, calibration has also been studied in an online learning or individual sequence style setup where data can be arbitrary/adversarial (Dawid, 1982; Foster and Vohra, 1998; Fudenberg and Levine, 1999). We refer to this setup as *forecast calibration.*

The literature on model calibration has evolved quite independently of the literature on forecast calibration. Neither setup perfectly represents the real world—data does not follow a distribution, nor is it adversarial. Yet, studies in both setups have led to the development of interesting and practically useful calibration algorithms. We place this thesis in context of both these rich strands of literature. The culminating project of this thesis (Chapter 6) provides one way to tie these views of calibration together.

1.1.1 Model calibration for binary classifiers

Let $g : \mathcal{X} \to [0, 1]$ be an ML model or binary classifier that takes as input a feature vector in the feature space $\mathcal{X}$ and outputs a score in $[0, 1]$. Let P be the data distribution over $\mathcal{X} \times \{0, 1\}$ and let $(X, Y) \sim P$ denote a random data-point. If g is a good model, we expect that higher scores $g(X)$ indicates a higher *chance*[1] of $Y = 1$. Model calibration requires that this hold in a particular sense defined next.

Definition 1.1 (Model calibration)**.** A model $g : \mathcal{X} \to [0, 1]$ is said to be calibrated if

$$P(Y = 1 \mid g(X)) = g(X). \tag{1.1}$$

Exact model calibration, as defined above, is a guiding ideal rather than a practically achievable goal. Even if real world data were being generated from some distribution P, we cannot learn P exactly. Thus model calibration can only be satisfied approximately. We formalize such a definition of approximate calibration in Chapter 3, which is based on Gupta et al. (2020).

[1] The word "chance" in non-technical and refers to a predicted score without a formal probabilistic interpretation. In particular, "chance" should not be interpreted as "probability".

Definition 1.2 ((ϵ, α)-calibration)**.** Let $\epsilon \in (0, 1)$ be a tolerance level of miscalibration and $\alpha \in (0, 1)$ be a tolerance level for probability of failure. A model $g : \mathcal{X} \to [0, 1]$ is said to be (ϵ, α)-calibrated (for the data-generating distribution P) if

$$P\left(|P(Y = 1 \mid g(X)) - g(X)| \geqslant \epsilon\right) \leqslant \alpha. \tag{1.2}$$

ML models do not satisfy approximate calibration (for small (ϵ, α)) out-of-the-box. However, even if an ML model is not calibrated, we expect it to satisfy a rough monotonicity property—higher scores should indicate a higher probability of the class being 1. For example, if g classifies well, it means that there exists a classification threshold $t \in [0, 1]$ such that $\mathbb{1}\left\{g(\cdot) \geqslant t\right\}$ is accurate. This intuition is central to the paradigm of post-hoc calibration. Post-hoc calibration methods produce calibrated models by recalibrating the scores produced by g. Section 1.2 discusses past work on post-hoc calibration. One of the primary goals of this thesis is a distribution-free analysis of post-hoc calibration techniques.

Historical context. It is natural to ask for the probabilities of ML classifiers to be meaningful, or in particular be calibrated, and this question has been considered by a number of "classical" papers (Platt, 1999; Zadrozny and Elkan, 2001; Zadrozny and Elkan, 2002; Provost and Domingos, 2003; Niculescu-Mizil and Caruana, 2005). Recently, interest in calibration has surged, stemming largely from the finding that deep neural networks are overconfident and benefit from post-hoc calibration (Guo et al., 2017). This has led to a number of empirical papers in the narrower field of deep neural network calibration; see Section 1.3 for pointers to some of this work. Our focus is on broad and fundamental calibration principles and post-hoc methods that can be applied on top of any model.

1.1.2 Forecast calibration for individual sequences

Can we produce calibrated scores without knowing *anything* about the label-generating process—even if the labels are being produced adversarially? That is, in the online learning setup popularized by the seminal works of Cover (1991), Vovk (1995), and Freund and Schapire (1997), also called individual sequences in information theory (Feder et al., 1992).

This fundamental question is formalized through the setup of forecast calibration. Forecast calibration is often studied in a broader setting not restricted to machine learning, so when discussing forecast calibration we typically switch terminology as follows:

$$\text{scores} \to \text{forecasts}, \quad \text{labels} \to \text{outcomes}.$$

Thus the problem of producing "probability scores for binary labels" becomes the problem of producing "probability forecasts for binary outcomes".

Let $y_1, y_2, \ldots \in \{0, 1\}^\infty$ be an infinite binary sequence generated by an unknown process. For example, y_t could be the indicator of whether it rains at a time t.[2] At each time t, a forecast

[2] Time is simply an index over the events we are interested in forecasting, with the understanding that the event at time $t = 1$ occurs before the event at time $t = 2$ and so on.

Panel 1 Calibration-Game-I (nature is a prescient adversary)

(Parenthesized sentences instantiate the setup for the canonical example of rain prediction.)
At time $t = 1, 2, \ldots,$

- The forecaster produces a forecast $p_t \in [0, 1]$. (In the case of rain prediction, p_t is the belief that the probability of rain at time t is p_t.)
- Nature reveals the outcome $y_t \in \{0, 1\}$. (In the case of rain prediction, $y_t = 0$ means that it does not rain at time t and $y_t = 1$ means that it rains at time t.)

Nature knows p_t before revealing y_t.

Panel 2 Calibration-Game-II (nature is adaptive, but not prescient)

At time $t = 1, 2, \ldots,$

- Forecaster plays $u_t \in \Delta([0, 1])$.
- Nature plays $y_t \in \{0, 1\}$.
- Forecaster predicts $p_t \sim u_t$.

Nature knows u_t, the distribution of p_t, before revealing y_t, but not p_t itself.

$p_t \in [0, 1]$ for the probability of y_t is to be made. Before revealing p_t, the forecaster knows (p_s, y_s) for $s \leqslant t$. Before revealing y_t, nature knows (p_s, y_s) for $s \leqslant t$, as well as p_t. We put this setup in Panel 1 and refer to it as Forecast-Calibration-Game-I.

We define what it means for the forecasts to be calibrated. For some $x \in [0, 1]$, define

$$N_x^T := \sum_{t=1}^{T} \mathbb{1}\{p_t = x\}$$

as the number of times the probability x was forecasted until time T. If $N_x^T > 0$,

$$p_x^T := \sum_{t=1}^{T} y_t \mathbb{1}\{p_t = x\} / N_x^T$$

is defined as the average of the outcomes y_t when x was forecasted.

Definition 1.3 (Forecast calibration). Forecasts $(p_1, p_2, \ldots) \in [0, 1]^\infty$ are said to be calibrated if

$$\text{for all } x \text{ such that } \lim_{T\to\infty} N_x^T \to \infty, \text{ we have } \lim_{T\to\infty} p_x^T = x. \tag{1.3}$$

In words, for each forecast x that is made infinitely often, the average of the observations y_t over instances on which x was forecasted, equals x.

The forecaster's goal is to ensure that the forecasts satisfy (1.3) no matter how nature behaves. Nature's goal is to make the forecaster appear miscalibrated.

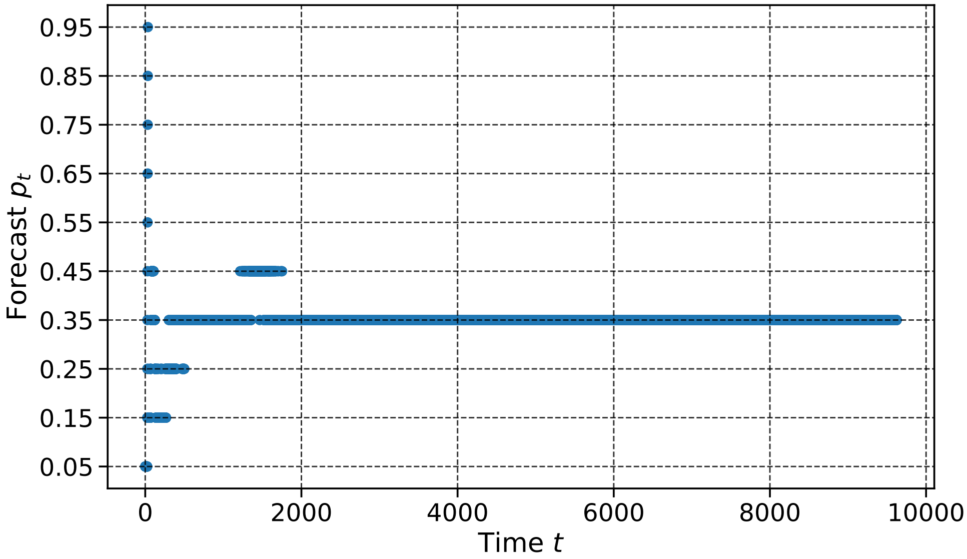


Figure 1.1: Foster (1999)'s ϵ-calibrated forecaster on Pittsburgh hourly rain data (2008-2012). The forecaster makes predictions on the grid $(0.05, 0.15, \ldots, 0.95)$. In the long run, the forecaster starts predicting 0.35 for every instance, closely matching the average number of instances on which it rained (≈ 0.37).

Since nature sees p_t, it is easy to satisfy her goal: play $y_t = \mathbb{1}\{p_t \leqslant 0.5\}$.[3] However, forecast calibration becomes possible with a mild weakening of nature. Namely, we allow the forecaster to make randomized forecasts, and nature is allowed to see everything but the random bits of the forecaster. Withholding access to the forecaster's random bits is an extremely mild restriction on nature—an equivalent way of stating it is that the pseudorandom bits on the forecaster's computer are statistically independent of the outcomes being forecasted using that computer.

This setup is capture in Forecast-Calibration-Game-II (Panel 2). The forecaster now plays a $u_t \in \Delta[0, 1]$, where $\Delta[0, 1]$ is the space of probability measures over $[0, 1]$. The actual forecast p_t is drawn from u_t in parallel with nature's play y_t. That is, nature sees u_t but not y_t before revealing y_t. In a seminal result, Foster and Vohra (1998) showed that the forecaster can satisfy (1.3) with probability one (over the random bits of the forecaster), irrespective of nature's strategy.

Although Foster and Vohra's result guarantees calibrated forecasting, this does not immediately imply that the forecasts are useful. To see this, suppose it rains on every alternate day, $y_t = \mathbb{1}\{t \text{ is odd }\}$. The forecast $p_t = \mathbb{1}\{t \text{ is odd }\}$ is calibrated and very useful (if you know p_t, you know y_t). The forecast $p_t = 0.5$ (for every t) is also calibrated, but not very useful.

Thus we need to assess how a forecaster guaranteed to be calibrated for adversarial sequences performs on real-world sequences. In order to do so, we implemented the calibrated forecaster of Foster (1999) on Pittsburgh hourly rain data from January 1, 2008, to December 31, 2012. The

[3] This simple construction has a significant implication for Bayesian statistics; it implies that nature can force a Bayesian following the coherency principle into a Russel's paradox (Dawid, 1982; Oakes, 1985; Dawid, 1985).

Panel 3 Post-hoc calibration of a pre-learnt model using held-out calibration data

Given a

$$\text{pre-learnt model } g : \mathcal{X} \to [0,1] \text{ and calibration data } \mathcal{D} \sim P^n,$$

produce an estimate

$$m : [0,1] \to [0,1] \text{ of the mapping } g(X) \mapsto P(Y = 1 \mid g(X)).$$

If m is a good estimate, then

$$h := m \circ g \equiv m(g(\cdot)) \text{ is better calibrated than } g \text{ (for } P\text{)}.$$

data was obtained from ncdc.noaa.gov/cdo-web/. All days on which the hourly precipitation in inches (HPCP) was at least 0.01 were considered as instance of $y_t = 1$. There are many missing rows in the data, but no complex data cleaning was performed since we are mainly interested in a simple illustrative simulation. Foster (1999)'s forecaster makes forecasts on a discrete ϵ-grid and achieves ϵ-calibration, a precursor to satisfying (1.3). We implement the algorithm for the grid $(0.05, 0.15, \ldots, 0.95)$. We observe (Figure 1.1) that after around 2000 instances, the forecaster *always* predicts 0.35. This is close to the overall average number of instances that it did rain, which is approximately 0.37.

Thus, while it is remarkable that calibration can be achieved against adversarial sequences, we must do more than calibration. In the ML setting, informative features are available to predict the label. The simplest way to capture this information is to assume that the data-points are drawn identically and independently from some unknown distribution. Then the calibration of an ML model can be assessed with respect to that unknown distribution. This is exactly what model calibration captures (see previous subsection). However, even if the data points are not independent and non-stationary, these predictive features contain useful information about the label. How can we leverage this information?

The culminating project of this thesis is an algorithm that is simultaneously robust to worst-case data and adaptive to the information offered by informative features (Chapters 6 and 7, which are based on Gupta and Ramdas (2023) and Chung et al. (2023)). See Chapter 2 for a longer preview of these works.

1.2 Achieving model calibration using post-hoc methods

Let $g : \mathcal{X} \to [0,1]$ be any pre-learnt model, such as a deep-net, random forest, or SVM with a sigmoid transformation (to ensure that the output is in $[0,1]$). We suspect that g is miscalibrated and want to calibrate it. Consider the function $f(X) = P(Y = 1 \mid g(X))$. It can be shown (see Proposition 3.1) that f is calibrated irrespective of the calibration of g.

Post-hoc calibration or recalibration methods estimate f by fitting a function $m : [0,1] \to [0,1]$ that estimates the map $g(X) \mapsto P(Y = 1 \mid g(X))$ is produced. Then $h := m \circ g \equiv m(g(\cdot))$ is

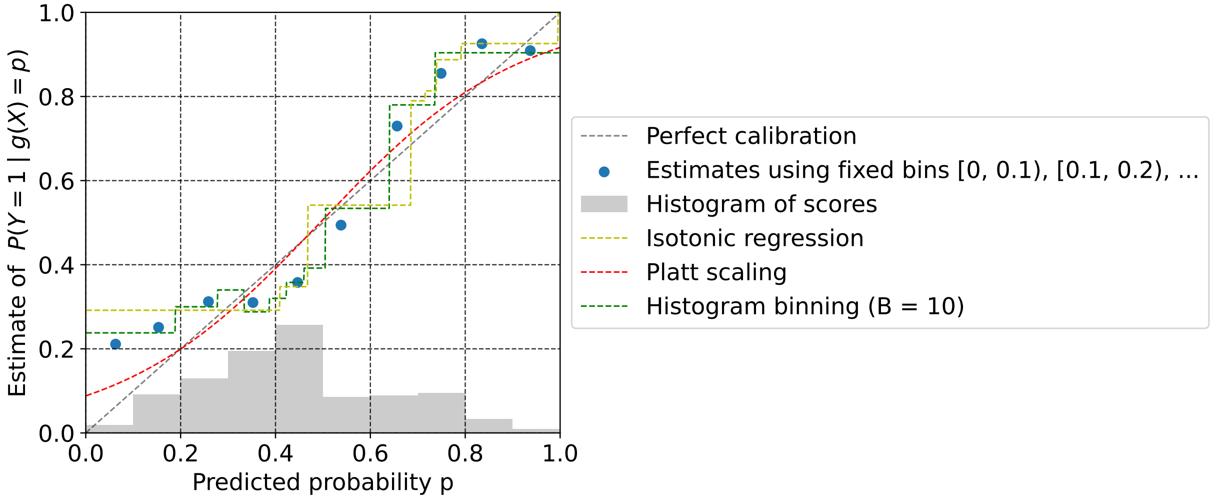


Figure 1.2: Post-hoc calibration of a logistic regression model $g : \mathcal{X} \to [0, 1]$. The plot is made on out-of-sample data not used while training g. The blue scatter plot is a *reliability diagram* for the model g, as described in Section 1.4. The scatter points deviate from the perfect calibration line, so we conclude that g is miscalibrated.
Post-hoc calibration methods produce an estimate $m : [0, 1] \to [0, 1]$ of the mapping $g(X) \mapsto P(Y = 1 \mid g(X))$. Platt scaling (Section 1.2.1) produces a smooth curve from a parametric family. Histogram binning (Section 1.2.2) and isotonic regression (Section 1.2.3) produce a piecewise constant curve—the interval $[0, 1]$ is divided into a number of bins and all scores in a given bin are mapped to a single output.

an estimate of f. The mapping m is learnt on fresh held-out i.i.d. data on which g was not learnt, called the *calibration data*. We denote the calibration data as

$$\mathcal{D} := (X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n) \overset{i.i.d.}{\sim} P. \tag{1.4}$$

The paradigm of post-hoc calibration methods is summarized in Panel 3. In a nutshell, post-hoc methods allow the (typically complex) modeling of the feature space $\mathcal{X}$ to be controlled by the method that is producing g. Once g is learnt, a simple scalar-to-scalar mapping can be learnt to calibrate it.

In Chapter 3, which is based on Gupta et al. (2020), we formalize Panel 3 in a distribution-free setup.

Three methods for post-hoc calibration were proposed in close succession: Platt scaling (Platt, 1999), histogram binning (Zadrozny and Elkan, 2001), and isotonic regression (Zadrozny and Elkan, 2002). Each of these methods is based on the inductive bias that the predicted scores $g(X)$ are roughly monotonic with $P(Y = 1 \mid g(X))$. We illustrate these methods on a UCI credit default dataset (Figure 1.2), a binary dataset with about 78% occurence of $Y = 0$ (no credit default).[4] For better illustration, we subsampled the $Y = 0$ instances to make them about 66%. There are 23 predictive features such as age, education, and past payment history. A

[4] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

logistic regression model was trained on 10,000 training points to learn a model g. After training, evaluation was performed on an unseen calibration set of size 5,000. The accuracy on this set was around 70%. To assess calibration, the prediction scores $g(x)$ were binned into consecutive bins $[0, 0.1), [0.1, 0.2), \ldots [0.9, 1.0]$ and for each bin, the average $g(x)$ and the fraction of instances of $y = 1$, were computed. These values are plotted as the blue scatter plot in Figure 1.1. The light grey histogram also shows the distribution of the scores $g(x)$ on the calibration data. If g was approximately calibrated, the blue points would be close to the perfect calibration line.[5]

However, g appears miscalibrated. So, we look to estimate the mapping m on the same calibration data, as described in Panel 3. The estimates produced by the aforementioned methods—Platt scaling, histogram binning, and isotonic regression—are plotted in Figure 1.2. In the following subsections, we describe these methods, and other related ones.

1.2.1 Platt scaling and beta scaling

Platt scaling (Platt, 1999) learns the mapping from a parametric family

$$\mathcal{M}_{\text{platt}} = \{m^{a,b} : a, b \in \mathbb{R}^2\}, \tag{1.5}$$

where $m^{a,b}$ is given by

$$m^{a,b}(z) = \text{sigmoid}(a \cdot z + b) = 1/(1 + e^{-(az+b)}). \tag{1.6}$$

The parameters (a, b) are learnt as those that maximize the likelihood of $\mathcal{D}$, assuming each Y_i is independently drawn from Bernoulli$(m^{a,b}(g(X_i)))$. In the credit default experiment (Figure 1.2), the learnt parameters were $a \approx 4.7$ and $b \approx -2.3$. Thus the inflection point of the curve is roughly around $0.49 \approx 2.3/4.7$.

A slightly different version of Platt scaling has the mapping $m^{a,b}$ given by

$$m^{a,b}(z) = \text{sigmoid}(a \cdot \text{logit}(z) + b) = 1/(1 + e^{-(az+b)}), \tag{1.7}$$

where $\text{logit}(z) = \log(z/1 - z)$. This is the version we focus on in Chapter 6, since it seemed to perform better in our experiments. We do not offer any detailed theory or comparison of the two alternatives in this thesis, but we have sufficient evidence that (1.7) often works well.

A recalibration method closely related to Platt scaling is beta scaling (Kull et al., 2017). The beta scaling mapping m has three parameters $(a, b, c) \in \mathbb{R}^3$:

$$m^{a,b,c}(z) := \text{sigmoid}(a \cdot \log(z) + b \cdot \log(1 - z) + c).$$

Observe that enforcing $b = -a$ recovers the Platt scaling mapping (1.7) since $\text{logit}(z) = \log(z) - \log(1 - z)$.

In Chapter 3, which reproduces Gupta et al. (2020), we show that Platt scaling cannot satisfy certain “distribution-free calibration guarantees”. See Chapter 2 for a longer preview of this work.

[5] This is typically called the X=Y line, referring to the X and Y axes. We include this in a footnote instead of the main text to avoid confusion with the random variables X and Y.

Algorithm 1.1 Histogram binning

```
 1: Input: #bins $B \in \mathbb{N}$, calibration data $\mathcal{D} = (X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$
 2: Output: Recalibration mapping $m : [0, 1] \to [0, 1]$
 3: Compute scores: $(S_1, S_2, \dots, S_n) \leftarrow (g(X_1), g(X_2), \dots, g(X_n))$
 4: Sort scores: $(S_{(1)}, S_{(2)}, \dots, S_{(n)}) \leftarrow$ order-statistics$(S_1, S_2, \dots, S_n)$
 5: Set $Y_i$ values as per the ordering of $(S_{(1)}, S_{(2)}, \dots, S_{(n)})$: $(Y_{(1)}, Y_{(2)}, \dots, Y_{(n)})$
 6: Set approximate #points-per-bin: $\Delta \leftarrow (n + 1)/B$
 7: Create an array to store bin biases: $\widehat{\Pi} \leftarrow$ empty array of size $B$
 8: Create an array of indices: $A \leftarrow$ 0-indexed array$([0, \lceil \Delta \rceil, \lceil 2\Delta \rceil, \dots, n + 1])$
 9: for $b \leftarrow 1$ to $B$ do
10:     Left order-statistic index: $l \leftarrow A_{b-1}$
11:     Right order-statistic index: $u \leftarrow A_b$
12:     Compute bias for bin $b$: $\widehat{\Pi}_b \leftarrow$ Mean$(Y_{(l+1)}, Y_{(l+2)}, \dots, Y_{(u-1)})$
13: end for
14: Set $(S_{(0)}, S_{(n+1)}) \leftarrow (0, 1)$
15: Define final mapping: $m(\cdot) \leftarrow \sum_{b=1}^{B} \mathbb{1}\left\{S_{(A_{b-1})} \leqslant \cdot < S_{(A_b)}\right\} \widehat{\Pi}_b$
```

Left endpoint	0.0	0.19	0.28	0.33	0.39	0.42	0.46	0.5	0.64	0.74
Right endpoint	0.19	0.28	0.33	0.39	0.42	0.46	0.5	0.64	0.74	1.0
Bin bias	0.24	0.3	0.34	0.29	0.32	0.36	0.39	0.53	0.78	0.90

Table 1.1: Approximate bin boundaries and biases learnt by histogram binning for a logistic regression model on credit default data (Figure 1.2 experiment).

1.2.2 Histogram binning

In histogram binning (Zadrozny and Elkan, 2001) one learns a nonparametric mapping m. This mapping is based on the idea of binning, wherein nearby values of $g(x)$ are grouped together into some number of bins, and a single estimate of the probability of $Y = 1$ is computed for each bin.

Algorithm 1.1 is a rendition of histogram binning directly borrowed from Gupta and Ramdas (2021). There is one hyperparameter, $B \in \mathbb{N}$, the number of bins. The interval $[0, 1]$ is partitioned into B bins using the $g(X_i)$ values, to ensure that each bin has the same number of calibration points (plus/minus one). Thus the bins have nearly *uniform (probability) mass.* Then, the calibration points are assigned to bins depending on the interval to which the score $g(X_i)$ belongs to, and the probability that $Y = 1$ is estimated for each bin as the average of the observed Y_i-values in that bin (line 12). This average estimates the *biases* of the bin ($\widehat{\Pi}_b$ estimates). The binning scheme and the bias estimates together define m (line 15).

The bins and biases estimated using histogram binning in the credit default experiment are displayed visually in Figure 1.2) and numerically in Table 1.1.

Left endpoint	0.0	0.41	0.47	0.69	0.72	0.74	0.79	0.996
Right endpoint	0.41	0.47	0.69	0.72	0.74	0.79	0.996	1.0
Bin bias	0.29	0.35	0.54	0.79	0.81	0.89	0.93	1.0

Table 1.2: Approximate bin boundaries and biases learnt by isotonic regression for a logistic regression model on credit default data (Figure 1.2 experiment).

Naeini et al. (2015) described a generative post-hoc calibration model that has histogram binning at its core, by specifying a prior over the number of bins and the bias parameters for each bin, and the data likelihood under each value of the parameters. They called their method BBQ for Bayesian Binning into Quantiles. Recently Valk and Kull (2023) suggested that instead of estimating a single probability of $Y = 1$ in each bin (like in Algorithm 1.1), we estimate the *calibration error* in each bin. Then, the post-hoc model is created by adding this estimated calibration error to the score of the pre-hoc model (see Figure 1 of their paper).

In Chapter 4, which reproduces Gupta and Ramdas (2021), we show that histogram binning satisfies "distribution-free calibration guarantees". These are the first such guarantees known for any post-hoc calibration method. See Chapter 2 for a longer preview of our work.

1.2.3 Isotonic regression

Isotonic regression is a shape-constrained regression method popularized by Barlow and Brunk (1972). The application to post-hoc calibration was considered by Zadrozny and Elkan (2002). The isotonic regression family corresponds to the nonparametric class of monotonically increasing mappings:

$$\mathcal{M}_{\text{isotonic}} = \{m : \text{for all } 0 \leqslant x \leqslant y \leqslant 1, m(x) \leqslant m(y)\}. \tag{1.8}$$

Let $Z_i = g(X_i)$. The isotonic estimator is derived from a solution of the following shape-constrained regression problem:

$$\begin{aligned} \underset{\widehat{\mu}_1, \widehat{\mu}_2, \ldots, \widehat{\mu}_n \in [0,1]}{\text{minimize}} \quad & \sum_{i=1}^{n} (Y_i - \widehat{\mu}_i)^2, \\ \text{such that} \quad & \forall i, j, \ \widehat{\mu}_i \leqslant \widehat{\mu}_j \iff Z_i \leqslant Z_j. \end{aligned} \tag{1.9}$$

This solution can be learnt efficiently using the pool-adjacent-violators-algorithm (PAVA) (Ayer et al., 1955; Barlow, 1972). Given an optimal solution $\widehat{\mu}_1^\star, \widehat{\mu}_2^\star, \ldots, \widehat{\mu}_n^\star$, $m : [0, 1] \to [0, 1]$ assigns to every $z \in [0, 1]$, the $\widehat{\mu}_i$ corresponding to the largest Z_i to the left of z:

$$m(z) = \widehat{\mu}_j^\star, \text{where } Z_j = \max\{Z_i : Z_i \leqslant z\}. \tag{1.10}$$

The above can also be written (in a perhaps easier-to-follow form) as

$$m(z) = \max\{\widehat{\mu}_i^\star : Z_i \leqslant z\}, \tag{1.11}$$

because of the monotonicity constraint in the optimization problem (1.9).

Thus, like histogram binning, the isotonic regression solution is also a number of partition of $[0, 1]$ into bins, and bias estimates for each bin. The bins and biases estimated using isotonic regression in the credit default experiment are displayed visually in Figure 1.2 and numerically in Table 1.2. Notice that histogram binning forms fewer bins than histogram binning. This is because of isotonic regression's monotonicity constraint. Histogram binning allows the bias for bin $[0.28, 0.33)$, which is 0.34, to be larger than the bias for bin $[0.33, 0.39)$, which is 0.29. Due to the monotonicity constraint, isotonic regression is forced to merge as part of a single bin $[0, 0.41)$.

A relaxation to the monotonicity constraint was considered by Tibshirani et al. (2011) and consequently adapted for post-hoc calibration by Naeini et al. (2014). Other Bayesian and ensemble versions of isotonic regression have also been considered (Allikivi and Kull, 2020; Naeini and Cooper, 2016). Recently, a calibration guarantee for isotonic regression was claimed by Laan et al. (2023).

1.3 Multiclass calibration

Consider the setup of multiclass classification, with $L \geqslant 3$ classes and labels $Y \in [L] := \{1, 2, \ldots, L \geqslant 3\}$. As in the binary case, we assume all (training and test) data is drawn i.i.d. from a fixed distribution P, and denote a general point from this distribution as $(X, Y) \sim P$. Consider a typical multiclass predictor, $\mathbf{h} : \mathcal{X} \to \Delta^{L-1}$, whose range Δ^{L-1} is the probability simplex in $\mathbb{R}^L$. A natural notion of calibration for $\mathbf{h}$, called *canonical calibration* is the following: for every $l \in [L], P(Y = l \mid \mathbf{h}(X) = \mathbf{q}) = q_l$. Here, q_l denotes the l-th component of $\mathbf{q}$. However, canonical calibration becomes infeasible to achieve or verify once L is even 4 or 5 (Vaicenavicius et al., 2019). Thus, there is interest in studying statistically feasible relaxations of canonical notion, such as confidence calibration (Guo et al., 2017), top-label calibration (Gupta and Ramdas, 2022b), class-wise calibration (Kull et al., 2017), and top-K-confidence calibration (Gupta et al., 2021).

We unified these various relaxations of canonical calibration into a single "multiclass-to-binary" framework (Gupta and Ramdas, 2022b). This paper is reproduced as Chapter 6. In this chapter, we also introduced the aforementioned notion of top-label calibration, which focuses on a single binary calibration requirement corresponding to the predicted top class, called the top-label in this context. A brief description of top-label calibration is provided next.

Top-label calibration. A classifier is said to be top-label calibrated if the reported probability for the top-label is calibrated (in a binary sense), conditioned on the top-label. Let $c : \mathcal{X} \to [L]$ denote a class predictor (for the top-label) and $h : \mathcal{X} \to [0, 1]$ a function that provides a probability score for the top-label $c(X)$. For an L-dimensional predictor $\mathbf{h} : \mathcal{X} \to \Delta^{L-1}$, one would use $c(\cdot) = \arg\max_{l \in [L]} h_l(\cdot)$ and $h(\cdot) = h_{c(\cdot)}(\cdot)$ (breaking ties arbitrarily). The forthcoming definition is for top-label calibration of (c, h); a vector-valued $\mathbf{h}$ is top-label calibrated if the induced (c, h) is top-label calibrated.

Definition 1.4 (Top-label calibration)**.** The predictor (c, h) is said to be top-label calibrated (for the data-generating distribution P) if

$$P(Y = c(X) \mid c(X), h(X)) = h(X). \tag{1.12}$$

In other words, if conditioned on the top-label $c(X)$, when the reported confidence $h(X)$ equals $p \in [0, 1]$, then the fraction of instances where the predicted label is correct also equals p.

Top-label calibration is related to and inspired from the popular notion of confidence calibration (Guo et al., 2017), but solves a key interpretability issue, discussed in detail in Chapter 5.

We next discuss post-hoc multiclass calibration methods. The literature on multiclass calibration has flourished since the finding that deep neural networks are overconfident and benefit from post-hoc calibration (Guo et al., 2017). Due to the fast-moving nature of this field, we do not provide an exhaustive review of all methods; the main focus of this thesis is on fundamental calibration principles which are readily studied in the setup of binary calibration.

1.3.1 Multiclass variants of Platt scaling

The class probabilities for a neural network are a softmax over logits $\mathbf{z} \in \mathbb{R}^L$ learnt by the network,

$$p_l = \frac{\exp(z_l)}{\sum_{s=1}^{L} \exp(z_s)},$$

where z_s is the s-th component of $\mathbf{z}$. The scaling methods we describe below are modifications of the above mapping from logits to probabilities. Temperature, vector, and matrix scaling were proposed by Guo et al. (2017) and Dirichlet calibration was proposed by Kull et al. (2019).

Temperature scaling is characterized by a scalar $t > 0$ and the post-hoc class probabilities,

$$q_l = \frac{\exp(tz_l)}{\sum_{s=1}^{L} \exp(tz_s)}. \tag{1.13}$$

If $t > 1$, the q-probabilities are more dispersed than the p-probabilities (larger ones becomes larger, smaller ones become smaller). If $t < 1$, the q-probabilities are more centered.

Vector scaling is characterized by two vectors $\mathbf{t}, \mathbf{b} \in \mathbb{R}^L$. The post-hoc class probabilities are defined as,

$$q_l = \frac{\exp(t_l z_l + b_l)}{\sum_{s=1}^{L} \exp(t_s z_s + b_s)}. \tag{1.14}$$

Vector scaling is equivalent to having L Platt scaling mappings, one for each class, and then normalizing the probabilities to sum to one.

Matrix scaling is characterized by a matrix $\mathbf{M} \in \mathbb{R}^{L \times L}$ and a vector $\mathbf{b} \in \mathbb{R}^L$. The post-hoc class probabilities are defined as,

$$q_l = \frac{\exp((\mathbf{Mz})_l + b_l)}{\sum_{s=1}^{L} \exp((\mathbf{Mz})_s + b_s)}, \tag{1.15}$$

where $(\mathbf{Mz})_s$ is the s-th component of the matrix-vector product of $\mathbf{M}$ and $\mathbf{z}$. Matrix scaling allows us to represent relationships between pairs of classes which is not possible by temperature or vector scaling. For instance if $M_{11} = 0.8$ and $M_{12} = 0.1$, this can be interpreted as "high values of p_1 should contribute to a high value of q_1 but also an increase in q_2". Restricted versions of matrix scaling, where the degrees of freedom of $\mathbf{M}$ are limited in some manner, are naturally derived and might be more practically suitable. Vector scaling is one such restriction, where the matrix $\mathbf{M}$ is limited to be diagonal. Another regularization technique called "Off-Diagonal and Intercept Regularisation" was proposed by Kull et al. (2019).

Dirichlet scaling is similar to matrix scaling but over logarithms of the p-probabilities, instead of the logits. Below, $\log \mathbf{p}$ is a component-wise natural logarithm applied to the vector $\mathbf{p}$:

$$q_l = \frac{\exp\left((\mathbf{M}\log\mathbf{p})_l + b_l\right)}{\sum_{s=1}^{L}\exp\left((\mathbf{M}\log\mathbf{p})_s + b_s\right)}. \tag{1.16}$$

1.3.2 Multiclass calibration using binary methods

A natural approach to multiclass calibration is via reduction to L one-versus-all binary problems. That is, to calibrate the pre-hoc model $\mathbf{g} : \mathcal{X} \to \Delta^{L-1}$, we calibrate the l-th component of $\mathbf{g}$ for the event $\mathbb{1}\{Y = l\}$, for each l. A normalization can be performed in the end to make the final calibrated probabilities sum to one. Any binary calibration method such as Platt scaling or histogram binning can be used to solve the L binary problems.

The earliest reference to this one-versus-all reduction (that we are aware of) is in the work of Zadrozny and Elkan (2002). This method was also used by Guo et al. (2017), as well as other follow-up papers such as Kull et al. (2019) and Kumar et al. (2019).

In Chapter 5, which is a reproduction of Gupta and Ramdas (2022b), we expanded this one-versus-all approach into a broader multiclass-to-binary (M2B) framework. Indeed, for each popular relaxation of canonical calibration that has been proposed, we show that an M2B reduction adapted to that relaxation can be derived. A one-versus-all reduction is then just a special case related to the class-wise calibration notion (Definition 5.2). Binary calibration is a cleaner problem more amenable to principled analysis, and an M2B reduction allows us to *lift* well-established binary methods to multiclass calibration.

1.3.3 Other multiclass calibration methods

We review a few more multiclass calibration methods that are not either adaptations-of-Platt-scaling or based on multiclass-to-binary reductions. Our review is not exhaustive by any means. New methods are proposed frequently and we only cover a subset of them that have become common baselines (as of 2023).

Intra-order preserving functions. Rahimi et al. (2020) characterized the class of mappings from $\mathbf{p} \in \mathbb{R}^L$ to $\mathbf{q} \in \mathbb{R}^L$ that are continuous, do not change the internal order of probabilities (the ranking of the L classes should be the same whether we rely on $\mathbf{p}$ or $\mathbf{q}$), and order-invariant among the components of $\mathbf{p}$ (in a certain sense; see Definition 3 of their paper). A mapping satisfying these properties is learnt based on a monotonic neural network (Wehenkel and Louppe, 2019), by optimizing regularized negative log-likelihood. See Section 4 in their paper for more details.

Gaussian process calibration. Wenger et al. (2020) described a Bayesian approach to post-hoc calibration. They described a Gaussian process prior over the post-hoc mapping, a categorical likelihood of the data given the post-hoc mapping, and a variational inference procedure for approximating the posterior.

Mutual information maximization-based binning. Patel et al. (2021) proposed a multiclass binning method wherein bin boundaries are shared across classes. The common binning scheme is identified so that the mutual information between the bin identities and the empirical class distribution is maximized (see equation (2) in their paper).

The state of the field on multiclass post-hoc calibration is in some aspects similar to the state of supervised learning prior to the advent and success of deep neural networks—a number of interesting competitive methods exist, but none have been unequivocally demonstrated to be better than all the rest. It is common to propose new methods that baseline against a slew of existing methods, beat existing methods, but get beaten by a future method.

1.4 Measuring calibration of models or forecasts

Given access to some test points, we would like to know how calibrated a given model g or sequence of forecasts is. In this section, we discuss some common techniques used for doing this, such as binning, reliability diagrams, validity plots, and notions of calibration error. We discuss binary calibration first, and multiclass calibration in Section 1.4.3.

Two peculiar technical difficulties arise when assessing calibration of a model g.

1. **Estimation of conditional probabilities.** For a test-point (x, y), $g(x)$ is immediately computed, but measuring calibration also requires estimating $\mathbb{P}(Y = 1 \mid g(X) = g(x))$. Compare this to measuring accuracy, F1-score, or AUROC, where we only need the outcome y that is directly observed. Informally, it is clear that if $\text{Range}(g)$ is not a discrete set, we are estimating infinitely many probabilities. This can only be done if we make (untestable) smoothness assumptions on the Bayes function $\pi(x) = \mathbb{P}(Y = 1 \mid g(X) = g(x))$, or target a coarser functional of π.[6]
2. **Unbiased estimators do not exist for absolute deviation.** Imminently, we will describe a popular metric called ℓ_1-calibration-error, which is the expected value of $|\mathbb{P}(Y = 1 \mid g(X)) - g(X)|$. Unbiased estimators do not exist for the ℓ_1-CE, even if the true value of $\mathbb{P}(Y = 1 \mid g(X))$ is exactly known. (More broadly, unbiased estimators do not exist for non-differentiable functionals such as the absolute value (Hirano and

[6] See Lee and Barber (2021) for a relevant formal treatment. The difficulty in estimating conditional probabilities is at the core of impossibility results in distribution-free uncertainty quantification by Barber (2020) and Gupta et al. (2020) (see Theorem 3.3 in Chapter 3).

Porter, 2012). For instance, given samples from $\mathcal{N}(\mu, 1)$, there is no unbiased estimator for $|\mu|$.) This issue is specific to the ℓ_1-CE since debiased estimates can be derived for related quantities such as $(\mathbb{P}(Y = 1 \mid g(X)) - g(X))^2$.

Both these issues are mitigated in the online forecasting setting where instead of estimating hypothetical population quantities, we can simply talk about calibration on the actual observed data (in effect, following the *weak prequential principle* (Dawid and Vovk, 1999)). We return to this in Section 1.4.2.

1.4.1 Binning-based calibration metrics

The common way of addressing the first issue above is by binning or discretizing g. The unit interval $[0, 1]$ is partitioned into $B \in \mathbb{N}$ non-overlapping intervals $\{B_b\}_{b\in[B]}$, called bins. For instance one could use the following *fixed-width* bins:

$$B_1 = \left[\frac{0}{B}, \frac{1}{B}\right), B_2 = \left[\frac{1}{B}, \frac{2}{B}\right), \ldots, B_B = \left[\frac{B-1}{B}, 1\right]. \tag{1.17}$$

We discuss a more popular strategy called *uniform-mass* binning shortly.

Let $\mathcal{B} : [0, 1] \to [B]$ be the binning function that maps a score $g(x)$ to the bin B_b that contains $g(x)$. $\mathcal{B}$ is often referred to as the "binning scheme". A binning-scheme induces a discretized version of g, $g_{\mathcal{B}} : \mathcal{X} \to [0, 1]$ given by

$$g_{\mathcal{B}}(x) = \text{mid-point}(B_{\mathcal{B}(g(x))}) = \text{MP}_{\mathcal{B}(g(x))}, \tag{1.18}$$

where $\text{MP}_b := \text{mid-point}(B_b)$. Instead of talking about calibration of the continuous-output g, we can now talk about calibraiton of $g_{\mathcal{B}}$. For $g_{\mathcal{B}}$, the conditional probabilities $P(Y = 1 \mid g_{\mathcal{B}}(X))$ are the B probabilities $P(Y = 1 \mid \mathcal{B}(X) = b)$, and can be estimated using plugin estimates. For a given a test dataset $(x'_1, y'_1), (x'_2, y'_2), \ldots, (x'_m, y'_m)$, define,

$$\begin{aligned} \text{FP}_b &:= \frac{\sum_{i:\mathcal{B}(x'_i)=b} y'_i}{|\{i : \mathcal{B}(x'_i) = b\}|} && \text{(fraction of positives in a bin)}, \\ \text{PP}_b &:= \frac{\sum_{i:\mathcal{B}(x'_i)=b} g(x'_i)}{|\{i : \mathcal{B}(x'_i) = b\}|} && \text{(mean predicted probability in a bin)}, \end{aligned}$$

unless $|\{i : \mathcal{B}(x'_i) = b\}| = 0$ in which we case we set $\text{FP}_b = \text{PP}_b = \text{MP}_b$. Also define

$$w_b := \frac{|\{i : \mathcal{B}(X'_i) = b\}|}{m} \qquad \text{(proportion of test points in bin } b\text{)}.$$

Instead of the earlier defined fixed-width bins, it is common to use *uniform-mass* binning where the bins have equal mass under the distribution of $g(X)$, so that

$$w_1 \approx w_2 \approx \ldots \approx w_B.$$

This can be ensured (almost) exactly if bin boundaries are defined using the test-data itself, or approximately if bin boundaries are defined using the validation or calibration data.

Finally, we note that it is possible that the classifier g is already *sufficiently* discrete, in which case the binning step can be skipped. For instance if $m \leqslant 50 \times \text{Range}(g)$, that is if more than 50 test-points-per-bin are available, then binning can be skipped.

Once the values $\text{FP}_b, \text{PP}_b, \text{MP}_b, w_b$ are ascertained, calibration can be assessed in the following ways.

1. **(Expected) calibration error (Naeini et al., 2015).** The ℓ_1-calibration-error (ℓ_1-CE) of a model g, sometimes called the ℓ_1-ECE for expected-calibration-error, is given by

$$\ell_1\text{-CE}(g) = \mathbb{E}\left|\mathbb{P}(Y = 1 \mid g(X)) - g(X)\right|,$$

where the expectation is over the distribution of X. A common way to estimate ℓ_1-CE is to use a plugin estimate of the ℓ_1-CE$(g_{\mathcal{B}})$:

$$\widehat{\ell_1\text{-CE}(g)} = \widehat{\ell_1\text{-CE}(g_{\mathcal{B}})} = \sum_{b \in [B]} w_b \cdot \left|\text{FP}_b - \text{PP}_b\right|.$$

Sometimes, it may be of interest to compare deviations from the mid-point MP instead of the mean predicted probability PP, so we get the following estimate:

$$\widehat{\ell_1\text{-CE}(g)} = \widehat{\ell_1\text{-CE}(g_{\mathcal{B}})} = \sum_{b \in [B]} w_b \cdot \left|\text{FP}_b - \text{MP}_b\right|.$$

Instead of the ℓ_1-CE, we can also consider the ℓ_p-CE,

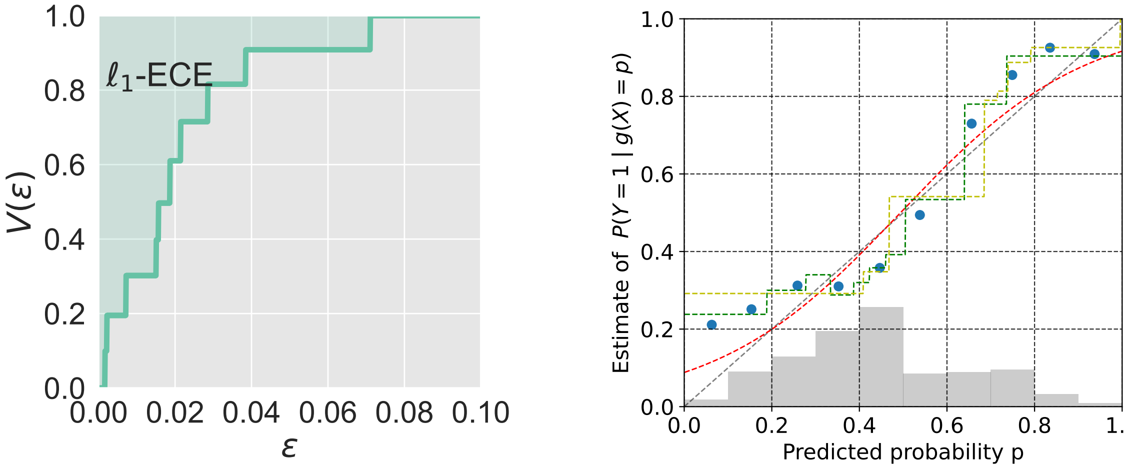
$$\ell_p\text{-CE}(g) := \left(\mathbb{E}\left|\mathbb{P}(Y = 1 \mid g(X)) - g(X)\right|^p\right)^{1/p},$$

which can in turn be estimated using a plugin estimate of ℓ_p-CE$(g_{\mathcal{B}})$. Of particular interest in literature has been the ℓ_2-CE and the ℓ_∞-CE. This ℓ_∞-CE also goes as the maximum calibration error, since

$$\ell_\infty\text{-CE}(g) = \max_p \left|\mathbb{P}(Y = 1 \mid g(X) = p) - p\right|.$$

The discretization-based estimation technique for ℓ_p-CE that we described here is a heuristic. It is a straightforward consequence of Jensen's inequality that ℓ_p-CE$(g_{\mathcal{B}}) \leqslant \ell_p$-CE$(g)$, and this has been noted by a number of papers (Kumar et al., 2019; Zhang et al., 2020; Widmann et al., 2019). Thus, using the plugin estimate for ℓ_p-CE$(g_{\mathcal{B}})$ is likely to underestimate ℓ_p-CE(g). However, we are not aware of interesting upper bounds on $|\ell_p\text{-CE}(g_{\mathcal{B}}) - \ell_p\text{-CE}(g)|$. Nixon et al. (2020) provide a broader review of common practices around the binning heuristic. We refer to Section 3.5 of this document for some more references and historical remarks.

2. **Reliability diagrams (DeGroot and Fienberg, 1983).** A reliability diagram for a classifier g is a plot of the fraction of positives in a bin (FP) versus the mean predicted



(a) An illustrative validity plot. We can read off that (ϵ, α)-calibration (1.2) is achieved for $(\epsilon, \alpha) = (0.04, 0.1)$ and $(0.03, 0.2)$. The ℓ_1-CE estimate is equal to the area-over-the-curve, which in this case is roughly 0.023.

(b) The blue scatter plot illustrates a reliability diagram, that is a plot of estimates of $P(Y = 1 \mid g(X) = p)$ against $p \in [0, 1]$. The grey bars plot the histogram of scores. All estimates are based on fixed-width bins (1.17).

Figure 1.3: Two visual ways of assessing calibration.

probability in a bin (PP). In addition, it is common to plot the bin-weights w as a histogram to show the frequency of the different bins. An example of a reliability diagram is the blue scatter plot in Figure 1.2. Guo et al. (2017) plotted the reliability diagram as a bar chart instead of a scatter plot, an unconventional choice that is now common in calibration literature due to the popularity of their work.

3. **Validity plots (Gupta and Ramdas (2021) and Chapter 4).** The original work in this thesis focuses on the *validity* of (ϵ, α)-approximate-calibration (Definition 1.2). Validity plots display the (ϵ, α) for which approximate calibration is satisfied on the given test set. Define the function $V : [0, 1] \to [0, 1]$ given by $V(\epsilon) = \mathbb{P}(|\mathbb{P}(Y = 1 \mid g(X)) - g(X)| \leqslant \epsilon)$. By definition of V, g is $(\epsilon, 1 - V(\epsilon))$-calibrated for every ϵ. V is estimated using FP and PP as follows,
$$\widehat{V}(\epsilon) = \sum_{b=1}^{B} w_b \mathbb{1}\left\{|\text{FP}_b - \text{PP}_b| \leqslant \epsilon\right\}.$$
The validity plot is a plot of $\widehat{V}$ against ϵ. Figure is an example from later in the thesis (Chapter 4). It turns out that the the area-over-the-curve of the validity plot is the plugin ℓ_1-CE estimate of g, as also illustrated in the figure.

In the following chapters, we use one or more of the techniques described above to assess calibration. However, a few other ways of measuring miscalibration have also been been proposed, and these are briefly reviewed below.

1. **Kernel density estimation based CE estimator (Zhang et al., 2020).** Instead of using bins, a kernel can be used to estimate the density $P(Y = 1 \mid g(X))$. Then, a plugin

estimator gives us a CE estimate. Consistency can be shown under certain smoothness conditions. Popordanoska et al. (2022) extended the kernel density estimation approach to multiclass canonical calibration.

2. **Kernel calibration error or KCE (Widmann et al., 2019)**. The notion of CE can be generalized using a universal kernel, to give the KCE. Unbiased estimators for the squared-KCE can be derived. While the KCE of a calibrated model must be zero, nonzero values of the KCE may be less directly interpretable (compared to CE values).
3. **Testing the hypothesis of calibration (Lee et al., 2022).** If g is calibrated, the CE as per any notion equals zero. Thus deviations of the CE from zero can be used to reject the hypothesis that g is calibrated. In particular, the above paper considers unbiased estimates of the squared ℓ_2-CE as a test statistic.

1.4.2 Measuring miscalibration of forecasts

When assessing calibration of a model g, the test data is simply a "window" into the out-of-sample performance of g on yet unseen data from the same distribution. On the other hand, when discussing forecast calibration, the actual outcome-forecast sequence $p_1, y_1, p_2, y_2, \ldots, p_T, y_T \in ([0,1] \times \{0,1\})^T$ takes centerstage, and counterfactual or future behavior of the mechanism generating forecasts is of lesser interest. Thus the question of the "true" conditional probability of $y = 1$ for instances where $p_t = p$ becomes secondary. Assessing forecasts based solely on the observed sequence is an instance of the weak prequential principle (Dawid and Vovk, 1999).

The implication of this philosophy in practice is that the binning scheme we use need not be statistically optimal in any sense, but only practically meaningful. So for instance, it may be natural to discretize probabilities of rain as 5%, 15%, 25%, etc. In this case, the fixed-width binning (1.17) with $B = 10$ is used. On the other hand, for an infrequent event such as occurrence of a rare disease in a patient, we may be more interested in calibration close to 0. Then we can (arbitrarily) define $B_1 = [0, 0.01), B_2 = [0.01, 0.05), B_3 = [0.05, 0.1), B_4 = [0.1, 1.0]$.

Once a binning scheme is fixed, we compute FP, PP, w on the given sequence as described in the previous subsection. Then we compute the CE or make reliability or validity plots. Again, the benefit of the prequential view is that the bias of CE estimators for population quantities is irrelevant. For instance, the biased ℓ_1-CE estimator $\sum w_b \left|\text{FP}_b - \text{MP}_b\right|$ is meaningful on its own without viewing it as an estimate of anything else.

1.4.3 Measuring miscalibration in the multiclass setting

We devote Chapter 5 to discussing multiclass calibration. In that chapter, we show that most popular notions of multiclass calibration reduce to one or more statements about binary calibration. Thus, to assess the multiclass notion, one can assess each of the individual binary calibration notions, using techniques described earlier in this section. For instance, multiclass calibration can be binarized as confidence calibration (Guo et al., 2017), so that the ℓ_1-CE discussed earlier becomes conf-CE, and the reliability diagram becomes a confidence reliability diagrams.

Chapter 2

Thesis overview

In this thesis, we pursued a principled study of post-hoc calibration of machine learning models. Our study took a "validity-first" perspective—first build a method/framework that guarantees calibration (making it valid), and then consider additional desirable properties such as accuracy. We showed validity guarantees under minimal assumptions about the data-generating procedure—specifically in the "distribution-free" setup (Chapter 3, Chapter 4, Chapter 5, Chapter 9) and the "adversarial" setup (Chapter 6, Chapter 7, Chapter 8).

Chapters 3–7 reproduce published papers (one chapter per paper) on post-hoc calibration. The following subsections provide a blurb on each paper that assumes knowledge of Chapter 1, but is otherwise self-contained. A sequential reading provides an overarching "story" of our work. Chapters 8 and 9 reproduce a couple of published papers on covariate-free online calibration and conformal prediction respectively. These topics are broadly connected to post-hoc calibration and were a source of understanding and inspiration. The blurb on these papers below also identifies a specific relationship to post-hoc calibration. Two open-source contributions arising from our work are described in Section 2.7. Chapter 10 concludes with a broad discussion and avenues for future work.

2.1 Distribution-free binary calibration

While a number of empirical papers on post-hoc calibration existed previously, we first formalized and studied a clean theoretical goal for post-hoc calibration, in the following work.

Chapter 3

Distribution-free binary classification: prediction sets, confidence intervals and calibration.
Chirag Gupta, Aleksandr Podkopaev, Aaditya Ramdas.
Advances in Neural Information Processing Systems (NeurIPS), 2020.

This goal was previewed in Definition 1.2. We also introduced a "distribution-free" version of this goal: learn a post-hoc mapping that is provably approximately calibrated, no matter how the data is distributed, as long as it is i.i.d.

We then related calibration to two other objects in uncertainty quantification, prediction sets and confidence sets, in the distribution-free framework. From this relationship, we deduced the main result of our paper, an impossibility result for calibration. Namely, we showed that no injective post-hoc calibration method, such as Platt scaling, can be distribution-free calibrated. This result necessitates the usage of binning methods such as histogram binning and isotonic regression to achieve distribution-free calibration.

This study inspired the following work where calibration guarantees are established for histogram binning, a post-hoc calibration method introduced in Section 1.2.2.

Chapter 4

Distribution-free calibration guarantees for histogram binning without sample splitting
Chirag Gupta, Aaditya Ramdas
International Conference on Machine Learning (ICML), 2021

Histogram binning was proposed without formal guarantees by Zadrozny and Elkan (2001), and has been shown to be practically competitive in a number of papers. When histogram binning is performed without sample splitting, the same calibration data is used to identify bin boundaries as well as estimate the bin biases. This “double-dipping” becomes a hurdle in proving calibration guarantees for histogram binning. We use a certain Markov property of order statistics to circumvent this issue. Our work shows that histogram binning is both practically competitive and theoretically valid for calibration. Based on our theory, we provide certain practical recommendations, that prove fruitful in our follow-up study on multiclass calibration.

2.2 Distribution-free multiclass calibration

As discussed in Section 1.3, there has been interest in binary relaxations of canonical multiclass calibration. In the following paper, we studied these relaxations.

Chapter 5

Top-label calibration and multiclass-to-binary reductions
Chirag Gupta, Aaditya Ramdas
International Conference on Learning Representations (ICLR), 2022

We introduced the notion of top-label calibration, which focuses on calibration of the predicted class. We showed how distribution-free top-label calibration is to be achieved using a modification of the binary histogram binning procedure discussed in the previous subsection. Empirically, top-label histogram binning gets close to state-of-the-art performance when used to calibrate deep-nets on the CIFAR-10 and CIFAR-100 datasets Table 5.2.

Our second contribution was a unification of various notions of multiclass calibration into a single multiclass-to-binary (M2B) framework. As a benefit of this unified view, for every M2B calibration notion, a separate post-hoc calibration method can be derived that first reduces the multiclass dataset to a number of binary datasets, and then applies binary calibration to each binary dataset. This methodology is shown to be theoretically valid and practically beneficial.

2.3 Online post-hoc calibration

In our distribution-free analysis, we make no assumptions other than that the data is i.i.d. However, even assuming that data follows a distribution is a simplification of the real-world. In the following work, we studied if calibration can be meaningfully achieved without assuming that the data comes from a distribution.

Chapter 6

Online Platt Scaling with Calibeating
Chirag Gupta, Aaditya Ramdas
International Conference on Machine Learning (ICML), 2023

We model data as an adversarial or arbitrary feature-outcome stream $(\mathbf{x}_t, y_t)_{t=1}^{\infty} \in (\mathcal{X} \times \{0, 1\})^{\infty}$. It is expected that $\mathbf{x}_t$ is statistically useful for predicting y_t, however, the joint distribution of $(\mathbf{x}_t, y_t)$ could change arbitrarily over time. Our solution is to combine post-hoc calibration with ideas from the rich field of forecast calibration (introduced in Section 1.1.2). We propose an online version of Platt scaling (Section 1.2.1), called OPS, and combine it with the forecast calibration technique of calibeating (Foster and Hart, 2023). Our work leads to two OPS+calibeating methods: one of them exhibits good performance and is guaranteed to be adversarially calibrated guarantees; the other one does not have guarantees but performs the best in our experiments, on a mixture of real-world and synthetic datasets. In particular, our methods are adaptive to distribution drifts or shifts (Quinonero-Candela et al., 2008).

2.4 An application to regression

Previous works in this thesis were focused on achieving calibration. The following is the most empirical work in this thesis, where we explored an application of our developed methods to real-world problems.

Chapter 7

Parity Calibration
Youngseog Chung, Aaron Rumack, Chirag Gupta
Conference on Uncertainty in Artificial Intelligence (UAI), 2023

Specifically, we were interested in applying online Platt scaling to a time-series task with non-stationary covariate-outcome distributions. While searching for such tasks, we realized that many interesting tasks were of real-valued forecasting, leading us to ask the question—what regression task can be solved using post-hoc binary calibration?

A natural suggestion is to make calibrated forecasts for the increase-decrease event in a timeseries; we called this "parity calibration". A rich literature on making calibrated forecasts for regression exists, but we show that existing regression forecasts are not parity calibrated (theoretically or practically). Our online Platt scaling method, when adapted to this problem, works well out-of-the-box.

2.5 A modification to the forecast calibration setup

In the following paper, we studied a slight modification of the (covariate-agnostic) forecast calibration setup introduced in Chapter 1.1.2.

Chapter 8

Faster online calibration without randomization: interval forecasts and the power of two choices
Chirag Gupta, Aaditya Ramdas
Conference on Learning Theory (COLT), 2022

First, we showed a lower bound for forecaster in Calibration-Game-II (Panel 2), which is an $\Omega(1/\sqrt{T})$-rate for a certain notion of calibration error. Then, we studied a slight change to the setup: the forecaster is allowed to make two nearby probabilistic forecasts, or equivalently an interval forecast of small width, and the endpoint closest to the revealed outcome is used to judge calibration. This "power-of-two-choices" accords the forecaster with significant power—we show that a faster calibration rate of $O(1/T)$ can be achieved.

Relationship to post-hoc calibration. Due to our work in Chapter 6, we now know how to use forecast calibration techniques for post-hoc calibration. Applying the power-of-two-choices framework for post-hoc calibration is of potential interest.

2.6 Nested conformal prediction and data-efficient uncertainty quantification

Another form of distribution-free uncertainty quantification that has been popular recently is conformal prediction, where one is interested in prediction intervals (see Chapter 3). This is the topic of the following work.

Chapter 9

Nested conformal prediction and quantile out-of-bag ensemble methods
Chirag Gupta, Arun Kumar Kuchibhotla, Aaditya Ramdas
Pattern Recognition 127 (Special Issue on Conformal Prediction), 2022

In this work, we proposed an alternative formulation of conformal, called nested conformal, that (in our view) is more natural than the traditional description. This work appeared on arXiv in 2019, and since then, a number of papers have utilized the nested view when discuss conformal prediction. Further, we proposed an efficient conformal prediction method based on quantile prediction and out-of-bag ensembling. At the time of writing, our method was the state-of-the-art method for conformal prediction.

Relationship to post-hoc calibration. A common technique in conformal prediction, split-conformal (Papadopoulos et al., 2002), is similar to post-hoc calibration in that it relies on a held-out dataset. Data efficient methods have been derived for conformal prediction that do not involve sample splitting, such as cross-conformal (Vovk, 2015), jackknife+ (Barber et al., 2021), out-of-bag methods (Johansson et al., 2014), and online conformal prediction (Gibbs and Candes, 2021). Our online Platt scaling method does online post-hoc calibration, but cross-validation and out-of-bag approaches for calibration are yet to be discovered.

2.7 Open-source contributions

We released an open-source implementation of binary and top-label histogram binning (described in Chapter 4 and Chapter 5) at the following link: https://github.com/AIgen/df-posthoc-calibration/.

We released an open-source version of our quantile out-of-bag conformal method (described in Chapter 9) at the following link: https://github.com/AIgen/QOOB.

Part II

Original contributions (primary)

This part collates novel research contributions made by the author on the main topic of this thesis, post-hoc calibration. Each chapter reproduces a separate published paper.

Chapter 3

Distribution-free binary classification: prediction sets, confidence intervals, and calibration

This chapter is based on Gupta et al. (2020).

We study three notions of uncertainty quantification—calibration, confidence intervals and prediction sets—for binary classification in the distribution-free setting, that is without making any distributional assumptions on the data. With a focus towards calibration, we establish a 'tripod' of theorems that connect these three notions for score-based classifiers. A direct implication is that distribution-free calibration is only possible, even asymptotically, using a scoring function whose level sets partition the feature space into at most countably many sets. Parametric calibration schemes such as variants of Platt scaling do not satisfy this requirement, while nonparametric schemes based on binning do. To close the loop, we derive distribution-free confidence intervals for binned probabilities for both fixed-width and uniform-mass binning. As a consequence of our 'tripod' theorems, these confidence intervals for binned probabilities lead to distribution-free calibration. We also derive extensions to settings with streaming data and covariate shift.

3.1 Introduction

Let $\mathcal{X}$ and $\mathcal{Y} = \{0, 1\}$ denote the feature and label spaces for binary classification. Consider a predictor $f : \mathcal{X} \to \mathcal{Z}$ that produces a prediction in some space $\mathcal{Z}$. If $\mathcal{Z} = \{0, 1\}$, f corresponds to a point prediction for the class label, but often class predictions are based on a 'scoring function'. Examples are, $\mathcal{Z} = \mathbb{R}$ for SVMs, and $\mathcal{Z} = [0, 1]$ for logistic regression, random forests with class probabilities, or deep models with a softmax top layer. In such cases, a higher value of $f(X)$ is often interpreted as higher belief that $Y = 1$. In particular, if $\mathcal{Z} = [0, 1]$, it is tempting to interpret $f(X)$ as a probability, and hope that

$$f(X) \approx \mathbb{P}(Y = 1 \mid X) = \mathbb{E}\left[Y \mid X\right]. \tag{3.1}$$

However, such hope is unfounded, and in general (3.1) will be far from true without strong distributional assumptions, which may not hold in practice. Valid uncertainty estimates that are related to (3.1) can be provided, but ML models do not satisfy these out of the box. This chapter discusses three notions of uncertainty quantification: calibration, prediction sets (PS) and confidence intervals (CI), defined next. A function $f : \mathcal{X} \to [0, 1]$ is said to be (perfectly) calibrated if

$$\mathbb{E}\left[Y \mid f(X) = a\right] = a \qquad \text{a.s. for all } a \text{ in the range of } f. \tag{3.2}$$

Define the set of all subsets of $\mathcal{Y}$, $\mathcal{L} \equiv \{\{0\}, \{1\}, \{0, 1\}, \varnothing\}$, and fix $\alpha \in (0, 1)$. A function $S : \mathcal{X} \to \mathcal{L}$ is a $(1 - \alpha)$-PS if

$$\mathbb{P}(Y \in S(X)) \geqslant 1 - \alpha. \tag{3.3}$$

In practice, PSs are typically studied for larger output sets, such as $\mathcal{Y}_{\text{regression}} = \mathbb{R}$ or $\mathcal{Y}_{\text{multiclass}} = \{1, 2, \ldots, L > 2\}$, but in this chapter, we pursue fundamental results for binary classification. Finally, let $\mathcal{I}$ denote the set of all subintervals of $[0, 1]$. A function $C : \mathcal{X} \to \mathcal{I}$ is a $(1 - \alpha)$-CI if

$$\mathbb{P}(\mathbb{E}\left[Y \mid X\right] \in C(X)) \geqslant 1 - \alpha. \tag{3.4}$$

All three notions are 'natural' in their own sense, but also different at first sight. We show that they are in fact tightly connected (see Figure 3.1), and focus on the implications of this result for calibration. Most of our results are in the distribution-free setting, where we are concerned with understanding what uncertainty quantification is possible without making distributional assumptions on the data. This chapter is based on the statistical setup of post-hoc uncertainty quantification, described next.

Post-hoc uncertainty quantification setup. Let P denote the data-generating distribution over $\mathcal{X} \times \mathcal{Y}$, and let $(X, Y) \sim P$ denote a general data point. Post-hoc uncertainty quantification is a common paradigm where the available labeled data is split into a *training set* and a *calibration set*. The training set is used to learn a predictor $f : \mathcal{X} \to [0, 1]$, and the calibration set is used to supplement f with uncertainty estimates (CIs or PSs), or learn a new calibrated predictor on top of f. (In practice, the validation set is often used as the calibration set.) All results in this chapter are conditional on the training set; thus the randomness is always over the calibration and test data. We denote the calibration set as $\mathcal{D}_n = \{(X_i, Y_i)\}_{i \in [n]}$, where n is the number of calibration points, and we use the shorthand $[n] := \{1, 2, \ldots n\}$. A prototypical test point is denoted as (X_{n+1}, Y_{n+1}). The calibration and test data is assumed to be drawn i.i.d. from P, denoted succinctly as $\{(X_i, Y_i)\}_{i \in [n+1]} \sim P^{n+1}$. The learner observes realized values of all random variables (X_i, Y_i), except Y_{n+1}. All sets and functions are implicitly assumed to be measurable.

Our work relies on some key ideas in the works of Vovk et al. (2005a, Section 5), Barber (2020), and Zadrozny and Elkan (2001). Other related work is cited as needed, and further discussed in Section 3.5. All proofs appear ordered in the Appendix.

3.2 Calibration, confidence intervals and prediction sets

A few additional concepts and definitions are needed in order to formally study calibration, CIs and PSs in the distribution-free post-hoc uncertainty quantification setup. These are defined next.

3.2.1 Approximate and asymptotic calibration

Calibration captures the intuition of (3.1) but is a weaker requirement, and was first studied in the meteorological literature for assessing probabilistic rain forecasts (Brier, 1950; Sanders, 1963; Murphy and Epstein, 1967; Dawid, 1982). Murphy and Epstein (1967) described the ideal notion of calibration, called *perfect calibration* (3.2), which has also been referred to as *calibration in the small* (Vovk and Petej, 2014), or sometimes simply as *calibration* (Guo et al., 2017; Vaicenavicius et al., 2019; Dawid, 1982). The types of functions that can achieve perfect calibration can be succinctly captured as follows.

Proposition 3.1. *A function $f : \mathcal{X} \to [0,1]$ is perfectly calibrated if and only if there exists a space $\mathcal{Z}$ and a function $g : \mathcal{X} \to \mathcal{Z}$, such that*

$$f(x) = \mathbb{E}\left[Y \mid g(X) = g(x)\right] \quad \textit{almost surely } P_X. \tag{3.5}$$

In other words, f is calibrated if and only if there exists another function g such that f is the expected value of Y given the output of g. Vaicenavicius et al. (2019) stated and gave a short proof for the 'only if' direction. While the other direction is also straightforward, together they lead to an appealingly simple and complete characterization. The proof of Proposition 3.1 is in Appendix 3.A.

It is helpful to consider two extreme cases of Proposition 3.1. First, setting g to be the identity function yields that the Bayes classifier $\mathbb{E}\left[Y|X\right]$ is perfectly calibrated. Second, setting $g(\cdot)$ to any constant implies that $\mathbb{E}\left[Y\right]$ is also a perfect calibrator. Naturally, we cannot hope to estimate the Bayes classifier without assumptions, but even the simplest calibrator $\mathbb{E}\left[Y\right]$ can only be approximated in finite samples. Since Proposition 3.1 states that calibration is possible iff the RHS of (3.5) is known exactly for some g, perfect calibration is impossible in practice. Thus we resort to satisfying the requirement (3.2) approximately, which is implicitly the goal of many empirical calibration techniques.

Definition 3.1 (Approximate calibration)**.** A predictor $f : \mathcal{X} \to [0,1]$ is (ϵ, α)-calibrated for some $\epsilon, \alpha \in [0,1]$ if with probability at least $1-\alpha$,

$$\left|\mathbb{E}\left[Y|f(X)\right] - f(X)\right| \leqslant \epsilon. \tag{3.6}$$

Clearly, every predictor f is $(1,0)$-calibrated and $(0,1)$-calibrated. Further, if f is (ϵ,α)-calibrated, then it is also (ϵ',α)-calibrated for $\epsilon' > \epsilon$ and (ϵ,α')-calibrated for $\alpha' > \alpha$, and so we are typically only interested in the smallest "pareto optimal boundary" pairs of (ϵ,α) for which approximate calibration holds, or specifically for a fixed α like 0.1, what is the smallest ϵ for which calibration holds.

Suppose f is not approximately calibrated for small values of ϵ and α. As mentioned in the Introduction, we can 'recalibrate' f using a post-hoc calibration algorithm $\mathcal{A}$. Such an $\mathcal{A}$ takes f

(learnt on the training data) as input along with independent calibration data $\mathcal{D}_n = \{(X_i, Y_i)\}_{i\in[n]}$, and outputs $\mathcal{A}(\mathcal{D}_n, f) = h_n : \mathcal{X} \to [0,1]$, a predictor with presumably improved calibration properties compared to the original f. This setup was popularized by Guo et al. (2017); in their work, f is a deep neural network and a proposed algorithm $\mathcal{A}$ is temperature scaling. In this chapter, we study when $\mathcal{A}$ can be shown to satisfy *distribution-free approximate calibration*:

$$P^{n+1}(|\mathbb{E}\left[Y | h_n(X_{n+1})\right] - h_n(X_{n+1})| \leqslant \epsilon) \geqslant 1 - \alpha \quad \text{for every } f, P. \tag{3.7}$$

Above, P^{n+1} denotes the product distribution of the i.i.d. calibration and test points, that is $\{(X_i, Y_i)\}_{i\in[n+1]} \sim P^{n+1}$. Note that $h_n = \mathcal{A}(\mathcal{D}_n, f)$ is random over the calibration data $\mathcal{D}_n$; we reinforce this by writing an n in the subscript. In the limit of infinite calibration data, a good calibration algorithm should guarantee approximate calibration with vanishing ϵ. This is formalized in the upcoming definition of asymptotic calibration. We use $(\mathcal{X} \times \mathcal{Y})^* = \bigcup_{n\in\mathbb{N}}(\mathcal{X} \times \mathcal{Y})^n$ to denote the space of the calibration data for arbitrary n, and $[0,1]^{\mathcal{X}}$ to denote a function from $\mathcal{X}$ to $[0,1]$ (such as f).

Definition 3.2 (Distribution-free asymptotic calibration)**.** A post-hoc calibration algorithm $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \times [0,1]^{\mathcal{X}} \to [0,1]^{\mathcal{X}}$ is said to be distribution-free asymptotically calibrated if there exists an $\alpha \in (0, 0.5)$ and a $[0,1]$-valued sequence $\{\epsilon_n\}_{n\in\mathbb{N}}$ with $\lim_{n\to\infty} \epsilon_n = 0$, such that for every n, $h_n = \mathcal{A}(\mathcal{D}_n, f)$ satisfies condition (3.7) with parameters (ϵ_n, α).

Note that condition (3.7) requires approximate calibration not only over all P, but also over all f. Thus asymptotic calibration requires $\mathcal{A}$ to calibrate *any fixed* f over *all distributions* P.

3.2.2 Prediction sets and confidence intervals with respect to f

To motivate a new definition of PSs and CIs with respect to f, we review a recent result on distribution-free CIs by Barber (2020), where the existence of 'informative' distribution-free CIs was discussed.

PSs and CIs are only 'informative' if the sets or intervals produced by them are small. To quantify this, we measure CIs using their width (denoted as $|C(\cdot)|$), and PSs using their diameter (defined as the width of the convex hull of the PS). For example, in the case of binary classification, the diameter of a PS is 1 if the prediction set is $\{0,1\}$, and 0 otherwise (since $Y \in \{0,1\}$ always holds, the set $\{0,1\}$ is 'uninformative'). A short CI such as $[0.39, 0.41]$ is more informative than a wider one such as $[0.3, 0.5]$.

For a given distribution, one might expect the diameter of a $(1-\alpha)$-PS to be larger than the width of a $(1-\alpha)$-CI, since we want to cover the actual value of Y and not its conditional expectation. As an example, if $\mathbb{E}\left[Y | X = x\right] = 0.5$ for every x, then the shortest possible CI is $(0.5, 0.5]$ whose diameter is 0. However, a $(1-\alpha)$-PS has no choice but to output $\{0,1\}$ for at least $(1-2\alpha)$ fraction of the points (and a random guess for the other 2α fraction), and thus must have expected diameter $\geqslant 1 - 2\alpha$ even in the limit of infinite data.

Recently, Barber (2020) built on an earlier result of Vovk et al. (2005a) to show that if an algorithm provides $(1-\alpha)$-CI for all product distributions P^{n+1} (of the training data and test point), then it also provides a $(1-\alpha)$-PS whenever the distribution of P_X is nonatomic, that is, it does not contain any atoms or ‘point masses’. (If the CI function is $C : \mathcal{X} \to \mathcal{I}$, then the corresponding PS function would be $S(\cdot) = C(\cdot) \cap \{0, 1\}$.) Since this implication holds for all nonatomic distributions P_X, including the ones with $\mathbb{E}[Y|X] \equiv 0.5$ discussed above, it implies that distribution-free CIs must necessarily be wide. Specifically, their widths cannot shrink to 0 as $n \to \infty$. This can be treated as an impossibility result for the existence of informative distribution-free CIs.

One way to circumvent the above impossibility result is to consider CIs at a ‘coarser resolution’. We introduce the notion of a CI or PS ‘with respect to a function f’ (w.r.t. f).

Definition 3.3 (CI or PS w.r.t. f)**.** Fix a predictor $f : \mathcal{X} \to [0, 1]$ and let $(X, Y) \sim P$. A function $C : [0, 1] \to \mathcal{I}$ is said to be a $(1 - \alpha)$-CI with respect to f if

$$\mathbb{P}(\mathbb{E}[Y \mid f(X)] \in C(f(X))) \geqslant 1 - \alpha. \tag{3.8}$$

Analogously, a function $S : [0, 1] \to \mathcal{L}$ is a $(1 - \alpha)$-PS with respect to f if

$$\mathbb{P}(Y \in S(f(X))) \geqslant 1 - \alpha. \tag{3.9}$$

These definitions can be extended in a natural way if Range$(f) \neq [0, 1]$, as we do in the published version of this chapter (Gupta et al., 2020). If f is injective (one-to-one), then (3.8) and (3.9) reduce to (3.4) and (3.3). The more interesting (and typical) case is when f is not injective. In this case, the level sets of f partition $\mathcal{X}$ at a coarser ‘resolution’: $\mathcal{X} = \cup_{z \in [0,1]}\{x : f(x) = z\}$, and we can ask the (easier) question of producing a single CI or PS with respect to every $z \in [0, 1]$, instead of every $x \in \mathcal{X}$.

Naturally, for (3.8) or (3.9) to hold, the functions C and S must depend on P. Similar to the post-hoc calibration setting, we ask if C or S can be *learnt* using independent calibration data $\mathcal{D}_n$ drawn from P. Let $\mathcal{C}$ denote an algorithm that produces a CI function using f and $\mathcal{D}_n$, $C_n = \mathcal{C}(\mathcal{D}_n, f) : [0, 1] \to \mathcal{I}$, where the notation C_n reinforces the dependence of the CI function on $\mathcal{D}_n$. Similarly, let $\mathcal{S}$ denote an algorithm that produces a PS function, $S_n = \mathcal{S}(\mathcal{D}_n, f) : [0, 1] \to \mathcal{L}$. Akin to distribution-free approximate calibration (3.7), we have the following definitions for distribution-free CIs and PSs. C_n is said to be a distribution-free CI w.r.t. a fixed f if

$$P^{n+1}(\mathbb{E}[Y_{n+1} \mid f(X_{n+1})] \in C_n(f(X_{n+1}))) \geqslant 1 - \alpha \quad \text{for every } P, \tag{3.10}$$

and S_n is said to be a distribution-free PS w.r.t. a fixed f if

$$P^{n+1}(Y_{n+1} \in S_n(f(X_{n+1}))) \geqslant 1 - \alpha \quad \text{for every } P. \tag{3.11}$$

Table 3.1 summarizes the notation introduced so far. In the rest of the chapter, whenever we refer to objects with an ‘n’ in the subscript such as h_n, C_n, S_n, they should be understood as the outputs of some algorithms $\mathcal{A}, \mathcal{C}, \mathcal{S}$ when supplied with input $\mathcal{D}_n$ and f.

3.2.3 When is distribution-free post-hoc uncertainty quantification possible?

Are distribution-free guarantees such as (3.7), (3.10), and (3.11) too restrictive, or can they be achieved? We show that the answer for calibration and CIs (roughly) depends on how ‘large’

Calibration data	$\mathcal{D}_n = \{(X_i, Y_i)\}_{i \in [n]}$
Test point	(X_{n+1}, Y_{n+1})
General data point	(X, Y)
Probability over i.i.d. calibration and test data	P, P^{n+1}
Predictor learnt on (a split of the) training data	$f : \mathcal{X} \to [0, 1]$
General functions with unspecified sources of randomness	f, C, S
Random functions of the calibration data $\mathcal{D}_n$	h_n, C_n, S_n

Table 3.1: Notation used in this chapter to study post-hoc uncertainty quantification.

the range of f is. The result of Barber (2020) implies that if f is injective—that is f maps unique elements to unique elements—then informative distribution-free CIs are impossible. On the other hand, if f maps all of $\mathcal{X}$ to a single element, a short interval around the empirical mean of the Y_i's achieves (3.10) since $\mathbb{E}\left[Y \mid f(X)\right] = \mathbb{E}\left[Y\right]$. In this work, we characterize the transition point between these two behaviors.

In Section 3.3, we extend the above impossibility result to all functions f whose range contains any sub-interval of $[0, 1]$, a condition satisfied by all parametric machine learning models. On the other hand, in Section 3.4 we propose algorithms that achieve distribution-free CIs for f with finite range. We also show a close relationship between approximate calibration and CIs w.r.t. f. Based on this relationship, the results for distribution-free CIs extend to distribution-free calibration, and vice-versa. Specifically, no parametric (post-hoc) calibration algorithm, such as Platt scaling (Platt, 1999) or temperature scaling (Guo et al., 2017), can be distribution-free calibrated. On the other hand, distribution-free calibration guarantees can be shown for the discrete binning method of histogram binning (Zadrozny and Elkan, 2001).

In contrast to CIs and calibration, it is well known that meaningful and informative distribution-free PSs can be produced for any f, using a technique known as split conformal prediction (Papadopoulos et al., 2002). The broader literature on (non-split) conformal prediction also deals with techniques that produce distribution-free PSs without fixing an f learnt on a separate split of the data (Vovk et al., 2005a; Gupta et al., 2022). We do not discuss algorithmic results for distribution-free PSs in this chapter and refer the reader to one of the aforementioned papers on conformal prediction.

3.3 Relating the notions of uncertainty quantification

The relationships between the notions of uncertainty quantification are summarized in Figure 3.1. In this figure, and in the rest of the section, we denote the distribution of the random variable $Z = f(X)$ as $P_{f(X)}$. In Section 3.3.1, we show that if an algorithm provides a CI w.r.t. f, it can be used to provide approximate calibration and vice-versa (Theorem 3.1). In Section 3.3.2, we show that if an algorithm constructs a distribution-free CI w.r.t. f, then the constructed CIs must also be PSs for a large class of distributions P for which $P_{f(X)}$ is nonatomic (Theorem 3.2).

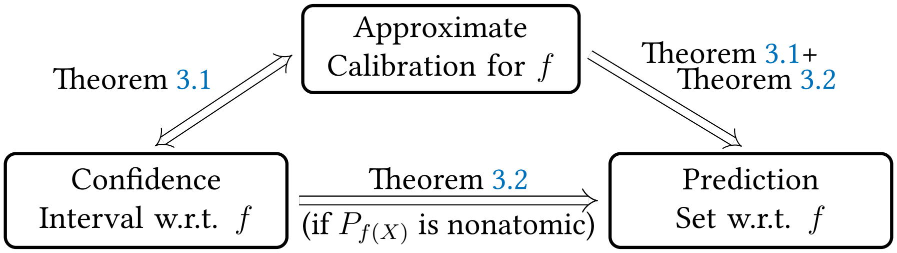

Approximate Calibration for f
Theorem 3.1
Theorem 3.1+ Theorem 3.2
Confidence Interval w.r.t. f
Theorem 3.2
(if $P_{f(X)}$ is nonatomic)
Prediction Set w.r.t. f

Figure 3.1: Relationship between notions of distribution-free uncertainty quantification.

Since we expect the width of CIs to be shorter than the diameter of PSs, this can be interpreted as an impossibility result for informative distribution-free CIs (Corollary 3.1). Merging these two results, in Section 3.3.3, we show that meaningful distribution-free calibration is not possible for certain scoring functions and post-hoc calibration algorithms (Theorem 3.3).

3.3.1 Relating calibration and confidence intervals

Suppose we are given a predictor $f : \mathcal{X} \to [0, 1]$ that is (ϵ, α)-calibrated. Then one can construct a function C that is a $(1 - \alpha)$-CI: for $x \in \mathcal{X}$,

$$\underbrace{|\mathbb{E}[Y \mid f(x)] - f(x)| \leqslant \epsilon}_{\text{calibration}} \implies \underbrace{\mathbb{E}[Y \mid f(x)] \in C(f(x))}_{\text{CI w.r.t. } f} := [f(x) - \epsilon, f(x) + \epsilon]. \tag{3.12}$$

On the other hand, given $C : [0, 1] \to \mathcal{I}$ that is a $(1 - \alpha)$-CI w.r.t. f, define for $z \in [0, 1]$, the left-endpoint, right-endpoint, and midpoint functions respectively:

$$u_C(z) := \sup\{t : t \in C(z)\}, \; l_C(z) := \inf\{t : t \in C(z)\}, \; m_C(z) := (u_C(z) + l_C(z))/2. \tag{3.13}$$

Consider the midpoint $m_C(f(x))$ as a 'corrected' prediction for $x \in \mathcal{X}$:

$$\widetilde{f}(x) := m_C(f(x)), \; x \in \mathcal{X}, \tag{3.14}$$

and let $\epsilon = \sup_{z \in \text{Range}(f)} \{|C(z)|/2\}$ be the largest interval radius. Then $\widetilde{f}$ is (ϵ, α)-calibrated. These claims are formalized next.

Theorem 3.1. *Fix any $\alpha \in (0, 1)$. Let $f : \mathcal{X} \to [0, 1]$ be a predictor that is (ϵ, α)-calibrated for some $\epsilon \in (0, 1)$. Then the function C in (3.12) is a $(1 - \alpha)$-CI with respect to f.*

Conversely, fix a scoring function $f : \mathcal{X} \to [0, 1]$. If C is a $(1 - \alpha)$-CI with respect to f, then the predictor $\widetilde{f}$ in (3.14) is (ϵ, α)-calibrated for $\epsilon = \sup_{z \in [0,1]} \{|C(z)|/2\}$.

The proof of the theorem is in Appendix 3.B. Note that Theorem 3.1 is not restricted to the post-hoc uncertainty quantification setting and the calibration and CI functions need not satisfy distribution-free guarantees as defined in (3.7) or (3.10). In contrast, the relationship between CIs and PSs stated in the following subsection is specific to the distribution-free setting.

3.3.2 Relating confidence intervals and prediction sets in the distribution-free setting

In this section, we relate CIs and PSs with respect to a fixed function $f : \mathcal{X} \to [0,1]$. Consider the following set of distributions, whose motivation becomes clearly shortly:

$$\mathcal{P}_f := \{\text{distributions } P \text{ over } \mathcal{X} \times \mathcal{Y} \ : P_{f(X)} \text{ is nonatomic}\}. \tag{3.15}$$

$P_{f(X)}$ being nonatomic means that the distribution of $f(X)$, when $(X,Y) \sim P$, contains no atoms or ‘point masses’. Suppose C_n satisfies (3.10), that is, it provides a CI guarantee w.r.t. f for all distributions P. We show that C_n can be used to provide a modified PS guarantee which is not distribution-free but holds for all $P \in \mathcal{P}_f$:

$$P^{n+1}(Y_{n+1} \in S_n(f(X_{n+1}))) \geqslant 1 - \alpha \quad \text{for every } P \in \mathcal{P}_f. \tag{3.16}$$

The following result is proved in Appendix 3.B.

Theorem 3.2. *Fix $f : \mathcal{X} \to [0,1]$ and $\alpha \in (0,1)$. If C_n is a distribution-free confidence interval with respect to f, as in (3.10), then $S_n(\cdot) = C_n(\cdot) \cap \{0,1\}$ is a $(1-\alpha)$-prediction set with respect to f for every $P \in \mathcal{P}_f$, as in (3.16).*

Above we transformed the CI function C_n to a PS function S_n by performing an intersection with the $\{0,1\}$. Based on the intuition discussed before Definition 3.3, Theorem 3.2 can be interpreted as an impossibility result for distribution-free valid CIs that are ‘informative’ for all distributions.

Corollary 3.1. *Fix $f : \mathcal{X} \to [0,1]$ and $\alpha \in (0,0.5)$. If C_n is a distribution-free confidence interval with respect to f (3.10), and $\mathcal{P}_f$ is non-empty, then there exists a distribution $P \in \mathcal{P}_f$ such that*

$$\mathbb{E}_{P^{n+1}}|C_n(f(X_{n+1}))| \geqslant 0.5 - \alpha.$$

Note that for every P, there exists a CI function with expected width equal to zero: $C_P(\cdot) = \{\mathbb{E}_P[Y \mid f(X) = \cdot]\}$. A desirable property for C_n is consistency: given enough samples from P, does C_n recover C_P? Corollary 3.1 shows that if $\mathcal{P}_f$ is non-empty, then no distribution-free CI function can be ‘distribution-free consistent’ for C_P — there exist $P \in \mathcal{P}_f$ for which the average width of the CI is lower bound by a constant independent of n.

Thus we would like to know when $\mathcal{P}_f$ is non-empty. First, note that if the range of f is countable, then for any P, $P_{f(X)}$ contains atoms (due to the subadditivity of measure, any distribution over a countable set must contain atoms). Thus $\mathcal{P}_f$ is empty and Corollary 3.1 does not apply. On the other hand, Lemma 3.1 in Appendix 3.B.5 shows that if the range of f is $[0,1]$ or contains any sub-interval of $[0,1]$, then $\mathcal{P}_f$ is non-empty (the proof relies on a technical probability theory result of Ershov (1975)). Thus Corollary 3.1 applies to all standard parametric machine learning models, whose range is usually $[0,1]$ or $(0,1)$. In the following subsection, we use Corollary 3.1 to show an impossibility result for certain post-hoc calibration algorithms.

3.3.3 Impossibility result for distribution-free post-hoc calibration

Proposition 3.1 shows that a function f is calibrated if and only if it takes the form (3.5) for some function g. Observe that g essentially provides a *partition* of $\mathcal{X}$ based on the level sets of g. Denote this partition as $\{\mathcal{X}_z\}_{z\in\mathcal{Z}}$, where $\mathcal{X}_z = \{x \in \mathcal{X} : g(x) = z\}$. Then we may equivalently define f in (3.5) through a set of values $\{f_z = P(Y = 1 \mid X \in \mathcal{X}_z)\}_{z\in\mathcal{Z}}$, setting $f(\cdot) = f_{g(\cdot)}$. In this sense, calibration can be viewed as a goal with two parts: (A) identify a 'meaningful' partition of $\mathcal{X}$ and (B) estimate the conditional probabilities for each partition.

Corollary 3.2 (to Proposition 3.1). *Any calibrated classifier f is characterized by an index set $\mathcal{Z}$,*

(A) a partition of $\mathcal{X}$ into subsets $\{\mathcal{X}_z\}_{z\in\mathcal{Z}}$, and

(B) corresponding conditional probabilities $\{f_z\}_{z\in\mathcal{Z}}$.

This interpretation motivates the underlying principle of post-hoc calibration. Existing ML techniques often implicitly do (A). They produce f that, while miscalibrated, may have some rough monotonicity with respect to the true probability: $f(x_1) \geqslant f(x_2) \iff \mathbb{P}(Y = 1 \mid X = x_1) \geqslant \mathbb{P}(Y = 1 \mid X = x_2)$ (see Zadrozny and Elkan (2002, Figures 1 and 2) for examples when such a hypothesis roughly holds on real data). In other words, the partitioning of $\mathcal{X}$ induced by the level sets of f, $\{\mathcal{X}_z = \{x : f(x) = z\}\}_{z\in[0,1]}$, is often informative, but $|z - \mathbb{P}(Y = 1 \mid X_{n+1} \in \mathcal{X}_z)|$ may be large. Post-hoc calibration techniques leverage the solution of (A) provided by f, and focus on (B); they use calibration data $\mathcal{D}_n$ to estimate $\mathbb{P}(Y = 1 \mid X \in \mathcal{X}_z)$ for every $z \in \text{Range}(f)$.

Thus a post-hoc calibration method 'recalibrates' f by mapping its output to a new value in $[0, 1]$. Let $h_n = \mathcal{A}(\mathcal{D}_n, f)$ be the output of a post-hoc calibration method $\mathcal{A}$ and let $m_n : [0, 1] \to [0, 1]$ be the implicit mapping function so that $h_n(x) = m_n(f(x))$. Consider three popular parametric algorithms for post-hoc calibration: Platt scaling (Platt, 1999), temperature scaling (Guo et al., 2017), and beta calibration (Kull et al., 2017). The mapping m_n learnt by each of these methods is strictly monotonic, and hence, injective (one-to-one).[1] Let us call these as 'injective' post-hoc calibration algorithms. We now state the impossibility result for distribution-free calibration.

Theorem 3.3. *It is impossible for an injective post-hoc calibration algorithm to be distribution-free asymptotically calibrated.*

The proof of Theorem 3.3 is in Appendix 3.B, but we briefly sketch its intuition below. Since the mapping m_n produced by $\mathcal{A}$ is injective, $\mathbb{E}[Y \mid h_n(X)] = \mathbb{E}[Y \mid m_n(f(X))] = \mathbb{E}[Y \mid f(X)]$. Thus a CI w.r.t. h_n is also a CI w.r.t. f. As a consequence, if h_n is distribution-free (ϵ_n, α)-calibrated, then by Theorem 3.1,

$$C_n(f(X)) := [h_n(X) - \epsilon_n, h_n(X) + \epsilon_n] = [m_n(f(X)) - \epsilon_n, m_n(f(X)) + \epsilon_n],$$

is a distribution-free $(1 - \alpha)$-CI w.r.t. f. Consider any standard parametric function f. As shown in Appendix 3.B.5, $\mathcal{P}_f$ is non-empty for such f. We can thus use Corollary 3.1 to conclude that the width of any distribution-free CI such as C_n must be lower bounded by $0.5 - \alpha$ (for all

[1] This assumes that the parameters satisfy natural constraints as discussed in the original papers: $a, b \geqslant 0$ for beta scaling with at least one of them nonzero, $A < 0$ for Platt scaling and $T > 0$ for temperature scaling.

n). Thus, $2\epsilon_n \geqslant 0.5 - \alpha$ for all n, which is a constant lower bound on ϵ_n (since $\alpha < 0.5$). We conclude that $\lim_{n\to\infty} \epsilon_n > 0$, and asymptotic calibration is impossible.

The implication of Theorem 3.3 is that injective algorithms such as Platt scaling, temperature scaling, and beta scaling cannot satisfy distribution-free calibration in any meaningful way. While all parameteric post-hoc calibration methods we are aware of are injective, we conjecture that a result like Theorem 3.3 holds even more generally for any parametric post-hoc calibration method, as long as its output is continuous.

Nonparametric calibration methods of isotonic regression (Zadrozny and Elkan, 2002) and histogram binning (Zadrozny and Elkan, 2001) are not injective, and thus can potentially satisfy distribution-free asymptotic calibration guarantees. In the following section, we analyze histogram binning and show that any scoring function can be 'binned' to achieve distribution-free calibration. We explicitly quantify the finite-sample approximate calibration guarantees that automatically also lead to asymptotic calibration. We also discuss calibration in the online setting and calibration under covariate shift.

3.4 Achieving distribution-free calibration

In Section 3.4.1, we prove a distribution-free approximate calibration guarantee given a fixed partitioning of the feature space into finitely many sets. This calibration guarantee also leads to distribution-free asymptotic calibration. In Section 3.4.2, we discuss a natural method for obtaining such a partition using sample-splitting, called histogram binning. Histogram binning inherits the bound in Section 3.4.1. This shows that binning schemes lead to distribution-free approximate calibration. In Section 3.4.3 and 3.4.4 we discuss extensions of this scheme for streaming data and covariate shift respectively.

3.4.1 Distribution-free calibration given a fixed sample-space partition

Suppose we have a fixed partition of $\mathcal{X}$ into B regions $\{\mathcal{X}_b\}_{b\in[B]}$, and let $\pi_b = \mathbb{E}\left[Y \mid X \in \mathcal{X}_b\right]$ be the expected label probability in region $\mathcal{X}_b$. Denote the partition-identity function as $\mathcal{B} : \mathcal{X} \to [B]$ where $\mathcal{B}(x) = b$ if and only if $x \in \mathcal{X}_b$. Given a calibration set $\{(X_i, Y_i)\}_{i\in[n]}$, let $N_b := |\{i \in [n] : \mathcal{B}(X_i) = b\}|$ be the number of points from the calibration set that belong to region $\mathcal{X}_b$. In this subsection, we assume that $N_b \geqslant 1$ (in Section 3.4.2 we show that the partition can be constructed to ensure that N_b is $\Omega(n/B)$ with high probability). Define

$$\widehat{\pi}_b := \frac{1}{N_b} \sum_{i:\mathcal{B}(X_i)=b} Y_i \qquad \text{and} \qquad \widehat{V}_b := \frac{1}{N_b} \sum_{i:\mathcal{B}(X_i)=b} (Y_i - \widehat{\pi}_b)^2 \tag{3.17}$$

as the empirical average and variance of the Y values in a partition. We now deploy an empirical Bernstein bound (Audibert et al., 2007) to produce a confidence interval for π_b.

Theorem 3.4. *For any $\alpha \in (0,1)$, with probability at least $1-\alpha$,*

$$|\pi_b - \widehat{\pi}_b| \leqslant \sqrt{\frac{2\widehat{V}_b \ln(3B/\alpha)}{N_b}} + \frac{3\ln(3B/\alpha)}{N_b}, \quad \textit{simultaneously for all } b \in [B].$$

The theorem is proved in Appendix 3.C. Using the crude deterministic bound $\widehat{V}_b \leqslant 1$ we get that the width of the confidence interval for partition b is $O(1/\sqrt{N_b})$. However, if for some b, $\mathcal{X}_b$ is highly informative or homogeneous in the sense that π_b is close to 0 or 1, we expect $\widehat{V}_b \ll 1$. In this case, Theorem 3.4 *adapts* and provides an $O(1/N_b)$ width confidence interval for π_b. Let $b^\star = \arg\min_{b\in[B]} N_b$ denote the index of the region with the minimum number of calibration examples.

Corollary 3.3. *For $\alpha \in (0,1)$, the function $h_n(\cdot) := \widehat{\pi}_{\mathcal{B}(\cdot)}$ is distribution-free (ϵ, α)-calibrated with*

$$\epsilon = \sqrt{\frac{2\widehat{V}_{b^\star} \ln(3B/\alpha)}{N_{b^\star}}} + \frac{3\ln(3B/\alpha)}{N_{b^\star}}.$$

Thus, $\{h_n\}_{n\in\mathbb{N}}$ is distribution-free asymptotically calibrated for any α.

The proof is in Appendix 3.C. Thus, any finite partition of $\mathcal{X}$ leads to asymptotic calibration. However, the finite sample guarantee of Corollary 3.3 can be unsatisfactory if the sample-space partition is chosen poorly, since it might lead to small $N_{b^\star}$. In Section 3.4.2, we present a data-dependent partitioning scheme that provably guarantees that $N_{b^\star}$ scales as $\Omega(n/B)$ with high probability. The calibration guarantee of Corollary 3.3 can also be stated conditional on a given test point:

$$|\mathbb{E}[Y \mid f(X)] - f(X)| \leqslant \epsilon, \text{ almost surely } P_X . \tag{3.18}$$

This holds since Theorem 3.4 provides simultaneously valid CIs for all regions $\mathcal{X}_b$.

3.4.2 Identifying a data-dependent partition using sample splitting

Here, we describe ways of constructing the partition $\{\mathcal{X}_b\}_{b\in[B]}$ through fixed-width binning. Binning uses a sample splitting strategy to learn the partition of $\mathcal{X}$ as described in Section 3.4.1. A split of the data is used to learn the partition and an independent split is used to estimate $\{\widehat{\pi}_b\}_{b\in[B]}$. Formally, the labeled data is split at random into a training set $\mathcal{D}_{\text{tr}}$ and a calibration set $\mathcal{D}_{\text{cal}}$. Then $\mathcal{D}_{\text{tr}}$ is used to train a scoring function $g : \mathcal{X} \to [0,1]$ (in general the range of g could be any interval of $\mathbb{R}$ but for simplicity we describe it for $[0,1]$). The scoring function g usually does not satisfy a calibration guarantee out-of-the-box but can be calibrated using binning.

A *binning scheme* $\mathcal{B}$ is any partition of $[0,1]$ into B non-overlapping intervals $I_1, \ldots, I_B$, such that $\bigcup_{b\in[B]} I_b = [0,1]$ and $I_b \cap I_{b'} = \varnothing$ for $b \neq b'$. $\mathcal{B}$ and g induce a partition of $\mathcal{X}$ as follows:

$$\mathcal{X}_b = \{x \in \mathcal{X} : \ g(x) \in I_b\}, \ b \in [B]. \tag{3.19}$$

The simplest binning scheme corresponds to *fixed-width binning*. In this case, bins have the form

$$I_i = \left[\frac{i-1}{B}, \frac{i}{B}\right), i = 1, \ldots, B-1 \text{ and } I_B = \left[\frac{B-1}{B}, 1\right].$$

However, fixed-width binning suffers from the drawback that there may exist bins with very few calibration points (low N_b), while other bins may get many calibration points. For bins with low N_b, the $\widehat{\pi}_b$ estimates cannot be guaranteed to be well calibrated, since the bound of Theorem 3.4 could be large. To remedy this, we consider *uniform-mass binning*, which aims to guarantee that each region $\mathcal{X}_b$ contains approximately equal number of calibration points. This is done by estimating the empirical quantiles of $g(X)$. First, the calibration set $\mathcal{D}_{\text{cal}}$ is randomly split into two parts, $\mathcal{D}_{\text{cal}}^1$ and $\mathcal{D}_{\text{cal}}^2$. For $j \in [B-1]$, the (j/B)-th quantile of $g(X)$ is estimated from $\{g(X_i), i \in \mathcal{D}_{\text{cal}}^1\}$. Let us denote the empirical quantile estimates as $\widehat{q}_j$. Then, the bins are defined as:

$$I_1 = [0, \widehat{q}_1), I_i = [\widehat{q}_{i-1}, \widehat{q}_i), i = 2, \ldots, B-1 \text{ and } I_B = (\widehat{q}_{B-1}, 1].$$

This induces a partition of $\mathcal{X}$ as per (3.19). Now, only $\mathcal{D}_{\text{cal}}^2$ is used for calibrating the underlying classifier, as per the calibration scheme defined in Section 3.4.1. Kumar et al. (2019) showed that uniform-mass binning provably controls the number of calibration samples that fall into each bin (see Appendix 3.F.2). Building on their result and Corollary 3.3, we show the following guarantee.

Theorem 3.5. *Fix $g : \mathcal{X} \to [0,1]$ and $\alpha \in (0,1)$. There exists a universal constant c such that if $|\mathcal{D}_{cal}^1| \geqslant cB\ln(2B/\alpha)$, then with probability at least $1-\alpha$,*

$$N_{b^\star} \geqslant \left|\mathcal{D}_{cal}^2\right|/2B - \sqrt{|\mathcal{D}_{cal}^2|\ln(2B/\alpha)/2}.$$

Thus even if $|\mathcal{D}_{cal}^1|$ does not grow with n, as long as $|\mathcal{D}_{cal}^2| = \Omega(n)$, uniform-mass binning is distribution-free $(\widetilde{O}(\sqrt{B\ln(1/\alpha)/n}), \alpha)$-calibrated, and hence distribution-free asymptotically calibrated for any α.

The proof is in Appendix 3.C. In words, if we use a small number of points (independent of n) for uniform-mass binning, and the rest to estimate bin probabilities, we achieve approximate/asymptotic distribution-free calibration. Note that the probability is conditional on a fixed predictor g, and hence also conditional on the training data $\mathcal{D}_{\text{tr}}$. Since Theorem 3.5 uses Corollary 3.3, the calibration guarantee can also be stated conditionally on a fixed test point, akin to equation (3.18).

3.4.3 Distribution-free calibration in the online setting

So far, we have considered the batch setting with a fixed calibration set of size n. However, often a practitioner might want to query additional calibration data until a desired confidence level is achieved. This is called the *online* or *streaming* setting. In this case, the results of Section 3.4 are no longer valid since the number of calibration samples is unknown a priori and may even be dependent on the data. In order to quantify uncertainty in the online setting, we use *time-uniform* concentration bounds (Howard et al., 2021; Howard et al., 2020); these hold simultaneously for all possible values of the calibration set size $n \in \mathbb{N}$.

Fix a partition of $\mathcal{X}$, $\{\mathcal{X}_b\}_{b \in [B]}$. For some value of n, let the calibration data be given as $\mathcal{D}_{\text{cal}}^{(n)}$. We use the superscript notation to emphasize the dependence on the current size of the calibration set. Let $\{(X_i^b, Y_i^b)\}_{i \in [N_b^{(n)}]}$ be examples from the calibration set that fall into the partition $\mathcal{X}_b$, where $N_b^{(n)} := |\{i \in [n] : \mathcal{B}(X_i) = b\}|$ is the total number of points that are mapped to $\mathcal{X}_b$. Let the empirical label average and cumulative (unnormalized) empirical variance be denoted as

$$\widehat{V}_b^+ = 1 \vee \sum_{i=1}^{N_b^{(n)}} \left(Y_i^b - \overline{Y}_{i-1}^b\right)^2, \text{ where } \overline{Y}_i^b := \frac{1}{i}\sum_{j=1}^{i} Y_j^b \text{ for } i \in [N_b^{(n)}]. \tag{3.20}$$

Note the normalization difference between $\widehat{V}_b^+$ and $\widehat{V}^b$ used in the batch setting. The following theorem constructs confidence intervals for $\{\pi_b\}_{b \in [B]}$ that are valid uniformly for any value of n.

Theorem 3.6. *For any $\alpha \in (0, 1)$, with probability at least $1 - \alpha$,*

$$|\pi_b - \widehat{\pi}_b| \leqslant \frac{7\sqrt{\widehat{V}_b^+ \ln\left(1 + \ln \widehat{V}_b^+\right) + 5.3 \ln\left(\frac{6.3B}{\alpha}\right)}}{N_b^{(n)}}, \quad \textit{simultaneously for all } b \in [B] \textit{ and all } n \in \mathbb{N}. \tag{3.21}$$

The proof is in Appendix 3.C. Due to the crude bound: $\widehat{V}_b^+ \leqslant N_b^{(n)}$, we can see that the width of confidence intervals roughly scales as $O(\sqrt{\ln(1+\ln N_b^{(n)})/N_b^{(n)}})$. In comparison to the batch setting, only a small price is paid for not knowing beforehand how many examples will be used for calibration.

3.4.4 Calibration under covariate shift

Here, we briefly consider the problem of calibration under covariate shift (Shimodaira, 2000). In this setting, calibration data $\{(X_i, Y_i)\}_{i \in [n]} \sim P^n$ is from a 'source' distribution P, while the test point is from a shifted 'target' distribution $(X_{n+1}, Y_{n+1}) \sim \widetilde{P} = \widetilde{P}_X \times P_{Y|X}$, meaning that the 'shift' occurs only in the covariate distribution while $P_{Y|X}$ does not change. We assume the likelihood ratio (LR)

$$w : \mathcal{X} \to \mathbb{R}; \quad w(x) := \mathrm{d}\widetilde{P}_X(x)/\mathrm{d}P_X(x)$$

is well-defined. The following is unambiguous: *if w is arbitrarily ill-behaved and unknown, the covariate shift problem is hopeless, and one should not expect any distribution-free guarantees.* Nevertheless, one can still make nontrivial claims using a 'modular' approach towards assumptions:

Condition (A): $w(x)$ is known exactly and is bounded.

Condition (B): an asymptotically consistent estimator $\widehat{w}(x)$ for $w(x)$ can be constructed.

We show the following: under Condition (A), a weighted estimator using w delivers approximate and asymptotic distribution-free calibration; under Condition (B), weighting with a plug-in estimator for w continues to deliver asymptotic distribution-free calibration. It is clear that Condition (B) will always require distributional assumptions: asymptotic consistency is nontrivial for ill-behaved w. Nevertheless, the above two-step approach makes it clear where the burden of assumptions lie: not with calibration step, but with the w estimation step. Estimation of w

is a well studied problem in the covariate-shift literature and there is some understanding of what assumptions are needed to accomplish it, but there has been less work on recognizing the resulting implications for calibration. Luckily, many practical methods exist for estimating w given unlabeled samples from $\widetilde{P}_X$ (Bickel et al., 2007; Huang et al., 2007; Kanamori et al., 2009). In summary, if Condition (B) is possible, then distribution-free calibration is realizable, and if Condition (B) is not met (even with infinite samples), then it implies that w is probably very ill-behaved, and so distribution-free calibration is also likely to be impossible.

For a fixed partition $\{\mathcal{X}_b\}_{b \in [B]}$, one can use the labeled data from the source distribution to estimate $\mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right]$ (unlike $\mathbb{E}_P\left[Y \mid X \in \mathcal{X}_b\right]$ as before), given oracle access to w:

$$\check{\pi}_b^{(w)} := \frac{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)Y_i}{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)}. \tag{3.22}$$

As preluded to earlier, assume that

$$\text{for all } x \in \mathcal{X},\ L \leqslant w(x) \leqslant U \text{ for some } 0 < L \leqslant 1 \leqslant U < \infty. \tag{3.23}$$

The 'standard' i.i.d. assumption on the test point equivalently assumes w is known and $L = U = 1$. We now present our first claim: $\check{\pi}_b^{(w)}$ satisfies a distribution-free approximate calibration guarantee. To show the result, we assume that the sample-space partition was constructed via uniform-mass binning (on the source domain) with sufficiently many points, as required by Theorem 3.5. This guarantees that all regions satisfy $|\{i : \mathcal{B}(X_i) = b\}| = \Omega(n/B)$ with high probability.

Theorem 3.7. *Assume w is known and bounded (3.23). Then for an explicit universal constant $c > 0$, with probability at least $1 - \alpha$,*

$$\left|\check{\pi}_b^{(w)} - \mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right]\right| \leqslant c\left(\frac{U}{L}\right)^2 \sqrt{\frac{B\ln(6B/\alpha)}{2n}}, \quad \textit{simultaneously for all } b \in [B],$$

as long as $n \geqslant c(U/L)^2 B \ln^2(6B/\alpha)$. Thus $h_n(\cdot) = \check{\pi}_{\mathcal{B}(\cdot)}^{(w)}$ is distribution-free asymptotically calibrated for any α.

The proof is in Appendix 3.D. Theorem 3.7 establishes distribution-free calibration under Condition (A). For Condition (B), using k *unlabeled* samples from the source and target domains, assume that we construct an estimator $\widehat{w}_k$ of w that is consistent, meaning

$$\sup_{x \in \mathcal{X}} |\widehat{w}_k(x) - w(x)| \xrightarrow{P} 0. \tag{3.24}$$

We now define an estimator $\check{\pi}_b^{(\widehat{w}_k)}$ by plugging in $\widehat{w}_k$ for w in the right hand side of (3.22):

$$\check{\pi}_b^{(\widehat{w}_k)} := \frac{\sum_{i:\mathcal{B}(X_i)=b} \widehat{w}_k(X_i)Y_i}{\sum_{i:\mathcal{B}(X_i)=b} \widehat{w}_k(X_i)}.$$

Proposition 3.2. *If $\widehat{w}_k$ is consistent (3.24), then $h_n(\cdot) = \check{\pi}_{\mathcal{B}(\cdot)}^{(\widehat{w}_k)}$ is distribution-free asymptotically calibrated for any $\alpha \in (0, 0.5)$.*

In Appendix 3.D, we illustrate through preliminary simulations that w can be estimated using unlabeled data from the target distribution, and consequently approximate calibration can be achieved on the target domain. Recently, Park et al. (2020) also considered calibration under covariate shift through importance weighting, but they do not show validity guarantees in the same sense as Theorem 3.7. For real-valued regression, distribution-free prediction sets under covariate shift were constructed using conformal prediction (Tibshirani et al., 2019) under Condition (A), and is thus a precursor to our modular approach.

3.5 Other related work

The problem of assessing the calibration of binary classifiers was first studied in the meteorological and statistics literature (Brier, 1950; Sanders, 1963; Murphy and Epstein, 1967; Murphy, 1972a; Murphy, 1972b; Murphy, 1973; Dawid, 1982; DeGroot and Fienberg, 1983; Bröcker, 2012; Ferro and Fricker, 2012); we refer the reader to the review by Dawid (2014) for more details. These works resulted in two common ways of measuring calibration: reliability diagrams (DeGroot and Fienberg, 1983) and estimates of the squared expected calibration error (ECE) (Sanders, 1963): $\mathbb{E}(f(X) - \mathbb{E}\left[Y \mid f(X)\right])^2$. Squared ECE can easily be generalized to multiclass settings and some related notions such as absolute deviation ECE and top-label ECE have also been considered, for instance (Guo et al., 2017; Naeini et al., 2015). ECE is typically estimated through binning, which provably leads to underestimation of ECE for calibrators with continuous output (Vaicenavicius et al., 2019; Kumar et al., 2019). Certain methods have been proposed to estimate ECE without binning (Zhang et al., 2020; Widmann et al., 2019), but they require distributional assumptions for provability.

While these papers have focused on the difficulty of *estimating* calibration error, ours is the first formal impossibility result for *achieving* calibration. In particular, Kumar et al. (2019, Theorem 4.1) show that the scaling-binning procedure achieves calibration error close to the best within a fixed, regular, injective parametric class. However, as discussed in Section 3.3.3 (after Theorem 3.3), we show that the best predictor in such an injective parametric class is itself not distribution-free calibrated. In summary, our results show not only that (some form of) binning is necessary for distribution-free calibration (Theorem 3.3), but also sufficient (Corollary 3.3).

Apart from classical methods for calibration (Platt, 1999; Zadrozny and Elkan, 2001; Zadrozny and Elkan, 2002; Niculescu-Mizil and Caruana, 2005), some new methods have been proposed recently, primarily for calibration of deep neural networks (Lakshminarayanan et al., 2017; Guo et al., 2017; Kumar et al., 2018; Tran et al., 2019; Seo et al., 2019; Kuleshov et al., 2018; Kendall and Gal, 2017; Wenger et al., 2020; Milios et al., 2018). These calibration methods perform well in practice but do not have distribution-free guarantees. A calibration framework that generalizes binning and isotonic regression is Venn prediction (Vovk et al., 2003; Vovk et al., 2005a; Vovk and Petej, 2014; Vovk et al., 2015; Lambrou et al., 2015); we briefly discuss this framework and show some connections to our work in Appendix 3.E.

Calibration has natural applications in numerous sensitive domains where uncertainty estimation is desirable (healthcare, finance, forecasting). Recently, calibrated classifiers have been used as a part of the pipeline for anomaly detection (Hendrycks et al., 2019; Lee et al., 2018) and label shift estimation (Saerens et al., 2002; Alexandari et al., 2020; Garg et al., 2020).

3.6 Conclusion

We analyzed post-hoc uncertainty quantification for binary classification problems from the standpoint of robustness to distributional assumptions. By connecting calibration to confidence intervals and prediction sets, we established that popular parametric 'scaling' methods cannot provide informative calibration in the distribution-free setting. In contrast, we showed that a nonparametric 'binning' method — histogram binning — satisfies approximate and asymptotic calibration guarantees without distributional assumptions. We also established guarantees for the cases of streaming data and covariate shift.

Takeaway message. Recent calibration methods that perform binning on top of parametric methods (Platt-binning (Kumar et al., 2019) and IROvA-TS (Zhang et al., 2020)) have achieved strong empirical performance. In light of our theoretical findings, we recommend some form of binning as the last step of calibrated prediction due to the robust distribution-free guarantees provided by Theorem 3.4.

3.7 Broader impact

Machine learning is regularly deployed in real-world settings, including areas having high impact on individual lives such as granting of loans, pricing of insurance and diagnosis of medical conditions. Often, instead of hard $0/1$ classifications, these systems are required to produce soft probabilistic predictions, for example of the probability that a startup may go bankrupt in the next few years (in order to determine whether to give it a loan) or the probability that a person will recover from a disease (in order to price an insurance product). Unfortunately, even though classifiers produce numbers between 0 and 1, these are well known to not be 'calibrated' and hence not be interpreted as probabilities in any real sense, and using them in lieu of probabilities can be both misleading (to the bank granting the loan) and unfair (to the individual at the receiving end of the decision).

Thus, following early research in meteorology and statistics, in the last couple of decades the ML community has embraced the formal goal of calibration as a way to quantify uncertainty as well as to interpret classifier outputs. However, there exist other alternatives to quantify uncertainty, such as confidence intervals for the regression function and prediction sets for the binary label. There is not much guidance on which of these should be employed in practice, and what the relationship between them is, if any. Further, while there are many post-hoc calibration techniques, it is unclear which of these require distributional assumptions to work and which do not—this is critical because making distributional assumptions (for convenience) on financial or medical data is highly suspect.

This chapter explicitly relates the three aforementioned notions of uncertainty quantification without making distributional assumptions, describes what is possible and what is not. Importantly, by providing distribution-free guarantees on well-known variants of binning, we identify a conceptually simple and theoretically rigorous way to ensure calibration in high-risk real-world settings. Our tools are thus likely to lead to fairer systems, better estimates of risks of high-stakes decisions, and more human-interpretable outputs of classifiers that apply out-of-the-box in many real-world settings because of the assumption-free guarantees.

Appendices for Chapter 3

The Appendix contains proofs of results in Chapter 3 ordered as they appear. Auxiliary results needed for some of the proofs are stated in Appendix 3.F.

3.A Proof of Proposition 3.1

The 'if' part of the theorem is due to Vaicenavicius et al. (2019, Proposition 1); we reproduce it for completeness. Let $\sigma(g), \sigma(f)$ be the sub σ-algebras generated by g and f respectively. By definition of f, we know that f is $\sigma(g)$-measurable and, hence, $\sigma(f) \subseteq \sigma(g)$. We now have:

$$\begin{aligned}\mathbb{E}\left[Y \mid f(X)\right] &= \mathbb{E}\left[\mathbb{E}\left[Y \mid g(X)\right] \mid f(X)\right] && \text{(by tower rule since } \sigma(f) \subseteq \sigma(g)\text{)}\\ &= \mathbb{E}\left[f(X) \mid f(X)\right] && \text{(by property (3.5))}\\ &= f(X).\end{aligned}$$

The 'only if' part can be verified for $g = f$. Since f is perfectly calibrated,

$$\mathbb{E}\left[Y \mid f(X) = f(x)\right] = f(x),$$

almost surely P_X.

□

3.B Proofs of results in Section 3.3

3.B.1 Proof of Theorem 3.1

Assume that one is given a predictor f that is (ϵ, α)-calibrated. Then the assertion follows from the definition of (ϵ, α)-calibration since:

$$|\mathbb{E}\left[Y \mid f(X)\right] - f(X)| \leqslant \epsilon \implies \mathbb{E}\left[Y \mid f(X)\right] \in C(f(X)).$$

Now we show the proof in the other direction. If m_C was injective, $\mathbb{E}\left[Y \mid m_C(f(X))\right] = \mathbb{E}\left[Y \mid f(X)\right]$ and thus if $\mathbb{E}\left[Y \mid f(X)\right] \in C(f(X))$ (which happens with probability at least $1 - \alpha$), we would have $\mathbb{E}\left[Y \mid m_C(f(X))\right] \in C(f(X))$ and so

$$|\mathbb{E}\left[Y \mid m_C(f(X))\right] - m_C(f(X)| \leqslant \sup_{z \in \mathrm{Range}(f)} \{|C(z)|/2\} = \epsilon.$$

This serves as an intuition for the proof in the general case, when m_C need not be injective. Note that,

$$\begin{aligned}|\mathbb{E}[Y \mid m_C(f(X))] - m_C(f(X))| &= |\mathbb{E}[Y \mid m_C(f(X))] - \mathbb{E}[m_C(f(X)) \mid m_C(f(X))]| \\ &\overset{(1)}{=} |\mathbb{E}[\mathbb{E}[Y \mid f(X)] \mid m_C(f(X))] - \mathbb{E}[m_C(f(X)) \mid m_C(f(X))]| \\ &\overset{(2)}{=} |\mathbb{E}[\mathbb{E}[Y \mid f(X)] - m_C(f(X)) \mid m_C(f(X))]| \\ &\overset{(3)}{\leqslant} \mathbb{E}[|\mathbb{E}[Y \mid f(X)] - m_C(f(X))| \mid m_C(f(X))], \end{aligned} \tag{3.25}$$

where we use the tower rule in (1) (since m_C is a function of f), linearity of expectation in (2) and Jensen's inequality in (3). To be clear, the outermost expectation above is over $f(X)$ (conditioned on $m_C(f(X))$). Consider the event

$$A : \mathbb{E}[Y \mid f(X)] \in C(f(X)).$$

On A, by definition we have:

$$|\mathbb{E}[Y \mid f(X)] - m_C(f(X))| = \frac{u_C(f(X)) - l_C(f(X))}{2} \leqslant \sup_{z \in \text{Range}(f)} \left(\frac{|C(z)|}{2}\right) = \epsilon.$$

By monotonicity property of conditional expectation, we also have that conditioned on A,

$$\mathbb{E}[|\mathbb{E}[Y \mid f(X)] - m_C(f(X))| \mid m_C(f(X))] \leqslant \mathbb{E}[\epsilon \mid m_C(f(X))] = \epsilon,$$

with probability 1. Thus by the relationship proved in the series of equations ending in (3.25), we have that conditioned on A, with probability 1,

$$|\mathbb{E}[Y \mid m_C(f(X))] - m_C(f(X))| \leqslant \epsilon.$$

Since we are given that C is a $(1-\alpha)$-CI with respect to f, $\mathbb{P}(A) \geqslant 1-\alpha$. For any event B, it holds that $\mathbb{P}(B) \geqslant \mathbb{P}(B|A)\,\mathbb{P}(A)$. Setting

$$B : |\mathbb{E}[Y \mid m_C(f(X))] - m_C(f(X))| \leqslant \epsilon,$$

we obtain:

$$\mathbb{P}\left(|\mathbb{E}[Y \mid m_C(f(X))] - m_C(f(X))| \leqslant \epsilon\right) \geqslant 1-\alpha.$$

Thus, we conclude that $m_C(f(\cdot))$ is (ϵ, α)-calibrated. □

3.B.2 Proof of Theorem 3.2

In the proof, we denote the operation $C(\cdot) \cap \{0,1\}$ as $\text{disc}(C)$ (for 'discretize'). Suppose C_n is a $(1-\alpha)$-CI with respect to f for all distributions P. We show that C_n covers the label Y_{n+1} itself for distributions $P \in \mathcal{P}_f$ (and thus $\text{disc}(C_n)$ would also cover the labels).

Consider any distribution $P \in \mathcal{P}_f$ is nonatomic. Fix a set of $m \geqslant n+1$ samples from the distribution P denoted as $\mathcal{T} = \{(A^{(j)}, B^{(j)})\}_{j \in [m]}$. Given $\mathcal{T}$, consider a distribution Q corresponding to the following sampling procedure for $(X, Y) \sim Q$:

> sample an index j uniformly at random from $[m]$ and set $(X, Y) = (A^{(j)}, B^{(j)})$.

The distribution function for Q is given by

$$m^{-1} \sum_{j=1}^{m} \delta_{(A^{(j)}, B^{(j)})}.$$

where $\delta_{(a,b)}$ denotes the points mass at (a, b). Note that Q is only defined conditional on $\mathcal{T}$. Observe the following facts about Q:

- supp$(Q) = \{(A^{(j)}, B^{(j)})\}_{j \in [m]}$.
- Consider any $(x, y) \in$ supp(Q). Let $(x, y) = (A^{(j)}, B^{(j)})$ for some $j \in [m]$. Then

$$\begin{aligned} \mathbb{E}_Q[Y \mid f(X) = f(x)] &= \mathbb{E}_Q\left[Y \mid f(X) = f(A^{(j)})\right] \\ &\overset{\xi_1}{=} \mathbb{E}_Q\left[Y \mid X = A^{(j)}\right] \\ &\overset{\xi_2}{=} B^{(j)} = y. \end{aligned}$$

  Above ξ_1 holds since $P_{f(X)}$ is nonatomic so that the $f(X^{(i)})$'s are unique almost surely. Note that $P_{f(X)}$ is nonatomic only if P_X itself is nonatomic. Thus the $A^{(j)}$'s are unique almost surely, and ξ_2 follow. In other words, if $(X, Y) \sim Q$, then we have

$$Y = \mathbb{E}_Q[Y \mid f(X)]. \tag{3.26}$$

Suppose the data distribution was Q, that is $\{(X_i, Y_i)\}_{i \in [n+1]} \sim Q^{n+1}$. Define the event that the CI guarantee holds as

$$E_1 : \mathbb{E}[Y_{n+1} \mid f(X_{n+1})] \in C_n(f(X_{n+1})), \tag{3.27}$$

and the event that the PS guarantee holds as

$$E_2 : Y_{n+1} \in C_n(f(X_{n+1})). \tag{3.28}$$

Then due to (3.26), the events are exactly the same under Q:

$$E_1 \overset{Q}{\equiv} E_2. \tag{3.29}$$

In particular, this means

$$Q^{n+1}(\mathbb{E}[Y_{n+1} \mid f(X_{n+1})] \in C_n(f(X_{n+1}))) = Q^{n+1}(Y_{n+1} \in C_n(f(X_{n+1}))). \tag{3.30}$$

If C_n is a distribution-free CI, then $Q^{n+1}(E_1) \geqslant 1 - \alpha$ and thus $Q^{n+1}(E_2) \geqslant 1 - \alpha$. This shows that for Q, disc(C_n) is a $(1 - \alpha)$-PI.

Note that Q corresponds to sampling with replacement from a fixed set $\mathcal{T}$, where each element of $\mathcal{T}$ is drawn with respect to P. Although $Q \neq P$, we expect that as $m \to \infty$ (while n is fixed), Q and P coincide. This would prove the result for general P. To formalize this intuition, we describe a distribution which is close to Q but corresponds to sampling *without replacement* from $\mathcal{T}$ instead.

For this, now suppose that $\{(X_i, Y_i)\}_{i \in [n+1]} \sim R^{n+1}$ where R^{n+1} corresponds to sampling without replacement from $\mathcal{T}$. Formally, to draw from R^{n+1}, we first draw a surjective mapping $\lambda : [n+1] \to [m]$ as

$$\lambda \sim \text{Unif}\,(n\text{-sized ordered subsets of } [m]),$$

and set $(X_i, Y_i) = (A^{(\lambda(i))}, B^{(\lambda(i))})$ for $i \in [n+1]$.

First we quantify precisely the intuition that as $m \to \infty$, Q^{n+1} and R^{n+1} are essentially identical. Consider the event "T : no index is repeated when sampling from Q^{n+1}". Let $\mathbb{P}(T) = \tau_m$ for some m and note that $\lim_{m\to\infty} \tau_m = 1$. Now consider any probability event E over $\{(X_i, Y_i)\}_{i \in [n+1]}$ (such as E_1 or E_2). We have

$$\begin{aligned} Q^{n+1}(E) &= Q^{n+1}(E|T) \cdot \mathbb{P}(T) + Q^{n+1}(E|T^c) \cdot \mathbb{P}(T^c) \\ &\in [Q^{n+1}(E|T) \cdot \mathbb{P}(T), Q^{n+1}(E|T) \cdot \mathbb{P}(T) + \mathbb{P}(T^c)]. \end{aligned}$$

Now observe that $Q^{n+1}(E|T) = R^{n+1}(E)$ to conclude

$$Q^{n+1}(E) \in [R^{n+1}(E) \cdot \mathbb{P}(T), R^{n+1}(E) \cdot \mathbb{P}(T) + \mathbb{P}(T^c)].$$

Since $m \geqslant n+1$, $\mathbb{P}(T) \neq 0$ so we can invert the above and substitute $\tau_m = \mathbb{P}(T)$ to get

$$R^{n+1}(E) \in \left[\tau_m^{-1}(Q^{n+1}(E) - (1 - \tau_m)),\ \tau_m^{-1} Q^{n+1}(E)\right]. \tag{3.31}$$

Consider $E = E_2$ defined in equation (3.28). We showed that $Q^{n+1}(E_2) \geqslant 1 - \alpha$. Thus from (3.31),

$$R^{n+1}(E_2) \geqslant \tau_m^{-1}(1 - \alpha - (1 - \tau_m)).$$

The above is with respect to R^{n+1} which is conditional on a fixed draw $\mathcal{T}$. However since the right hand side is independent of $\mathcal{T}$, we can also include the randomness in $\mathcal{T}$ to say:

$$\mathbb{P}_{R^{n+1}, \mathcal{T}}(E_2) \geqslant \tau_m^{-1}(1 - \alpha - (1 - \tau_m)). \tag{3.32}$$

Observe that if we consider the marginal distribution over R^{n+1} and $\mathcal{T}$ (that is we include the randomness in $\mathcal{T}$ as above), $\{(X_i, Y_i)\}_{i \in [n+1]} \overset{iid}{\sim} P$. (This is not true if we do not marginalize over $\mathcal{T}$, since due to sampling without replacement, the (X_i, Y_i)'s are not independent.) Thus equation (3.32) can be restated as

$$P^{n+1}(E_2) \geqslant \tau_m^{-1}(1 - \alpha - (1 - \tau_m)),$$

Since m can be set to any number and $\lim_{m\to\infty} \tau_m = 1$, we can indeed conclude

$$P^{n+1}(E_2) \geqslant 1 - \alpha.$$

Recall that E_2 is the event that $Y_{n+1} \in C_n(X_{n+1})$; equivalently $Y_{n+1} \in \text{disc}(C_n(X_{n+1}))$. Thus $\text{disc}(C_n)$ provides a $(1-\alpha)$-PI for all $P \in \mathcal{P}_f$. □

3.B.3 Proof of Corollary 3.1

Consider any distribution $Q \in \mathcal{P}_f$ such that $Q_{f(X)}$ is nonatomic. Then define P such that $P_X = Q_X$ and $P(Y = 1 \mid X) = 0.5$ a.s. Q_X. Clearly, $P_{f(X)} = Q_{f(X)}$ is nonatomic, so that $P \in \mathcal{P}_f$. Further, $\mathbb{E}_P\left[Y_{n+1} \mid f(X)\right] = 0.5$ a.s. $P_{f(X)}$.

Since C_n is a distribution-free CI w.r.t. f and $P \in \mathcal{P}_f$, by Theorem 3.2, C_n must provide both a prediction set and a confidence interval for P:

$$P^{n+1}(\mathbb{E}\left[Y_{n+1} \mid f(X_{n+1})\right] \in C_n(f(X_{n+1}))) \geqslant 1 - \alpha,$$

and

$$P^{n+1}(Y_{n+1} \in C_n(f(X_{n+1}))) \geqslant 1 - \alpha.$$

Thus by a union bound

$$P^{n+1}(\{Y_{n+1}, \mathbb{E}\left[Y_{n+1} \mid f(X_{n+1})\right]\} \subseteq C_n(f(X_{n+1}))) \geqslant 1 - 2\alpha. \tag{3.33}$$

Note that if

$$\{Y_{n+1}, \mathbb{E}\left[Y_{n+1} \mid f(X_{n+1})\right]\} \subseteq C_n(f(X_{n+1})),$$

then $|C_n(X_{n+1})| \geqslant |Y_{n+1} - \mathbb{E}\left[Y_{n+1} \mid f(X_{n+1})\right]| \geqslant 0.5$. Thus

$$P^{n+1}(|C_n(f(X_{n+1}))| \geqslant 0.5) \geqslant 1 - 2\alpha.$$

Consequently we have

$$\begin{aligned}\mathbb{E}_{P^{n+1}}|C_n(f(X_{n+1}))| &\geqslant 0.5(1 - 2\alpha)\\ &= 0.5 - \alpha.\end{aligned}$$

This concludes the proof. □

3.B.4 Proof of Theorem 3.3

Suppose $\mathcal{A}$ is distribution-free asymptotically calibrated for some $\alpha \in (0, 0.5)$ and some $\{\epsilon_n \in [0,1]\}_{n \in \mathbb{N}}$ with $\lim_{n\to\infty} \epsilon_n = 0$. We show that this assumption leads to a contradiction to Corollary 3.1.

Consider any function $f : \mathcal{X} \to [0,1]$. By the definition of asymptotic calibration, $h_n = \mathcal{A}(\mathcal{D}_n, f)$ is (ϵ_n, α)-calibrated for every $n \in \mathbb{N}$. Approximate calibration implies that for the event $E_1 : |\mathbb{E}\left[Y \mid h_n(X)\right] - h_n(X)| \leqslant \epsilon_n$, we have $P^{n+1}(E_1) \geqslant 1 - \alpha$. Following the intuition of Theorem 3.1, observe that the event E_1 is clearly identical to the event $E_2 : \mathbb{E}\left[Y \mid h_n(X)\right] \in [h_n(X) - \epsilon_n, h_n(X) + \epsilon_n]$. Thus $P^{n+1}(E_2) \geqslant 1 - \alpha$. Next, note that since the mapping m_n produced by $\mathcal{A}$ is injective, $\mathbb{E}\left[Y \mid h_n(X)\right] = \mathbb{E}\left[Y \mid m_n(f(X))\right] = \mathbb{E}\left[Y \mid f(X)\right]$. Thus, defining $C_n(f(X)) := [m_n(f(X)) - \epsilon_n, m_n(f(X)) + \epsilon_n] = [h_n(X) - \epsilon_n, h_n(X) + \epsilon_n]$, we have that

$$1 - \alpha \leqslant P^{n+1}(E_2)$$

$$
\begin{aligned}
&= P^{n+1}(\mathbb{E}\left[Y \mid h_n(X)\right] \in [h_n(X) - \epsilon_n, h_n(X) + \epsilon_n]) \\
&= P^{n+1}(\mathbb{E}\left[Y \mid f(X)\right] \in [h_n(X) - \epsilon_n, h_n(X) + \epsilon_n]) \\
&= P^{n+1}(\mathbb{E}\left[Y \mid f(X)\right] \in C_n(f(X))),
\end{aligned}
$$

showing that the defined C_n is a distribution-free $(1-\alpha)$-CI w.r.t. f. Further, since we have that $\sup_{z\in[0,1]} |C_n(z)| = 2\epsilon_n$, for any distribution P, we have

$$\lim_{n\to\infty} \mathbb{E}_{P^{n+1}}|C_n(f(X_{n+1}))| \leqslant 2 \lim_{n\to\infty} \epsilon_n = 0.$$

Thus, there exists a constant m such that for all $n \geqslant m$ and any distribution P,

$$\mathbb{E}_{P^{n+1}} |C_n(f(X_{n+1}))| < 0.5 - \alpha. \tag{3.34}$$

(Note that this requires $0.5 - \alpha > 0$, which is true since $\alpha \in (0, 0.5)$.)

Clearly, Corollary 3.1 is in contradiction to (3.34), as long as the assumptions required for Corollary 3.1 hold. We already have that C_n is a distribution-free $(1-\alpha)$-CI w.r.t. f. All we need to do is exhibit a function f such that $\mathcal{P}_f \neq \emptyset$. Indeed, Lemma 3.1 shows that any f whose range contains an interval of $[0, 1]$ suffices.

Having satisfied the assumptions of Corollary 3.1, we conclude that there exists a distribution $Q \in \mathcal{P}_f$ such that

$$\mathbb{E}_{Q^{n+1}} |C_n(f(X_{n+1}))| \geqslant 0.5 - \alpha.$$

This contradicts (3.34). Hence our hypothesis that $\mathcal{A}$ is distribution-free asymptotically calibrated must be false, concluding the proof. □

3.B.5 Characterizing a class of functions f for which $\mathcal{P}_f$ is non-empty

Lemma 3.1. *If Range(f) contains a sub-interval of $[0, 1]$, then $\mathcal{P}_f$ is non-empty.*

Proof. Let the interval $I = [a, b]$ with $a < b \in [0, 1]$ be contained in Range(f), that is,

$$\forall z \in I, \exists x \in \mathcal{X} : f(x) = z. \tag{3.35}$$

Let λ denote the Lebesgue measure on $[0, 1]$ and $\mathcal{B}_{[0,1]}$ the Borel σ-algebra on $[0, 1]$. Define the uniform probability measure I on P':

$$P'(S) = \lambda(S \cap I)/\lambda(I); \;\; S \in \mathcal{B}_{[0,1]}. \tag{3.36}$$

This is well defined since $\lambda(I) = b - a > 0$. Clearly, P' does not have atoms on $\mathcal{B}_{[0,1]}$.

We now want to construct a measure $P^\star$ on the Borel σ-algebra on $\mathcal{X}$ such that the push-forward of $P^\star$ under f is P'. One can easily check that $\left\{f^{-1}(S) : S \in \mathcal{B}_{[0,1]}\right\}$ defines a σ-algebra on $\mathcal{X}$. Then, one can define a measure $P^\star$ over this σ-algebra as $P^\star(f^{-1}(S)) = P'(S)$. Can $P^\star$ be extended to the Borel σ-algebra over $\mathcal{X}$? Ershov (1975) studied this problem, leading to the following result.

Theorem 3.8 (Theorem 2.5 by Ershov (1975), adapted)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be complete and separable metric spaces with $\mathcal{B}_{\mathcal{X}}$ and $\mathcal{B}_{\mathcal{Y}}$ being the corresponding Borel σ-algebras. Let $f : (\mathcal{X}, \mathcal{B}_{\mathcal{X}}) \to (\mathcal{Y}, \mathcal{B}_{\mathcal{Y}})$ be a measurable mapping and ν a probability measure on $(\mathcal{Y}, \mathcal{B}_{\mathcal{Y}})$. If f satisfies*

$$f^{-1}(B) \neq \varnothing \quad \textit{for all } B \in \mathcal{B}_{\mathcal{Y}} : \nu(B) > 0, \tag{3.37}$$

then there exists a probability measure μ on $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$ satisfying

$$\mu(f^{-1}(B)) = \nu(B), \quad \forall B \in \mathcal{B}_{\mathcal{Y}}.$$

We invoke Ershov's result with $\mathcal{Y} = [0,1]$ and $\nu = P'$. Assumption (3.35) guarantees that condition (3.37) is fulfilled. We conclude that there exists a probability measure $P^\star$ on $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$, for which $P^\star_{f(X)} = P'$ is non-atomic. Thus $P^\star \in \mathcal{P}_f$, concluding the proof.

□

3.C Proofs of results in Section 3.4 (other than Section 3.4.4)

3.C.1 Proof of Theorem 3.4

Let $E_{\mathcal{B}(x)}$ be the event that $(\mathcal{B}(X_1), \ldots, \mathcal{B}(X_n)) = (\mathcal{B}(x_1), \ldots, \mathcal{B}(x_n))$. On the event $E_{\mathcal{B}(x)}$, within each region $\mathcal{X}_b$, the number of point from the calibration set is known and the Y_i's in each bin represent independent Bernoulli random variables that share the same mean $\pi_b = \mathbb{E}[Y \mid X \in \mathcal{X}_b]$. Consider any fixed region $\mathcal{X}_b$, $b \in [B]$. Using Theorem 3.11, we obtain that:

$$\mathbb{P}\left(|\pi_b - \widehat{\pi}_b| > \sqrt{\frac{2\widehat{V}_b \ln(3B/\alpha)}{N_b}} + \frac{3\ln(3B/\alpha)}{N_b} \;\middle|\; E_{\mathcal{B}(x)} \right) \leqslant \alpha/B.$$

Applying union bound across all regions of the sample-space partition, we get that:

$$\mathbb{P}\left(\forall b \in [B] : \; |\pi_b - \widehat{\pi}_b| \leqslant \sqrt{\frac{2\widehat{V}_b \ln(3B/\alpha)}{N_b}} + \frac{3\ln(3B/\alpha)}{N_b} \;\middle|\; E_{\mathcal{B}(x)} \right) \geqslant 1 - \alpha.$$

Because this is true for any $E_{\mathcal{B}(x)}$, we can marginalize to obtain the assertion of the theorem in unconditional form. □

3.C.2 Proof of Corollary 3.3

We convert the per-bin confidence interval of Theorem 3.4 to a calibration guarantee using the same intuition as that of Theorem 3.1. Define the function $C : [B] \to \mathcal{I}$ given by

$$C_n(b) = \left[\widehat{\pi}_b - \left(\sqrt{\frac{2\widehat{V}_b \ln(3B/\alpha)}{N_b}} + \frac{3\ln(3B/\alpha)}{N_b} \right), \widehat{\pi}_b + \sqrt{\frac{2\widehat{V}_b \ln(3B/\alpha)}{N_b}} + \frac{3\ln(3B/\alpha)}{N_b} \right],$$

for every $b \in [B]$. Then by Theorem 3.4, C_n provides a '$(1-\alpha)$-CI with respect to $\mathcal{B} : \mathcal{X} \to [B]$'. While Definition 3.3 defined CIs with respect to a function whose range is $[0, 1]$, it can be naturally extended for CIs with respect to functions with another range such as $\mathcal{B}$. In Section 3.3.1, the calibrated function constructed from C was defined as $\widetilde{f}(x) := m_C(f(x))$; the same construction applies even if Range$(f) \neq [0, 1]$. Specifically, for the C defined above, $\widetilde{f}(x) = \widehat{\pi}_{\mathcal{B}(x)}$. The arguments in the proof of the CI-to-calibration part of Theorem 3.1 give that $\widetilde{f}$ is (ϵ, α)-calibrated with

$$\epsilon = \sup_{b \in [B]} |C(b)|/2 = \sqrt{\frac{2\widehat{V}_{b^\star} \ln(3B/\alpha)}{N_{b^\star}}} + \frac{3\ln(3B/\alpha)}{N_{b^\star}}.$$

This shows the approximate calibration result. Next, we show the asymptotic calibration result. Suppose some bin b has $\mathbb{P}(\mathcal{B}(X) = b) = 0$. Then, a test point X_{n+1} almost surely does not belong to the bin, and the bin can be ignored for our calibration guarantee. Thus without loss of generality, suppose every $b \in [B]$ satisfies

$$\mathbb{P}(\mathcal{B}(X) = b) > 0.$$

Let $\min_{b \in [B]} \mathbb{P}(\mathcal{B}(X) = b) = \tau > 0$. Then for a fixed number of samples n, any particular bin b, and any constant $\alpha \in (0, 1)$ we have by Hoeffding's inequality with probability $1 - \alpha/B$

$$N_b \geqslant n\tau - \sqrt{\frac{n\ln(B/\alpha)}{2}}.$$

Taking a union bound, we have with probability $1 - \alpha$, simultaneously for every $b \in [B]$,

$$N_b \geqslant n\tau - \sqrt{\frac{n\ln(B/\alpha)}{2}} = \Omega(n),$$

and in particular $N_{b^\star} = \Omega(n)$ where $b^\star = \arg\min_{b \in [B]} N_b$. Thus by the first part of this corollary, h_n is ϵ_n calibrated where $\epsilon_n = O(\sqrt{n^{-1}}) = o(1)$. This concludes the proof. □

3.C.3 Proof of Theorem 3.5

Denote $|\mathcal{D}^2_{cal}| = n$. Let $p_j = \mathbb{P}(g(X) \in I_j)$ be the true probability that a random point falls into partition $\mathcal{X}_j$. Assume c is such that we can use Lemma 3.2 to guarantee that with probability at least $1 - \alpha/2$, uniform mass binning scheme is 2-well-balanced. Hence, with probability at least $1 - \alpha/2$:

$$\frac{1}{2B} \leqslant p_j \leqslant \frac{2}{B}, \ \forall j \in [B]. \tag{3.38}$$

Moreover, by Hoeffding's inequality we get that for any fixed region of sample-space partition, with probability at least $1 - \alpha/2B$, for a fixed $j \in [B]$,

$$N_j \geqslant np_j - \sqrt{\frac{n\ln(2B/\alpha)}{2}}. \tag{3.39}$$

Hence, by union bound across applied accross all regions and using (3.38), we get that with probability at least $1 - \alpha/2$:

$$N_{b^\star} \geqslant \frac{n}{2B} - \sqrt{\frac{n\ln(2B/\alpha)}{2}},$$

where the first term dominates asymptotically (for fixed B). Hence, we get that with probability at least $1-\alpha$, $N_{b^\star} = \Omega\left(n/B\right)$. By invoking the result of Corollary 3.3 and observing that $\widehat{V}_b \leqslant 1$, we conclude that uniform mass binning is (ϵ, α)-calibrated with $\epsilon = O(\sqrt{B\ln(B/\alpha)/n})$ as desired. This also leads to asymptotic calibration by Corollary 3.3. □

3.C.4 Proof of Theorem 3.6

The proof is based on the result for an empirical-Bernstein confidence sequences for bounded observations (Howard et al., 2021). We condition on the event $E^{\infty}_{\mathcal{B}(x)} : (\mathcal{B}(X_1), \mathcal{B}(X_1), \dots) = (\mathcal{B}(x_1), \mathcal{B}(x_2), \dots)$, that is the random variables denoting which partition the infinite stream of samples fall in (thus allowing our bound to hold for every possible value of n). On $E^{\infty}_{\mathcal{B}(x)}$, the label values within each partition of the sample-space partition represent independent Bernoulli random variable that share the same mean $\pi_b = \mathbb{E}\left[Y \mid X \in \mathcal{X}_b\right], b \in [B]$. Consequently, the bound obtained can be marginalized over $E^{\infty}_{\mathcal{B}(x)}$ to obtain the assertion of the theorem in unconditional form. Now we show the bound that applies conditionally on $E^{\infty}_{\mathcal{B}(x)}$.

Consider any fixed region of the sample-space partition $\mathcal{X}_b$ and the corresponding points $\left\{\left(X_i^b, Y_i^b\right)\right\}_{i=1}^{N_b}$. Then $S_t = \left(\sum_{i=1}^t Y_i^b\right) - t\pi_b$ is a sub-exponential process with variance process:

$$\widehat{V}_t^+ = \sum_{i=1}^{t} \left(Y_i^b - \overline{Y}_{i-1}^b\right)^2.$$

Howard et al. (2020, Proposition 2) implies that S_t is also a sub-gamma process with variance process $\widehat{V}_t$ and the same scale $c=1$. Since the theorem holds for any sub-exponential uniform boundary, we choose one based on analytical convenience. Recall definition of the polynomial stitching function

$$\mathcal{S}_\alpha(v) := \sqrt{k_1^2 v l(v) + k_2^2 c^2 l^2(v)} + k_2 c l(v), \quad \text{where} \begin{cases} l(v) := \ln h(\ln_\eta(v/m)) + \ln(l_0/\alpha), \\ k_1 := (\eta^{1/4} + \eta^{-1/4})/\sqrt{2}, \\ k_2 := (\sqrt{\eta}+1)/\sqrt{2}. \end{cases}$$

where $l_0 = 1$ for the scalar case. Note that for $c > 0$ it holds that $\mathcal{S}_\alpha(v) \leqslant k_1\sqrt{vl(v)} + 2ck_2 l(v)$. From Howard et al. (2021, Theorem 1), it follows that $u(v) = \mathcal{S}_\alpha(v \vee m)$ is a sub-gamma uniform boundary with scale c and crossing probability α. Applying Theorem 3.10 with $h(k) \leftarrow (k+1)^s\zeta(s)$ where $\zeta(\cdot)$ is Riemann zeta function and parameters $\eta \leftarrow e$, $s \leftarrow 1.4$, $c \leftarrow 1$, $m \leftarrow 1$ and $\alpha \leftarrow \alpha/(2B)$, yields that $k_2 \leqslant 1.88$, $k_1 \leqslant 1.46$ and $l(v) = 1.4 \cdot \ln\ln\left(ev\right) + \ln(2\zeta(1.4)B/\alpha)$. Since Theorem 3.10 provides a bound that holds uniformly across time t, then it provides a guarantee for $t = N_b$, in particular. Hence, with probability at least $1 - \alpha/B$,

$$|\pi_b - \widehat{\pi}_b| \leqslant \frac{1.46\sqrt{\widehat{V}_b^+ \cdot 1.4 \cdot \ln\ln\left(e\left(\widehat{V}_b^+ \vee 1\right)\right) + \ln(6.3B/\alpha)}}{N_b} +$$

$$\frac{5.27 \cdot \ln\ln\left(e\left(\widehat{V}_b^+ \vee 1\right)\right) + 3.76\ln(6.3B/\alpha)}{N_b}$$

$$\leqslant \frac{7\sqrt{\widehat{V}_b^+ \cdot \ln\ln\left(e\left(\widehat{V}_b^+ \vee 1\right)\right)} + 5.3\ln(6.3B/\alpha)}{N_b}.$$

using that $\sqrt{x+y} \leqslant \sqrt{x} + \sqrt{y}$ and $\ln\ln(ex) \leqslant \sqrt{x \ln\ln ex}$ for $x \geqslant 1$. Finally, we apply a union bound to get a guarantee that holds simultaneously for all regions of the sample-space partition. □

3.D Calibration under covariate shift (including proofs of results in Section 3.4.4)

The results from Section 3.4.4 are proved in Appendix 3.D.1 (Theorem 3.7) and 3.D.3 (Proposition 3.2). To show Theorem 3.7, we first propose and analyze a slightly different estimator than (3.46) that is unbiased for $\pi_b^{(w)}$, but needs additional oracle access to the parameters $\{m_b\}_{b\in[B]}$ defined as

$$m_b = P(X \in \mathcal{X}_b) \,/\, \widetilde{P}(X \in \mathcal{X}_b).$$

The ratio m_b denotes the 'relative mass' of region $\mathcal{X}_b$. (For simplicity, we assume that $\widetilde{P}(X \in \mathcal{X}_b) > 0$ for every b since otherwise the test-point almost surely does not belong to $\mathcal{X}_b$ and estimation in that bin is not relevant for a calibration guarantee.) We then show that m_b can be estimated using w, which would lead to the proposed estimator $\breve{\pi}_b^{(w)}$. First, we establish the following relationship between $\mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right]$ and $\mathbb{E}_P\left[Y \mid X \in \mathcal{X}_b\right]$.

Proposition 3.3. *Under the covariate shift assumption, for any $b \in [B]$,*

$$\mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right] = m_b \cdot \mathbb{E}_P\left[w(X)Y \mid X \in \mathcal{X}_b\right].$$

Proof. Observe that

$$\frac{d\widetilde{P}(X \mid X \in \mathcal{X}_b)}{dP(X \mid X \in \mathcal{X}_b)} = \frac{d\widetilde{P}(X)}{dP(X)} \cdot \frac{P\left(X \in \mathcal{X}_b\right)}{\widetilde{P}(X \in \mathcal{X}_b)} = w(X) \cdot m_b.$$

Thus we have,

$$\begin{aligned}
\mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right] &\overset{(1)}{=} \mathbb{E}_{\widetilde{P}}\left[\mathbb{E}_{\widetilde{P}}\left[Y \mid X\right] \mid X \in \mathcal{X}_b\right] \\
&\overset{(2)}{=} \mathbb{E}_{\widetilde{P}}\left[\mathbb{E}_P\left[Y \mid X\right] \mid X \in \mathcal{X}_b\right] \\
&\overset{(3)}{=} \mathbb{E}_P\left[\frac{d\widetilde{P}(X \mid X \in \mathcal{X}_b)}{dP(X \mid X \in \mathcal{X}_b)} \cdot \mathbb{E}_P\left[Y \mid X\right] \mid X \in \mathcal{X}_b\right] \\
&\overset{(4)}{=} m_b \cdot \mathbb{E}_P\left[w(X)\mathbb{E}_P\left[Y \mid X\right] \mid X \in \mathcal{X}_b\right]
\end{aligned}$$

$$\begin{aligned}&\overset{(5)}{=} m_b \cdot \mathbb{E}_P\left[\mathbb{E}_P\left[w(X)Y \mid X\right] \mid X \in \mathcal{X}_b\right] \\ &\overset{(6)}{=} m_b \cdot \mathbb{E}_P\left[w(X)Y \mid X \in \mathcal{X}_b\right],\end{aligned}$$

where in (1) we use the tower rule, in (2) we use the covariate shift assumption, (3) can be seen by using the integral form of the expectation, (4) uses the observation at the beginning of the proof, (5) uses that $w(X)$ is a function of X and finally, (6) uses the tower rule. □

Let N_b denote the number of calibration points from the source domain that belong to bin b. Given Proposition 3.3, a natural estimator for $\mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right]$ is given by:

$$\widehat{\pi}_b^{(w)} := \frac{1}{N_b} \sum_{i:\mathcal{B}(X_i)=b} m_b w(X_i) Y_i. \tag{3.40}$$

Estimation properties of $\widehat{\pi}_b^{(w)}$ are given by the following theorem.

Theorem 3.9. *Assume that* $\sup_x w(x) = U < \infty$. *For any* $\alpha \in (0, 1)$, *with probability at least* $1 - \alpha$,

$$\left|\widehat{\pi}_b^{(w)} - \mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right]\right| \leqslant \sqrt{\frac{2\widehat{V}_b^{(w)} \ln(3B/\alpha)}{N_b}} + \frac{3 m_b U \ln(3B/\alpha)}{N_b}, \quad \textit{simultaneously for all } b \in [B],$$

where $\widehat{V}_b^{(w)} = \frac{1}{N_b} \sum_{i:\mathcal{B}(X_i)=b} (m_b w(X_i) Y_i - \widehat{\pi}_b^{(w)})^2$.

The proof is given in Appendix 3.D.2. Next, we discuss a way of estimating m_b using likelihood ratio w instead of relying on oracle access. Observe that

$$\frac{d\widetilde{P}(X \mid X \in \mathcal{X}_b)}{dP(X \mid X \in \mathcal{X}_b)} = \frac{d\widetilde{P}(X)}{dP(X)} \cdot \frac{P\left(X \in \mathcal{X}_b\right)}{\widetilde{P}(X \in \mathcal{X}_b)} = w(X) \cdot m_b.$$

Thus we have,

$$\mathbb{E}_P\left[w(X) \mid X \in \mathcal{X}_b\right] = m_b^{-1} \mathbb{E}_P\left[\frac{d\widetilde{P}(X \mid X \in \mathcal{X}_b)}{dP(X \mid X \in \mathcal{X}_b)} \mid X \in \mathcal{X}_b\right] = m_b^{-1}, \tag{3.41}$$

which suggests a possible estimator for m_b given by

$$\widehat{m}_b = \left(\frac{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)}{N_b}\right)^{-1}, \quad b \in [B]. \tag{3.42}$$

On substituting this estimate for m_b in (3.40), we get a new estimator

$$\frac{\sum_{i:\mathcal{B}(X_i)=b} w(X_i) Y_i}{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)},$$

which is exactly $\breve{\check{\pi}}_b^{(w)}$. With this observation, we now prove Theorem 3.7.

3.D.1 Proof of Theorem 3.7

Let us define $r_b := 1/m_b$ and

$$\widehat{r}_b = \frac{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)}{N_b}. \tag{3.43}$$

Step 1 (Uniform lower bound for N_b). Since the regions of the sample-space partition were constructed using uniform-mass binning, the guarantee of Theorem 3.5 holds. Precisely, we have that with probability at least $1 - \alpha/3$, simultaneously for every $b \in [B]$,

$$N_b \geqslant \frac{n}{2B} - \sqrt{\frac{n \ln(6B/\alpha)}{2}}.$$

Step 2 (Approximating r_b). Observe that the estimator (3.43) is an average of N_b random variables bounded by the interval $[0, U]$. Let $E_{\mathcal{B}(x)}$ be the event that $(\mathcal{B}(X_1), \dots, \mathcal{B}(X_n)) = (\mathcal{B}(x_1), \dots, \mathcal{B}(x_n))$. On the event $E_{\mathcal{B}(x)}$, within each region $\mathcal{X}_b$, the number of point from the calibration set is known and the Y_i's in each bin represent independent Bernoulli random variables that share the same mean $\mathbb{E}\left[w(X) \mid X \in \mathcal{X}_b\right]$. Consider any fixed region $\mathcal{X}_b$, $b \in [B]$. By Hoeffding's inequality, it holds that

$$\mathbb{P}\left(|r_b - \widehat{r}_b| > \sqrt{\frac{U^2 \ln(6B/\alpha)}{2N_b}} \;\middle|\; E_{\mathcal{B}(x)}\right) \leqslant \alpha/(3B).$$

Applying union bound across all regions of the sample-space partition, we get that:

$$\mathbb{P}\left(\exists b \in [B] :\ |r_b - \widehat{r}_b| > \sqrt{\frac{U^2 \ln(6B/\alpha)}{2N_b}} \;\middle|\; E_{\mathcal{B}(x)}\right) \leqslant \alpha/3.$$

Because this is true for any $E_{\mathcal{B}(x)}$, we can marginalize to obtain that with probability at least $1 - \alpha/3$,

$$\forall b \in [B],\ |r_b - \widehat{r}_b| \leqslant \sqrt{\frac{U^2 \ln(6B/\alpha)}{2N_b}}. \tag{3.44}$$

Step 3 (Going from r_b to m_b). Define $r^\star = \min_{b \in [B]} \mathbb{E}\left[w(X) \mid X \in \mathcal{X}_b\right]$. Suppose $\forall b \in [B]$, $|r_b - \widehat{r}_b| \leqslant \epsilon$ and $\epsilon < r^\star/2$. Then, we have with probability at least $1 - \alpha/3$:

$$|m_b - \widehat{m}_b| = \left|\frac{1}{r_b} - \frac{1}{\widehat{r}_b}\right| = \left|\frac{r_b - \widehat{r}_b}{r_b \cdot \widehat{r}_b}\right| \leqslant \frac{\epsilon}{r_b^2 |1 - \epsilon/r_b|} \leqslant \frac{2\epsilon}{r_b^2} = 2m_b^2 \epsilon, \quad \forall b \in [B]. \tag{3.45}$$

We now set $\epsilon = \sqrt{\frac{U^2 \ln(6B/\alpha)}{2N_b}}$ as specified in equation (3.44) and verify that $\epsilon < r^\star/2$.

- First, from step 1, with probability at least $1 - \alpha/3$, $N_{b^\star} = \Omega(n/B)$ and thus $N_b = \Omega(n/B)$ for every $b \in [B]$.

- By the condition in the theorem statement, for every $b \in [B]$,

$$\epsilon = \sqrt{\frac{U^2 \ln(6B/\alpha)}{2N_b}} = O\left(\sqrt{\frac{U^2 B \ln(6B/\alpha)}{n}}\right) = O\left(\sqrt{\frac{U^2 B \ln(6B/\alpha)}{\left(\frac{U^2 B \ln(6B/\alpha)}{L^2}\right)}}\right) = O\left(L\right).$$

  Finally recall that $L \leqslant r^\star$. Thus we can pick c in the theorem statement to be large enough such that $\epsilon < L/2 \leqslant r^\star/2$.

Thus for $\epsilon = \sqrt{\frac{U^2 \ln(6B/\alpha)}{2N_b}}$, by a union bound over the event in (3.44) and step 1, the conditions for (3.45) are satisfied with probability at least $1 - 2\alpha/3$. Hence we have for some large enough constant $c > 0$,

$$|m_b - \widehat{m}_b| \leqslant c m_b^2 \cdot \sqrt{\frac{U^2 B \ln(6B/\alpha)}{2n}} \leqslant c \cdot \frac{U}{L^2}\sqrt{\frac{B \ln(6B/\alpha)}{2n}}.$$

The final inequality holds by observing that $m_b \leqslant 1/L$ which follows from relationship (3.41) and the assumption that $\inf_x w(x) \geqslant L$.

Step 4 (Computing the final deviation inequality for $\widecheck{\pi}_b^{(w)}$). Recall the definitions of the two estimators:

$$\widehat{\pi}_b^{(w)} := \frac{1}{N_b} \sum_{i:\mathcal{B}(X_i)=b} m_b w(X_i) Y_i,$$

and

$$\widecheck{\pi}_b^{(w)} := \frac{1}{N_b} \sum_{i:\mathcal{B}(X_i)=b} \widehat{m}_b w(X_i) Y_i,$$

which differ by replacing m_b by its estimator $\widehat{m}_b$ defined in (3.42). By triangle inequality,

$$|\widecheck{\pi}_b - \mathbb{E}\left[Y \mid X \in \mathcal{X}_b\right]| \leqslant \left|\widecheck{\pi}_b^{(w)} - \widehat{\pi}_b^{(w)}\right| + \left|\widehat{\pi}_b^{(w)} - \mathbb{E}\left[Y \mid X \in \mathcal{X}_b\right]\right|.$$

Theorem 3.9 bounds the term $\left|\widehat{\pi}_b^{(w)} - \mathbb{E}\left[Y \mid X \in \mathcal{X}_b\right]\right|$ with high probability. In the proof of Theorem 3.9, we can replace the empirical Bernstein's inequality by Hoeffding's inequality to obtain with probability at least $1 - \alpha/3$,

$$\left|\widehat{\pi}_b^{(w)} - \mathbb{E}\left[Y \mid X \in \mathcal{X}_b\right]\right| \leqslant \sqrt{\frac{U^2 \ln(6B/\alpha)}{2N_b}} \leqslant \left(\frac{U}{L}\right)^2 \sqrt{\frac{\ln(6B/\alpha)}{2N_b}},$$

simultaneously for all $b \in [B]$ (the last inequality follows since $L \leqslant 1 \leqslant U$). To bound $\left|\widehat{\pi}_b^{(w)} - \widecheck{\pi}_b^{(w)}\right|$, first note that:

$$\left|\widehat{\pi}_b^{(w)} - \widecheck{\pi}_b^{(w)}\right| = \left|\frac{1}{N_b} \sum_{i:\mathcal{B}(X_i)=b} \left(\widehat{m}_b - m_b\right) w(X_i) Y_i\right|$$

$$\begin{aligned} &\leqslant U \cdot \left| \frac{1}{N_b} \sum_{i:\mathcal{B}(X_i)=b} (\widehat{m}_b - m_b) \right| \\ &= U \cdot |\widehat{m}_b - m_b| . \end{aligned}$$

Then we use the results from steps 1 and 3 to conclude that with probability at least $1 - 2\alpha/3$,

$$\left| \check{\pi}_b^{(w)} - \widehat{\pi}_b^{(w)} \right| \leqslant c \cdot \left(\frac{U}{L} \right)^2 \sqrt{\frac{B \ln(6B/\alpha)}{2n}}, \quad \text{and} \quad N_b \geqslant n/B - \sqrt{\frac{n \ln(6B/\alpha)}{2}}.$$

simultaneously for all $b \in [B]$. Thus by union bound, we get that it holds with probability at least $1 - \alpha$,

$$|\check{\pi}_b - \mathbb{E}\left[Y \mid X \in \mathcal{X}_b\right]| \leqslant c \cdot \left(\frac{U}{L} \right)^2 \sqrt{\frac{B \ln(6B/\alpha)}{2n}},$$

simultaneously for all $b \in [B]$ and large enough absolute constant $c > 0$. This concludes the proof. □

3.D.2 Proof of Theorem 3.9

Consider the event $E_{\mathcal{B}(x)}$ defined as $(\mathcal{B}(X_1), \dots, \mathcal{B}(X_n)) = (\mathcal{B}(x_1), \dots, \mathcal{B}(x_n))$. Conditioned on $E_{\mathcal{B}(x)}$, since $\sup_x w(x) \leqslant U$, we get that $\widehat{\pi}_b^{(w)}$ is an average of independent nonnegative random variables $m_b w(X_i) Y_i$ that are bounded by $m_b U$ and share the same mean $m_b \, \mathbb{E}_P\left[w(X)Y \mid X \in \mathcal{X}_b\right] = \mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right]$ (by Proposition 3.3).Using Theorem 3.11 for a fixed $b \in [B]$, we obtain:

$$\mathbb{P}\left(\left| \widehat{\pi}_b^{(w)} - \mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right] \right| > \sqrt{\frac{2\widehat{V}_b \ln(3B/\alpha)}{N_b}} + \frac{3 m_b U \ln(3B/\alpha)}{N_b} \;\middle|\; E_{\mathcal{B}(x)} \right) \leqslant \alpha/B.$$

Applying a union bound over all $b \in [B]$, we get:

$$\mathbb{P}\left(\forall b \in [B] : \left| \widehat{\pi}_b^{(w)} - \mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right] \right| \leqslant \sqrt{\frac{2\widehat{V}_b \ln(3B/\alpha)}{N_b}} + \frac{3 m_b U \ln(3B/\alpha)}{N_b} \;\middle|\; E_{\mathcal{B}(x)} \right) \geqslant 1 - \alpha.$$

Because this is true for any $E_{\mathcal{B}(x)}$, we can marginalize to obtain the assertion of the theorem in unconditional form. □

3.D.3 Proof of Proposition 3.2

Fix any $\alpha \in (0, 0.5)$. For any $k \in \mathbb{N}$ observe that by triangle inequality,

$$\left| \check{\pi}_b^{(\widehat{w}_k)} - \mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right] \right| \leqslant \left| \check{\pi}_b^{(w)} - \mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right] \right| + \left| \check{\pi}_b^{(w)} - \check{\pi}_b^{(\widehat{w}_k)} \right| .$$

Consider any $\epsilon > 0$. Note that by Theorem 3.7, there exists sufficiently large n such that the first term is larger than $\epsilon/2$ with probability at most $\alpha/2$ simultaneously for all $b \in [B]$. Hence, it suffices to show that there exists a large enough k such that the probability of the second term exceeding $\epsilon/2$ is at most $\alpha/2$ simultaneously for all $b \in [B]$. While analyzing the second term, we treat n as a constant while leveraging the consistency of $\widehat{w}_k$ as $k \to \infty$. For simplicity, denote $\Delta_k = \sup_x |w(x) - \widehat{w}_k(x)|$. Then for any $b \in [B]$:

$$\begin{aligned}
\left|\breve{\pi}_b^{(w)} - \breve{\pi}_b^{(\widehat{w}_k)}\right| &= \left|\frac{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)Y_i}{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)} - \frac{\sum_{i:\mathcal{B}(X_i)=b} \widehat{w}_k(X_i)Y_i}{\sum_{i:\mathcal{B}(X_i)=b} \widehat{w}_k(X_i)}\right| \\
&\overset{(1)}{\leqslant} \left|\frac{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)Y_i}{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)} - \frac{\sum_{i:\mathcal{B}(X_i)=b} \widehat{w}_k(X_i)Y_i}{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)}\right| \\
&\quad + \left|\frac{\sum_{i:\mathcal{B}(X_i)=b} \widehat{w}_k(X_i)Y_i}{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)} - \frac{\sum_{i:\mathcal{B}(X_i)=b} \widehat{w}_k(X_i)Y_i}{\sum_{i:\mathcal{B}(X_i)=b} \widehat{w}_k(X_i)}\right| \\
&\overset{(2)}{\leqslant} n \cdot \Delta_k \cdot \left|\frac{1}{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)}\right| \\
&\quad + \left|\frac{1}{\sum_{i:\mathcal{B}(X_i)=b} w(X_i)} - \frac{1}{\sum_{i:\mathcal{B}(X_i)=b} \widehat{w}_k(X_i)}\right| \left|\sum_{i:\mathcal{B}(X_i)=b} \widehat{w}_k(X_i)Y_i\right| \\
&\overset{(3)}{\leqslant} \frac{n}{L} \cdot \Delta_k + \left(\frac{n \cdot \Delta_k}{(L - \Delta_k)L}\right) \cdot ((U + \Delta_k) \cdot n),
\end{aligned}$$

where (1) is due to the triangle inequality, (2) is due to the facts that the number of points in any bin is at most n and that absolute difference between $\widehat{w}$ and w is at most Δ_k, (3) combines the aforementioned reasons in (2) and the assumptions: $L \leqslant \inf_x w(x) \leqslant \sup_x w(x) \leqslant U$. Since $\Delta_k \xrightarrow{P} 0$, clearly there exists a large enough k such that:

$$\mathbb{P}\left(\left|\breve{\pi}_b^{(w)} - \breve{\pi}_b^{(\widehat{w}_k)}\right| \geqslant \epsilon/2\right) \leqslant \alpha/2.$$

Thus we conclude that $\breve{\pi}_{\mathcal{B}(\cdot)}^{(\widehat{w}_k)}$ is asymptotically calibrated at level α. □

3.D.4 Preliminary simulations

This section is structured as follows. We first describe the overall procedure for calibration under covariate shift. The finite-sample calibration guarantee of Theorem 3.7 holds for oracle w whereas in our experiments we will estimate w; to assess the loss in calibration due to this approximation, we introduce some standard techniques used in literature. The preliminary experiments are performed with simulated data which are described after this. Finally, we propose a modified estimator $\widetilde{\pi}_b^{(\widehat{w})}$ of $\mathbb{E}_{\widetilde{P}}\left[Y \mid X \in \mathcal{X}_b\right]$ which appears natural but has poor performance in practice.

Procedure. We describe how to construct approximately calibrated predictions practically. This involves approximating the importance weights w and the relatives mass terms $\{m_b\}_{b\in[B]}$. The summarized calibration procedure consists of the following steps:

1. Split the calibration set into two parts and use the first to perform *uniform mass* binning
2. Given unlabeled examples from both source and target domain, estimate $\widehat{w}$. The unconstrained Least-Squares Importance Fitting (uLSIF) procedure (Kanamori et al., 2009) is used for this.
3. Compute for every $b \in [B]$, the estimator as per (3.22), replacing w with $\widehat{w}$:

$$\check{\pi}_b^{(\widehat{w})} := \frac{\sum_{i:\mathcal{B}(X_i)=b} \widehat{w}(X_i)Y_i}{\sum_{i:\mathcal{B}(X_i)=b} \widehat{w}(X_i)}. \tag{3.46}$$

4. On a new test point from the target distribution, output the calibrated estimate $\check{\pi}^{(\widehat{w})}_{\mathcal{B}(X_{n+1})}$.

Assessment through reliability diagrams and ECE. Given a test set (from the target distribution) of size m: $\{(X_i', Y_i')\}_{i\in[m]}$ and a function $g : \mathcal{X} \to [0,1]$ that outputs approximately calibrated probabilities, we consider the reliability diagram to estimate its calibration properties. A reliability diagram is constructed using splitting the unit interval $[0,1]$ into non-overlapping intervals $\{I_b\}_{b\in[B']}$ for some B' as

$$I_i = \left[\frac{i-1}{B'}, \frac{i}{B'}\right), \ i = 1, \dots, B'-1 \ \text{ and } \ I_{B'} = \left[\frac{B'-1}{B'}, 1\right].$$

Let $\mathcal{B}' : [0,1] \to [B']$ denote the binning function that corresponds to this binning. We then compute the following quantities for each bin $b \in [B']$:

$$\begin{aligned} \mathrm{FP}(I_b) &= \frac{\sum_{i:\mathcal{B}'(X_i')=b} Y_i'}{|\{i : \mathcal{B}'(X_i') = b\}|} && \text{(fraction of positives in a bin),} \\ \mathrm{MP}(I_b) &= \frac{\sum_{i:\mathcal{B}'(X_i')=b} g(X_i')}{|\{i : \mathcal{B}'(X_i') = b\}|} && \text{(mean predicted probability in a bin).} \end{aligned}$$

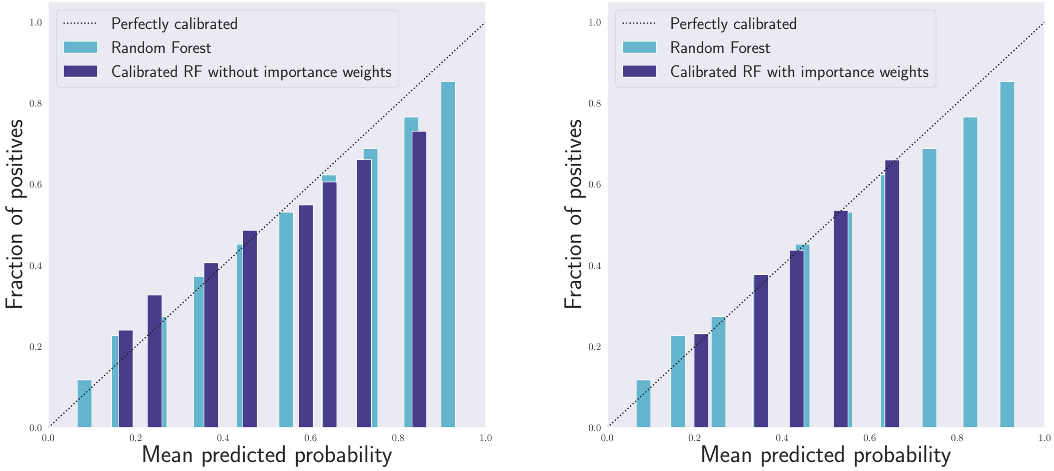
If g is perfectly calibrated, the reliability diagram is diagonal. Define the proportion of points that fall into various bins as:

$$\widehat{p}_b = \frac{|\{i : \mathcal{B}'(X_i') = b\}|}{m}, \quad b \in [B'].$$

Then ECE (or ℓ_1-ECE) is defined as:

$$\mathrm{ECE}(g) = \sum_{b\in[B']} \widehat{p}_b \cdot |\mathrm{MP}(I_b) - \mathrm{FP}(I_b)| .$$

ECE can also be defined in the ℓ_p sense and for multiclass problems but we limit our attention to the ℓ_1-ECE for binary problems.



(a)

(b)

Figure 3.2: In Figure 3.2a uncalibrated Random Forest (ECE ≈ 0.023) is compared with calibration that does not take the covariate shift into account (ECE ≈ 0.047). In Figure 3.2b uncalibrated Random Forest is compared with calibration that takes the covariate shift into account (ECE ≈ 0.017).

Simulations with synthetic data. We illustrate the performance of our proposed estimator (3.22) using the following simulated example, for which we can explicitly control the covariate shift. Consider the following data generation pipeline: for the source domain each component of the feature vector is drawn from Beta(α, β) where $\alpha = \beta = 1$, which corresponds to uniform draws from the unit cube. For the target distribution each component can be drawn independently from Beta(α', β'). If the dimension is d, the true likelihood ratio is given as

$$w(x) = \frac{d\widetilde{P}_X(x)}{dP_X(x)} = \frac{B^d(\alpha; \beta)}{B^d(\alpha'; \beta')} \prod_{i=1}^{d} \frac{(x_{(i)})^{\alpha'-1}(1 - x_{(i)})^{\beta'-1}}{(x_{(i)})^{\alpha-1}(1 - x_{(i)})^{\beta-1}},$$

where $x_{(i)}$ are the coordinates of feature vector x. We set $d = 3$ and $\alpha' = 2, \beta' = 1$ so that $w(x) = 8 \cdot x_{(1)}x_{(2)}x_{(3)}$. The labels for both source and target distributions are assigned according to:

$$\mathbb{P}(Y = 1 \mid X = x) = \frac{1}{2}\left(1 + \sin\left(\omega\left(x_{(1)}^2 + x_{(2)}^2 + x_{(3)}^2\right)\right)\right),$$

for $\omega = 20$. As the underlying classifier we use a Random Forest with 100 trees (from `sklearn`). 14700 data points were used to train the underlying Random Forest classifier, 2000 data points from both source and target were used for the estimation of importance weights. The parameters σ and λ for uLSIF were tuned by leave-one-out cross-validation: we considered 25 equally spaced values on a log-scale in range $(10^{-2}, 10^{2})$ for σ and 100 equally spaced values on a log-scale in range $(10^{-3}, 10^{3})$ for λ. Uniform mass binning was performed with 10 bins and 1940 data points from the source domain were used to estimate the quantiles. 7840 source data points were used for the calibration and finally, 28000 data points from the target domain were used for evaluation purposes. We note that this simulation is a 'proof-of-concept'; the sample sizes we used are not necessarily optimal can presumably be improved.

We compare the unweighted estimator (3.17) which corresponds to weighing points in each bin equally as we would do if there was no covariate shift, and the estimator (3.22) that uses an estimate of w to account for covariate shift. The reliability diagrams are presented in Figure 3.2, with the ECE reported in the caption. For the ECE estimation and reliability diagrams, we used $B' = 10$.

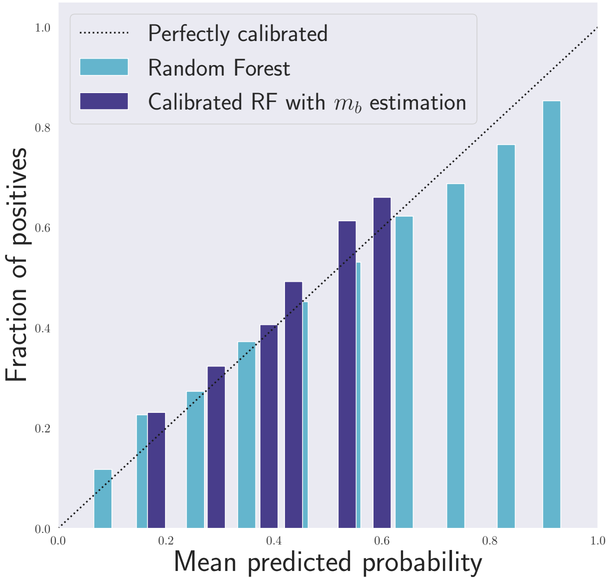


Figure 3.3: Calibration of Random Forest with m_b estimated as per equation (3.42) (ECE ≈ 0.05).

Alternative estimator for m_b. Estimator (3.42) is one way of estimating m_b using the w values, that leads to (3.22). However, there exists another natural estimator which we propose and show some preliminary empirical results for. Suppose we have access to additional unlabeled data from the source and target domains ($\{X_i^s\}_{i\in[n_s]}$, and $\{X_i^t\}_{i\in[n_t]}$ respectively). From the definition of $m_b = P_X(X \in \mathcal{X}_b)/\widetilde{P}_X(X \in \mathcal{X}_b)$, a natural estimator is,

$$\widehat{m}_b = \frac{\frac{1}{n_s}\left|\{i \in [n_s] : \mathcal{B}(X_i^s) = b\}\right|}{\frac{1}{n_t}\left|\{i \in [n_t] : \mathcal{B}(X_i^t) = b\}\right|}, \quad b \in [B]. \tag{3.47}$$

In this case, the estimator (3.40) reduces to:

$$\widetilde{\pi}_b^{(\widehat{w})} = \frac{\widehat{m}_b}{N_b} \sum_{i:\mathcal{B}(X_i)=b} \widehat{w}(X_i) Y_i.$$

We show experimental results with this estimation procedure. We used 8500 data points from the source domain and 8000 points from the target domain to compute (3.47). The reliability diagram and ECE with this estimator is reported in Figure 3.3. On our simulated dataset, we observe that the estimators $\widetilde{\pi}_b^{(\widehat{w})}$ perform significantly worse than the estimators $\breve{\pi}_b^{(\widehat{w})}$. While this is only a single experimental setup, we outline some drawbacks of this estimation method that may lead to poor performance in general.

1. $\widetilde{\pi}_b^{(\widehat{w})}$ requires access to additional unlabeled data from the source and target domains without leading to increase in performance.
2. The denominator of $\widehat{m}_b$ could be badly behaved if the number of points from the target domain in bin b are small. We could perform uniform-mass binning on the target domain to avoid this, but in this case N_b may be small which would lead to the estimator $\widetilde{\pi}_b^{(\widehat{w})}$ performing poorly.

Our overall recommendation through these preliminary experiments is to use the estimator $\widehat{\pi}_b^{(\widehat{w})}$ as proposed in Section 3.4.4 instead of $\widetilde{\pi}_b^{(\widehat{w})}$.

3.E Venn prediction

Venn prediction (Vovk et al., 2003; Vovk et al., 2005a; Vovk and Petej, 2014; Lambrou et al., 2015) is a calibration framework that provides distribution-free guarantees, which are different from the ones in Definitions 3.1 and 3.2. For a multiclass problem with L labels, Venn prediction produces L predictions, one of which is guaranteed to be perfectly calibrated (although it is impossible to know which one). These are called multiprobabilistic predictors, formally defined as a collection of predictions $(f_1, f_2, \dots f_L)$ where each $f_i \in \{\mathcal{X} \to \Delta_{L-1}\}$ (here Δ_{L-1} is the probability simplex in $\mathbb{R}^L$). Vovk and Petej (2014) defined two calibration guarantees for multiprobabilistic predictors, the first being oracle calibration.

Definition 3.4 (Oracle calibration). $(f_1, f_2, \dots f_L)$ is oracle calibrated if there exists an oracle selector S such that f_S is perfectly calibrated.

Venn predictors satisfy oracle calibration (Vovk and Petej, 2014, Theorem 1) with $S = Y$. In the binary case, this means that when $Y = 1$, $f_1(X)$ is perfectly calibrated but we do not have any guarantee on $f_0(X)$; on the other hand if $Y = 0$, $f_0(X)$ is perfectly calibrated but we know nothing about $f_1(X)$. Since Y is unknown, oracle calibration seems to us to primarily serve as theoretical guidance, but does not give a clear prescription on what to output and what theoretical guarantee that output satisfies. In practice, it seems reasonable to suspect that if $f_0(X)$ and $f_1(X)$ are close, then their average should be approximately calibrated in the sense of Definition 3.1, but to the best of our knowledge, such results have not been shown formally (other aggregate functions apart from average are also suggested (without formal guarantees) by Vovk and Petej (2014, Section 4)). For instance, it may be tempting to think that oracle calibration of a multiprobabilistic predictor leads to approximate calibration in the following way. Consider the prediction function

$$f(X) = \frac{\min f_i(X) + \max f_i(X)}{2},$$

and the radius of the interval $[\min f_i(X), \max f_i(X)]$:

$$\epsilon(X) = \frac{\max f_i(X) - \min f_i(X)}{2}.$$

Since Venn predictors satisfy oracle calibration, one might conjecture that f is (ϵ, α)-calibrated (per Definition 3.1) for the given function ϵ and for any $\alpha \in (0, 1)$. We examined this claim but were unable to prove such a guarantee formally. In fact, it seems that no general calibration guarantee should be possible with the size of the calibration interval being $O(\epsilon(X))$; we evidence this through the following construction.

Consider a setup, with no covariates and only label values Y, and a single bin that contains all points (in the Venn prediction language: a taxonomy under which all points are equivalent). For a test-point Y_{n+1} and any predictor f, note that $\mathbb{E}\left[Y_{n+1} \mid f\right]$ is simply equal to $\mathbb{E}\left[Y_{n+1}\right]$ since any information used to construct f is independent of Y_{n+1}. To ensure calibration, we may look for a guarantee of the following form for some δ:

$$\left|\mathbb{E}\left[Y_{n+1} \mid f\right] - f\right| = \left|\mathbb{E}\left[Y_{n+1}\right] - f\right| \leqslant \delta.$$

In essence, f is an estimator for the parameter $\mathbb{E}\left[Y\right]$ with a corresponding deviation bound of δ. Without distributional assumptions, we only expect to estimate such a parameter with error at best $\delta = O(1/\sqrt{n})$ for a fixed constant probability of failure. On the other hand, the Venn prediction interval $[\min f_i, \max f_i]$ often has radius $O(1/n)$. Thus for valid approximate calibration, we would need to provide a larger interval than $[\min f_i, \max f_i]$, even though one of the f_i's is perfectly calibrated. Given this example, our conjecture is that it might be possible to show that there always exists an $f_i(X)$ that is $(n^{-0.5}\text{polylog}\,(1/\alpha)), \alpha)$ calibrated. Without knowing which $f_i(X)$ to pick, perhaps one can show that an aggregate point in the interval $[\min f_i, \max f_i]$ is $((\max f_i - \min f_i) + n^{-0.5}\text{polylog}\,(1/\alpha)\,, \alpha)$-calibrated. In Section 3.4, we showed such a result for histogram binning (which can be interpreted as a Venn predictor). It would be interesting to study if such results can be shown for general Venn predictors.

Another guarantee for multiprobabilistic predictors is calibration in the large.

Definition 3.5 (Calibration in the large)**.** $(f_1, f_2, \ldots f_L)$ is calibrated in the large if the following is satisfied: $\mathbb{E}\left[Y\right] \in \left[\mathbb{E}\min f_i(X), \mathbb{E}\max f_i(X)\right]$.

Vovk and Petej (2014, Theorem 2) show that Venn predictors satisfy calibration in the large. Due to the expectation signs and the coverage of the marginal probability $\mathbb{E}\left[Y\right]$, calibration in the large does not lead to a clear interpretable guarantee for uncertainty quantification, but rather a minimum requirement that serves as a guiding principle.

3.F Auxiliary results

3.F.1 Concentration inequalities

Theorem 3.10 (Howard et al. (2021), Theorem 4)**.** *Suppose $Z_t \in [a, b]$ a.s. for all t. Let $(\widehat{Z}_t)$ be any $[a, b]$-valued predictable sequence, and let u be any sub-exponential uniform boundary with crossing probability α for scale $c = b - a$. Then:*

$$\mathbb{P}\left(\forall t \geqslant 1 : \left|\overline{Z}_t - \mu_t\right| < \frac{u\left(\sum_{i=1}^{t}\left(Z_i - \widehat{Z}_i\right)^2\right)}{t}\right) \geqslant 1 - 2\alpha.$$

Theorem 3.11 (Partial statement of Audibert et al. (2007), Theorem 1)**.** *Let $X_1, \ldots, X_n$ be i.i.d. random variables bounded in $[0, s]$, for some $s > 0$. Let $\mu = \mathbb{E}[X_1]$ be their common expected value. Consider the empirical mean $\overline{X}_n$ and variance V_n defined respectively by*

$$\overline{X}_n = \frac{\sum_{i=1}^n X_i}{n}, \quad \textit{and} \quad V_n = \frac{\sum_{i=1}^n (X_i - \overline{X}_n)^2}{n}.$$

Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\left|\overline{X}_n - \mu\right| \leqslant \sqrt{\frac{2V_n \log(3/\delta)}{n}} + \frac{3s\log(3/\delta)}{n}.$$

3.F.2 Uniform-mass binning

Kumar et al. (2019) defined well-balanced binning and showed that uniform mass-binning is well-balanced.

Definition 3.6 (Well-balanced binning)**.** A binning scheme $\mathcal{B}$ of size B is β-well-balanced ($\beta \geqslant 1$) for some classifier g if

$$\frac{1}{\beta B} \leqslant \mathbb{P}\left(g(X) \in I_b\right) \leqslant \frac{\beta}{B},$$

simultaneously for all $b \in [B]$.

To perform uniform-mass binning labeled examples are required at the stage of training the base classifier $g(\cdot)$. We denote this data as $\mathcal{D}^1_{\text{cal}}$. Procedures based on uniform-mass binning are well-balanced if $|\mathcal{D}^1_{\text{cal}}|$ is sufficiently large.

Lemma 3.2 (Kumar et al. (2019), Lemma 4.3)**.** *For a universal constant $c > 0$, if $|\mathcal{D}^1_{cal}| \geqslant cB\ln(B/\alpha)$, then with probability at least $1 - \alpha$, the uniform mass binning scheme $\mathcal{B}$ is 2-well-balanced.*

The calibration guarantees in Section 3.4 depend on the minimum number of training points $N_{b^\star}$ in any bin. Uniform mass-binning guarantees that $N_{b^\star} = \Omega(n/B)$. This is used in the proof of Theorem 3.5.

Chapter 4

Distribution-free calibration guarantees for histogram binning without sample splitting

This chapter is based on Gupta and Ramdas (2021).

We prove calibration guarantees for the popular histogram binning (also called uniform-mass binning) method of Zadrozny and Elkan (2001). Histogram binning has displayed strong practical performance, but theoretical guarantees have only been shown for sample split versions that avoid 'double dipping' the data. We demonstrate that the statistical cost of sample splitting is practically significant on a credit default dataset. We then prove calibration guarantees for the original method that double dips the data, using a certain Markov property of order statistics. Based on our results, we make practical recommendations for choosing the number of bins in histogram binning. In our illustrative simulations, we propose a new tool for assessing calibration—validity plots—which provide more information than an ECE estimate. Code for this work has been made publicly available at https://github.com/aigen/df-posthoc-calibration.

4.1 Introduction

In classification, the goal is to learn a model that uses observed feature measurements to make a class prediction on the categorical outcome. However, for safety-critical areas such as medicine and finance, a single class prediction might be insufficient and reliable measures of confidence or certainty may be desired. Such uncertainty quantification is often provided by predictors that produce not just a class label, but a probability distribution over the labels. If the predicted probability distribution is consistent with observed empirical frequencies of labels, the predictor is said to be calibrated (Dawid, 1982).

In this chapter we study the problem of calibration for binary classification; let $\mathcal{X}$ and $\mathcal{Y} = \{0, 1\}$ denote the feature and label spaces. We focus on the recalibration or post-hoc calibration setting, a standard statistical setting where the goal is to recalibrate existing ('pre-learnt') classifiers that are powerful and (statistically) efficient for classification accuracy, but do not satisfy calibration properties out-of-the-box. This setup is popular for recalibrating pre-trained deep nets. For example, Guo et al. (2017, Figure 4) demonstrated that a pre-learnt ResNet is initially miscalibrated, but can be effectively post-hoc calibrated. In the case of binary classification, the pre-learnt model can be an arbitrary predictor function that provides a classification 'score' $g \in \mathcal{G}$, where $\mathcal{G}$ is the space of all measurable functions from $\mathcal{X} \to [0, 1]$. Along with g, we are given access to a calibration dataset of size $n \in \mathbb{N}$, $\mathcal{D}_n = \{(X_i, Y_i)\}_{i \in [n]}$, drawn independently from a distribution $P \equiv P_X \times P_{Y|X}$. The goal is to define a calibrator $H : \mathcal{G} \times (\mathcal{X} \times [0, 1])^n \to \mathcal{G}$, that 'recalibrates' g to an approximately calibrated predictor $H(g, \mathcal{D}_n)$ (formally defined shortly). We denote $H(g, \mathcal{D}_n)$ as h. *All probabilities in this chapter are conditional on g and thus conditional on the data on which g is learnt.*

Let $\mathbb{E}[\cdot]$ denote the expectation operator associated with P, interpreted marginally or conditionally depending on the context. The predictor h is said to be perfectly calibrated if $\mathbb{E}[Y \mid h(X)] = h(X)$ (almost surely). While perfect calibration is impossible in finite samples, we desire a framework to make transparent claims about how close h is to being perfectly calibrated. The following notion proposed by Gupta et al. (2020) defines a calibrator that provides *probably approximate calibration* for chosen levels of approximation $\epsilon \in (0, 1)$ and failure $\alpha \in (0, 1)$. For brevity, we skip the qualification 'probably approximate'.

Definition 4.1 (Marginal calibration[1]). A calibrator $H : (g, \mathcal{D}_n) \mapsto h$ is said to be (ϵ, α)-marginally calibrated if for every predictor $g \in \mathcal{G}$ and distribution P over $\mathcal{X} \times [0, 1]$,

$$\mathbb{P}(|\mathbb{E}[Y|h(X)] - h(X)| \leqslant \epsilon) \geqslant 1 - \alpha. \tag{4.1}$$

The above probability is taken over both X and $\mathcal{D}_n$ since $h = H(g, \mathcal{D}_n)$ contains the randomness of $\mathcal{D}_n$. The qualification *marginal* signifies that the inequality $|\mathbb{E}[Y \mid h(X)] - h(X)| \leqslant \epsilon$ may not hold conditioned on X or $h(X)$, but holds only on *average*. We now define a more stringent conditional notion of calibration, which requires that approximate calibration hold simultaneously (or conditionally) for every value of the prediction.

Definition 4.2 (Conditional calibration). A calibrator $H : (g, \mathcal{D}_n) \mapsto h$ is (ϵ, α)-conditionally calibrated if for every predictor $g \in \mathcal{G}$ and distribution P over $\mathcal{X} \times [0, 1]$,

$$\mathbb{P}(\forall r \in \text{Range}(h), |\mathbb{E}[Y \mid h(X) = r] - r| \leqslant \epsilon) \geqslant 1 - \alpha. \tag{4.2}$$

In contrast to (4.1), the $\mathbb{P}$ above is only over $\mathcal{D}_n$. Evidently, if H is conditionally calibrated, it is also marginally calibrated. The conditional calibration property (4.2) has a PAC-style interpretation: with probability $1 - \alpha$ over $\mathcal{D}_n$, h satisfies the following deterministic property:

$$\forall r \in \text{Range}(h), |\mathbb{E}[Y \mid h(X) = r] - r| \leqslant \epsilon. \tag{4.3}$$

Marginal calibration does not have such an interpretation; we cannot infer from (4.1) a statement of the form "with probability $1 - \gamma$ over $\mathcal{D}_n$, h satisfies $\cdots$ ".

Marginal and conditional calibration assess the truth of the event $\mathbb{1}\{|\mathbb{E}[Y \mid h(X)] - h(X)| \leqslant \epsilon\}$ for a given ϵ. Instead we can consider bounding the expected value of $|\mathbb{E}[Y \mid h(X)] - h(X)|$ for $X \sim P_X$. This quantity is known as the expected calibration error.

[1] This definition is unrelated to that of Gneiting et al. (2007, Definition 1c), where marginal calibration refers to an asymptotic notion of calibration in the regression setting.

Definition 4.3 (Expected Calibration Error (ECE))**.** For $p \in [1, \infty)$, the ℓ_p-ECE of a predictor h is

$$\ell_p\text{-ECE}(h) = \left(\mathbb{E}_X |\mathbb{E}\left[Y \mid h(X)\right] - h(X)|^p\right)^{1/p}. \tag{4.4}$$

Note that the expectation above is only over $X \sim P_X$ and not over $\mathcal{D}_n$. We can ask for bounds on the ECE of $h = H(g, \mathcal{D}_n)$ that hold with high-probability or in-expectation over the randomness in $\mathcal{D}_n$. The conditional calibration property (4.3) for h implies a bound on the ℓ_p-ECE for every p, as formalized by the following proposition which also relates ℓ_p-ECE for different p.

Proposition 4.1. *For any predictor h and $1 \leqslant p \leqslant q < \infty$,*

$$\ell_p\textit{-ECE}(h) \leqslant \ell_q\textit{-ECE}(h). \tag{4.5}$$

Further, if (4.3) *holds, then ℓ_p-ECE$(h) \leqslant \epsilon, \forall p \in [1, \infty)$.*

The proof (in Appendix 4.A) is a straightforward application of Hölder's inequality. Informally, one can interpret the L.H.S. of (4.3) as the ℓ_∞-ECE of h so that (4.5) holds for $1 \leqslant p \leqslant q \leqslant \infty$. Thus conditional calibration is the strictest calibration property we consider: if H is (ϵ, α)-conditionally calibrated, then (a) H is (ϵ, α)-marginally calibration and (b) with probability $1 - \alpha$, ℓ_p-ECE$(h) \leqslant \epsilon$.

Example 4.1. We verify Proposition 4.1 on a simple example, which also helps build intuition for the various notions of calibration. Suppose h takes just two values: $\mathbb{P}(h(X) = 0.2) = 0.9$ and $\mathbb{P}(h(X) = 0.8) = 0.1$. Let $\mathbb{E}\left[Y \mid h(X) = 0.2\right] = 0.3$ and $\mathbb{E}\left[Y \mid h(X) = 0.8\right] = 0.6$. Then ℓ_1-ECE$(h) = 0.11 < \ell_2$-ECE$(h) \approx 0.114$. Marginal calibration (4.1) for $H(\cdot, \cdot) \equiv h$ is satisfied for $(\epsilon \geqslant 0.1, \alpha \leqslant 0.9)$, while the conditional calibration requirement (4.3) is only satisfied for $\epsilon \geqslant 0.2$.

In this chapter, we show that the histogram binning method of Zadrozny and Elkan (2001), described shortly, is calibrated in each of the above senses (marginal and conditional calibration; high-probability and in-expectation bounds on ECE), if the number of bins is chosen appropriately.

Some safety-critical domains may require calibration methods that are robust to the data-generating distribution. We refer to Definitions 4.1 and 4.2 as distribution-free (DF) guarantees since they are required to hold for all distributions over (X, Y) without restriction. This chapter is in the DF setting: the only assumption we make is that the calibration data $\mathcal{D}_n$ and (X, Y) are independent and identically distributed (i.i.d.). Gupta et al. (2020, Theorem 3) showed that if H is DF marginally calibrated with a meaningful value of ϵ (formally, ϵ can be driven to zero as sample size grows to infinity), then H must necessarily produce only discretized predictions (formally, Range(h) must be at most countable). We refer to such H as 'binning methods' — this emphasizes that H essentially partitions the sample-space into a discrete number of 'bins' and provides one prediction per bin (see Proposition 1 (Gupta et al., 2020)). Since our goal is DF calibration, we focus on binning methods.

4.1.1 Prior work on binning

Binning was initially introduced in the calibration literature for assessing calibration. Given a continuous scoring function h, if we wish to plot a reliability diagram (Sanders, 1963; Niculescu-Mizil and Caruana, 2005) or compute an ECE estimate (Miller, 1962; Sanders, 1963; Naeini et al., 2015), then h must first be discretized using binning. A common binning scheme used for this purpose is 'fixed-width binning', where $[0, 1]$ is partitioned into $B \in \mathbb{N}$ intervals (called bins) of width $1/B$ each and a single prediction is assumed for every bin. For example, if $B = 10$, then the width of each bin is 0.1, and if (say) $h(x) \in [0.6, 0.7)$ then the prediction is assumed to be 0.65.

Gupta et al. (2020, Theorem 3) showed that some kind of binning is in fact necessary to *achieve* DF calibration. The first binning method for calibration was proposed by Zadrozny and Elkan (2001) to calibrate a naive Bayes classifier. Their procedure is as follows. First, the interval $[0, 1]$ is partitioned into $B \in \mathbb{N}$ bins using the histogram of the $g(X_i)$ values, to ensure that each bin has the same number of calibration points (plus/minus one). Thus the bins have nearly 'uniform (probability) mass'. Then, the calibration points are assigned to bins depending on the interval to which the score $g(X_i)$ belongs to, and the probability that $Y = 1$ is estimated for each bin as the average of the observed Y_i-values in that bin. This average estimates the 'bias' of the bin. The binning scheme and the bias estimates together define h. A slightly modified version of this procedure is formally described in Algorithm 4.1.

While Algorithm 4.1 was originally called histogram binning, it has also been referred to as uniform-mass binning in some works. In the rest of this chapter, we use the latter terminology. Specifically, we refer to it as UMD, short for Uniform-Mass-Double-dipping. This stresses that the same data is used twice, both to determine inter-bin boundaries and to calculate intra-bin biases. UMD continues to remain a competitive benchmark in empirical work (Guo et al., 2017; Naeini et al., 2015; Roelofs et al., 2022), but no finite-sample calibration guarantees have been shown for it. (See however the following paragraph.) Some asymptotic consistency results for a histogram regression algorithm closely related to UMD were shown by Parthasarathy and Bhattacharya (1961) (see also the work by Lugosi and Nobel (1996)).

After the publication of this chapter in ICML 2021, we found out about a paper by Naeini et al. (2014), where a calibration bound for UMD is claimed (Theorem 3.1). However, their application of Hoeffding's inequality (namely quation (1)) in the proof of their result is incorrect, since it has not been shown that the points over which Hoeffding's inequality is applied are i.i.d. The technical contribution of our paper is exactly to show how to rescue the i.i.d. structure despite double dipping.

Zadrozny and Elkan (2002) proposed another popular binning method based on isotonic regression, for which some non-DF analyses exist (see Dai et al. (2020) and references therein). Recently, two recalibration methods closely related to UMD have been proposed, along with some theoretical guarantees that rely on sample-splitting — scaling-binning (Kumar et al., 2019) and sample split uniform-mass binning (Gupta et al., 2020).

In the scaling-binning method, the binning is performed on the output of another continuous recalibration method (such as Platt scaling (Platt, 1999)), and the bias for each bin is computed as the average of the output of the scaling procedure in that bin. This is unlike other binning methods, where the bias of each bin is computed as the average of the true outputs Y_i in that bin. Kumar et al. (2019, Theorem 4.1) showed that under some assumptions on the scaling class (which includes injectivity), the ECE of the sample split scaling-binning procedure is ϵ-close to $\sqrt{2}\,\ell_2$-ECE of the scaling procedure if, roughly, $n = \Omega(\log B/\epsilon^2)$. However, the results of Gupta et al. (2020, Section 3.3) imply that there exist data distributions on which any injective scaling procedure itself has trivial ECE.

In sample split uniform-mass binning, the first split of the data is used to define the bin boundaries so that the bins are balanced. The second split of the data is used for estimating the bin biases, using the average of the Y_i-values in the bin. We refer to this version as UMS, for Uniform-Mass-Sample-splitting. Gupta et al. (2020, Theorem 5) showed that UMS is (ϵ, α)-marginally calibrated if (roughly) $n = \Omega(B \log(B/\alpha)/\epsilon^2)$. To the best of our knowledge, this is the only known DF guarantee for a calibration method. However, in Section 4.2 we demonstrate that the constants in this guarantee are quite conservative, and the loss in performance due to sample splitting is practically significant on a real dataset.

4.1.2 Our contribution

We show tight DF calibration guarantees for the original method proposed by Zadrozny and Elkan (2001), UMD. While the existing theoretical analyses rely on sample splitting (Kumar et al., 2019; Gupta et al., 2020), it has been observed in experiments that double dipping to perform both bin formation and bias estimation on the same data leads to excellent practical performance (Zadrozny and Elkan, 2001; Guo et al., 2017; Kumar et al., 2019; Roelofs et al., 2022). Our work fills this gap in theory and practice.

We exploit a certain Markov property of order statistics, which are a set of classical, elegant results that are not well known outside of certain subfields of statistics (for one exposition of the Markov property, see Arnold et al. (2008, Chapter 2.4)). The strength of these probabilistic results is not widely appreciated — judging by their non-appearance in the ML literature — nor have they had implications for any modern AI applications that we are aware of. Thus, we consider it a central contribution of this work to have recognized that these mathematical tools can be brought to bear in order to shed light on a contemporary ML algorithm.

A simplified version of the Markov property is as follows: for order statistics $Z_{(1)}, Z_{(2)}, \ldots, Z_{(n)}$ of samples $\{Z_i\}_{i \in [n]}$ drawn i.i.d from any absolutely continuous distribution Q, and any indices $1 < i < j \leqslant n$, we have that

$$Z_{(j)} \perp Z_{(i-1)}, Z_{(i-2)}, \ldots, Z_{(1)} \mid Z_{(i)}.$$

For example, given the empirical median M, the points to its left are conditionally independent of the points to its right. Further each of these have a distribution that is identical to that of i.i.d. draws from $Z \sim Q$ when restricted to $Z < M$ (or $Z > M$). The implication is that if we form bins using the order statistics of the scores as the bin boundaries, then (a) the points within any bin are independent of the points outside that bin, and (b) conditioned on being in a given bin, say B_i, the points in the bin are i.i.d. with distribution $Q_{Z|Z \in B_i}$. When we split a calibration sample $\mathcal{D}$ and use one part $\mathcal{D}_1$ for binning and the other $\mathcal{D}\backslash\mathcal{D}_1$ for estimating bin probabilities, the points in $\mathcal{D}\backslash\mathcal{D}_1$ that belong to B_i are also conditionally i.i.d. with distribution $Q_{Z|Z\in B_i}$, which is exactly what we accomplished without sample splitting. In short, the Markov property allows us to 'double dip' the data, i.e., use the same data for binning and estimating within-bin probabilities.

Organization. Section 4.2 motivates our research problem by showing that UMS is sample-inefficient both in theory and practice. Empirical evidence is provided through a novel diagnostic tool called validity plots (Section 4.2.1). Section 4.3 presents UMD formally along with its analysis (main results in Theorems 4.1 and 4.2). Section 4.4 contains illustrative simulations. Proofs are in the supplement.

4.2 Sample split uniform-mass binning is inefficient

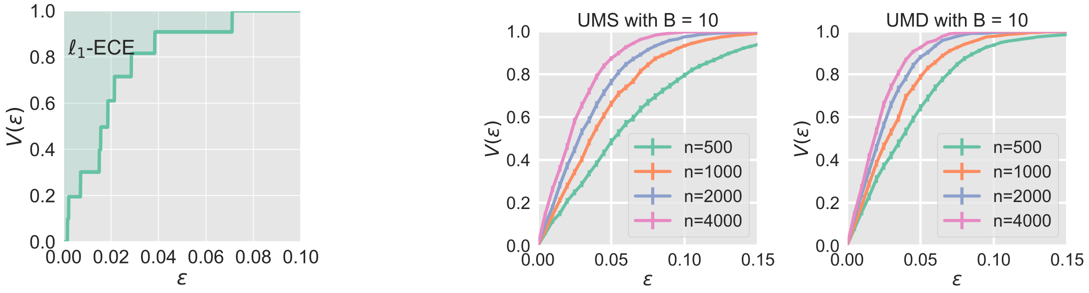
The DF framework encourages development of algorithms that are robust to arbitrarily distributed data. At the same time, the hope is that the DF guarantees are adaptive to real data and give meaningful bounds in practice. In this section, we assess if the practical performance of uniform-mass-sample-splitting (UMS) is well explained by its DF calibration guarantee (Gupta et al., 2020). As far as we know, this is the only known DF guarantee for a calibration method. However, we demonstrate that the guarantee is quite conservative. Further, we demonstrate that sample splitting leads to a drop in performance on a real dataset.

Suppose we wish to guarantee $(\epsilon, \alpha) = (0.1, 0.1)$-marginal calibration with $B = 10$ bins using UMS. We unpacked the DF calibration bound for UMS, and computed that to guarantee $(0.1, 0.1)$-marginal calibration with 10 bins, roughly $n \geqslant 17500$ is required. The detailed calculations can be found in Appendix 4.B. This sample complexity seems conservative for a binary classification problem. In Section 4.2.2, we use an illustrative experiment to show that the n required to achieve the desired level of calibration is indeed much lower than 17500. Our experiment uses a novel diagnostic tool called validity plots, introduced next.

4.2.1 Validity plots

Validity plots assess the marginal calibration properties of a calibration method by displaying estimates of the LHS of (4.1) as ϵ varies. Define the function $V : [0,1] \to [0,1]$ given by $V(\epsilon) = \mathbb{P}(|\mathbb{E}[Y \mid h(X)] - h(X)| \leqslant \epsilon)$. By definition of V, h is $(\epsilon, 1 - V(\epsilon))$-marginally calibrated for every ϵ. For this reason, we call the graph of V, $\{(\epsilon, V(\epsilon)) : \epsilon \in [0,1]\}$, as the 'validity curve'. (The term "curve" is used informally since V may have jumps.) Note the following neat relationship between the ℓ_1-ECE and the area-under-the-curve (AUC) of the validity curve:

$$
\begin{aligned}
\mathbb{E}[\ell_1\text{-ECE}(h)] &= \mathbb{E}[|\mathbb{E}[Y \mid h(X)] - h(X)|] \\
&= \int_0^1 \mathbb{P}(|\mathbb{E}[Y \mid h(X)] - h(X)| > \epsilon)\, d\epsilon
\end{aligned}
$$



(a) An illustrative validity plot. We can read off that marginal calibration is achieved for $(\epsilon, \alpha) = (0.04, 0.1)$ and $(0.03, 0.2)$. The ℓ_1-ECE estimate is roughly 0.023.

(b) Validity plots comparing UMD and UMS on the CREDIT dataset. The plots show that UMD has higher validity $V(\epsilon)$ for the same values of n, ϵ, and thus lower ℓ_1-ECE. For example, for $n = 1000$ and $\epsilon = 0.05$, UMS has $V(\epsilon) \approx 0.63$, while UMD has $V(\epsilon) \approx 0.79$.

Figure 4.1: Validity plots display estimates of $V(\epsilon) = \mathbb{P}(|\mathbb{E}[Y \mid h(X)] - h(X)| \leqslant \epsilon)$ as ϵ varies. Validity plots are described in Section 4.2.1. The experimental setup for Figure 4.1b is presented in Section 4.2.2.

$$\begin{aligned} &= 1 - \int_0^1 \mathbb{P}(|\mathbb{E}[Y \mid h(X)] - h(X)| \leqslant \epsilon)\, d\epsilon \\ &= 1 - \int_0^1 V(\epsilon)\, d\epsilon = 1 - \text{AUC(validity curve)}. \end{aligned}$$

A validity plot is a finite sample estimate of the validity curve on a single calibration set $\mathcal{D}_n$ and test set $\mathcal{D}_{\text{test}}$. We now outline the steps for constructing a validity plot. First, h is learned using $\mathcal{D}_n$ and g. Next, if h is not a binning method, it must be discretized through binning in order to enable estimation of $\mathbb{E}[Y \mid h(X)]$. This is identical to the binning step required by plugin ECE estimators and reliability diagrams. For example, one can use fixed-width binning as described in the first paragraph of Section 4.1.1. In this chapter, we empirically assess only binning methods, and so an additional binning step is not necessary. Next, the empirical distribution on $\mathcal{D}_{\text{test}}$ is used as a proxy for the true distribution of (X, Y), to estimate $V(\epsilon)$:

$$\begin{aligned} \widehat{V}(\epsilon) &= \frac{\sum_{(X_i,Y_i)\in\mathcal{D}_{\text{test}}} \mathbb{1}\left\{\left|\mathbb{E}_{\widehat{P}}[Y \mid h(X) = h(X_i)] - h(X_i)\right| \leqslant \epsilon\right\}}{|\mathcal{D}_{\text{test}}|}, \text{ where} \\ \mathbb{E}_{\widehat{P}}[Y \mid h(X) = h(x)] &\equiv \frac{\sum_{(X_i,Y_i)\in\mathcal{D}_{\text{test}}} Y_i \mathbb{1}\{h(X_i) = h(x)\}}{\sum_{(X_i,Y_i)\in\mathcal{D}_{\text{test}}} \mathbb{1}\{h(X_i) = h(x)\}}. \end{aligned} \tag{4.6}$$

For different values of $\epsilon \in [0, 1]$ on the X-axis, the estimate of $V(\epsilon)$ is plotted on the Y-axis to form the validity plot. Like the AUC of a validity curve corresponds to $\mathbb{E}[\ell_1\text{-ECE}]$, the AUC of a validity plot corresponds to the plugin ℓ_1-ECE estimate (Naeini et al., 2015). (There may be small differences in practice since we draw the validity plot for a finite grid of values in $[0, 1]$.) Thus validity plots convey the ℓ_1-ECE estimate and more.

Figure 4.1a displays an illustrative validity plot for a binning method with $B = 10$. V is a right-continuous step function with at most $|\text{Range}(h)| \leqslant B$ many discontinuities. Each ϵ for which there is a discontinuity in V corresponds to a bin that has $|\mathbb{E}[Y \mid h(X) = r] - r| = \epsilon$, and the incremental jump in the value of V, $V(\epsilon) - V(\epsilon^-)$, corresponds to the fraction of test points in that bin. Figure 4.1a was created using UMD, and thus each jump corresponds to roughly a $1/B = 0.1$ fraction of the test points. The ϵ values for the bins are approximately $10^{-3} \cdot (1.5, 2, 8, 16, 17, 19, 22, 29, 39, 71)$.

Unlike reliability diagrams (Niculescu-Mizil and Caruana, 2005), validity plots do not convey the predictions $h(X)$ to which the ϵ values correspond to, or the direction of miscalibration (whether $h(X)$ is higher or lower than $\mathbb{E}[Y \mid h(X)]$). On the other hand, validity plots convey the bin frequencies for every bin without the need for a separate histogram (such as the top panel in Niculescu-Mizil and Caruana (2005, Figure 1)). In our view, validity plots also 'collate' the right entity; we can easily read off from a validity plot practically meaningful statements such as "for 90% of the test points, the miscalibration is at most 0.04".

We can create a smoother validity plot that better estimates V by using multiple runs based on subsampled or bootstrapped data. To do this, for every $\epsilon \in [0, 1]$, $\widehat{V}(\epsilon)$ is computed separately for each run and the mean value is plotted as the estimate of $V(\epsilon)$. In our simulations, we always perform multiple runs, and also show $\pm$std-dev-of-mean in the plot. Figure 4.1b displays such validity plots (further details presented in the following subsection).

It is well known that plugin ECE estimators for a binned method are biased towards slightly overestimating the ECE (e.g., see Bröcker (2012), Kumar et al. (2019), and Widmann et al. (2019)). For the same reasons, $\widehat{V}(\epsilon)$ is a biased underestimate of $V(\epsilon)$. In other words, the validity plot is on average below the true validity curve. The reason for this bias is that to estimate ECE as well as to create validity plots, we compute $\left|\mathbb{E}_{\widehat{P}}[Y \mid h(X)] - h(X)\right|$ which can be written as $|\mathbb{E}[Y \mid h(X)] + \text{mean-zero-noise} - h(X)|$. On average, the noise term will lead to overestimating $|\mathbb{E}[Y \mid h(X)] - h(X)|$. However, the noise term is small if there is enough test data (if n_b is the number of test points in bin b, then the noise term is $O(\sqrt{1/n_b})$ w.h.p.). Further, it is highly unlikely that the noise will help some methods and hurts others. Thus validity plots can be reliably used to make inferences on the relative performance of different calibration methods. While there exist unbiased estimators for $(\ell_2\text{-ECE})^2$ (Bröcker, 2012; Widmann et al., 2019), we are not aware of any unbiased ℓ_1-ECE estimators. If such an estimator is proposed in the future, the same technique will also improve validity plots.

4.2.2 Comparing UMS and UMD using validity plots

Figure 4.1b uses validity plots to assess UMS and UMD on CREDIT, a UCI credit default dataset[2]. The task is to accurately predict the probability of default. The experimental protocol is as follows. The entire feature matrix is first normalized[3]. CREDIT has 30K (30,000) samples which are randomly split (once for the entire experiment) into splits (A, B, C) = (10K, 5K, 15K). First, g is formed by training a logistic regression model on split A and then re-scaling the learnt model using Platt scaling on split B (Platt scaling before binning was suggested by Kumar et al. (2019); we also observed that this helps in practice). Next, the calibration set $\mathcal{D}_n$ is formed by randomly

[2] Yeh and Lien (2009); https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[3] using Python's `sklearn.preprocessing.scale`

subsampling n ($\leqslant$10K) points from split C (without replacement). From the remaining points in split C, a test set of size 5K is subsampled (without replacement). The entire subsampling from split C is repeated 100 times to create 100 different calibration and test sets. For a given subsample, UMS/UMD with $B = 10$ is trained on the calibration set (with 50:50 sample splitting for UMS), and $\widehat{V}(\epsilon)$ for every ϵ is estimated on the test set. Finally, the (mean$\pm$std-dev-of-mean) of $\widehat{V}(\epsilon)$ is plotted with respect to ϵ. This experimental setup assesses marginal calibration for a fixed g, in keeping with our post-hoc calibration setting.

The validity plot in Figure 4.1b (left) indicates that the desired $(0.1, 0.1)$-marginal calibration is achieved by UMS with just $n = 1000$. Contrast this to $n \geqslant 17500$ required by the theoretical bound, as computed in Appendix 4.B. In fact, $n = 4000$ nearly achieves $(0.05, 0.1)$-marginal calibration. This gap occurs because the analysis of UMS is complex, with constants stacking up at each step.

Next, consider the validity plot for UMD in Figure 4.1b (right). By avoiding sample splitting, UMD achieves $(0.1, 0.1)$-marginal calibration at $n = 500$. In Section 4.3 we show that $n \geqslant 1500$ is provably sufficient for $(0.1, 0.1)$-marginal calibration and $n \geqslant 2900$ is sufficient for $(0.1, 0.1)$-conditional calibration. Some gap in theory and practice is expected since the theoretical bound is DF, and thus applies no matter how anomalous the data distribution is. However, the gap is much smaller compared to UMS, due to a clean analysis. In Section 4.4, we illustrate that the gap nearly vanishes for larger n. Section 4.4 also introduces the related concept of *conditional* validity plots that assess conditional calibration.

4.3 Distribution-free analysis of uniform-mass binning without sample splitting

Define the random variables $S = g(X)$; $S_i = g(X_i)$ for $i \in [n]$, called scores. Let $(S, Y) \sim Q$ and $S \sim Q_S$. In binning, we wish to use the calibration data $\{(S_i, Y_i)\}_{i \in [n]} \sim Q^n$ to (a) define a binning function $\mathcal{B} : [0, 1] \to [B]$ for some number of bins $B \in \mathbb{N}$, and (b) estimate the biases in the bins $\{\Pi_b := \mathbb{E}\left[Y \mid \mathcal{B}(S) = b\right]\}_{b \in [B]}$. We denote the bias estimates as $\widehat{\Pi}_b$. The approximately calibrated function is then defined as $h(\cdot) = \widehat{\Pi}_{\mathcal{B}(\cdot)}$.

Suppose the number of recalibration points is $n \approx 150$. In the absence of known properties of the data (i.e., in the DF setting), it seems reasonable to have $B = 1$ and define $H(g, \mathcal{D}_n)$ as the constant function $h(\cdot) := n^{-1} \sum_{i=1}^{n} Y_i$. Formally, $n = 150$ leads to the following Hoeffding-based confidence interval: with probability at least 0.9, $|n^{-1} \sum_{i=1}^{n} Y_i - \mathbb{E}Y| \leqslant \sqrt{\log(2/0.1)/(2 \cdot 150)} \approx 0.1$. In other words, if $n = 150$, H satisfies $(0.1, 0.1)$-marginal calibration. Of course, having a single bin completely destroys sharpness of h, but it's an instructive special case.

Suppose now that $n \approx 300$, and we wish to learn a non-constant h using two bins. If g is informative, we hope that $\mathbb{E}\left[Y \mid g(X) = \cdot\right]$ is roughly a monotonically increasing function. In light of this belief, it seems reasonable to choose a threshold t and identify the two bins as: $g(X) \leqslant t$ and $g(X) > t$. A natural choice for t is $M = \text{Median}(S_1, \ldots, S_n)$ since this ensures that both bins get the same number of points (plus/minus one). This is the motivation for UMD.

In this case, h and $\widehat{\Pi}$ are defined as,

$$h(\cdot) := \begin{cases} \widehat{\Pi}_1 := \text{Average}(Y_i : S_i \leqslant M) \text{ if } g(\cdot) \leqslant M \\ \widehat{\Pi}_2 := \text{Average}(Y_i : S_i > M) \text{ if } g(\cdot) > M. \end{cases} \tag{4.7}$$

Suppose M were the true median of Q_S instead of the empirical median. Then h has a calibration guarantee obtained by applying a Bernoulli concentration inequality separately for both bins and using a union bound (this is done formally by Gupta et al. (2020, Theorem 4)). In UMS, we try to emulate the true median case by using one split of the data to estimate the median. $\widehat{\Pi}$ is then computed on the second (independent) split of the data, and concentration inequalities can be used to provide calibration guarantees.

UMD does not sample split: in equation (4.7) above, M is computed using the same data that is later used to estimate $\widehat{\Pi}$. On the face of it, this double dipping eliminates the independence of the Y_i values required to apply a concentration inequality. However, we show that the independence structure can be retained if UMD is slightly modified. This subtle modification is to remove a single point from the bias estimation, namely the Y_i corresponding to the median M. (In comparison, in UMS we typically remove a fixed ratio of n.) The informal argument is as follows.

For simplicity, suppose Q_S is absolutely continuous (with respect to the Lebesgue measure), so that the S_i's are almost surely distinct, and suppose that the number of samples is odd: $n = 2m + 1$. Denote the ordered scores as $S_{(1)} < S_{(2)} < \ldots < S_{(n)}$ and let $Y_{(i)}$ denote the label corresponding to the score $S_{(i)}$. Thus $\widehat{\Pi}_1 = m^{-1} \sum_{i=1}^{m} Y_{(i)}$ and $M = S_{(m+1)}$. Clearly, $(S_{(i)}, Y_{(i)})$ is not independent of $S_{(m+1)}$ for any i. However, it turns out that the following property is true: conditioned on $S_{(m+1)}$, the unordered values $\{(S_{(i)}, Y_{(i)})\}_{i \in [m]}$ can be viewed as m *independent* samples identically distributed as (S, Y), given $S < S_{(m+1)}$. (Note that (S, Y) is an unseen and independent random variable.) Thus, we can use Hoeffding's inequality to assert: $\mathbb{P}(|\mathbb{E}\left[Y \mid M, S < M\right] - \widehat{\Pi}_1| \geqslant \epsilon \mid M, S < M) \leqslant 2 \exp\left(-2m\epsilon^2\right)$. This can be converted to a calibration guarantee on the first bin. The same bound can be shown if $S > M$, for the estimate $\widehat{\Pi}_2 = m^{-1} \sum_{i=m+1}^{2m+1} Y_{(i)}$. Using a union bound gives a calibration guarantee that holds for both bins simultaneously, which in turn gives conditional calibration.

In the following subsection, we show some key lemmas regarding the order statistics of the S_i's. These lemmas formalize what was argued above: *careful double dipping does not eliminate the independence structure*. In Section 4.3.2, we formalize the modified UMD algorithm, and prove that it is DF calibrated. Based on the guarantee for the modified version, Corollary 4.1 finally shows that the original UMD itself is DF calibrated.

Simplifying assumption. In the following analysis, we assume that $g(X)$ is absolutely continuous with respect to the Lebesgue measure, and thus has a probability density function (pdf). This assumption is made at no loss of generality, for reasons discussed in Appendix 4.C.1.

4.3.1 Key lemmas on order statistics

Consider two indices $i, j \in [n]$. The score S_i is not independent of the order statistic $S_{(j)}$. However, it turns out that conditioned on $S_{(j)}$, the distribution of S_i given $S_i < S_{(j)}$, is identical to the distribution of an unseen score S, given $S < S_{(j)}$. The following lemmas (both proved in Appendix 4.A) state versions of this fact that are useful for our analysis of UMD.

We first set up some notation. S is assumed to have a pdf, denoted as f. For some $1 \leqslant l < u \leqslant n$, consider the set of indices $\{i : S_{(l)} < S_i < S_{(u)}\}$, and index them arbitrarily as $\{t_1, t_2, \ldots, t_{u-l-1}\}$. This is just an indexing and not an ordering; in particular it is not necessary that $S_{t_1} = S_{(l+1)}$. For $j \in \{l+1, \ldots, u-1\}$, define $S_{\{j\}} = S_{t_{j-l}}$. Thus the set $\{S_{\{j\}} : j \in \{l+1, \ldots, u-1\}\}$ corresponds to the *unordered* S_i values between $S_{(l)}$ and $S_{(u)}$.

Lemma 4.1. *Fix $l, u \in [n]$ such that $l < u$. The conditional density of the unordered S_i values between the order statistics $S_{(l)}, S_{(u)}$, $f(S_{\{l+1\}}, \ldots, S_{\{u-1\}} \mid S_{(l)}, S_{(u)})$, is identical to the density of independent $S_i' \sim Q_S$, conditional on lying between $S_{(l)}, S_{(u)}$:*

$$f(S_1', \ldots, S_{u-l-1}' \mid S_{(l)}, S_{(u)}, S_{(l)} < \{S_i'\}_{i \in [u-l-1]} < S_{(u)}).$$

In the final analysis, $S_{(l)}$ and $S_{(u)}$ will represent the scores at consecutive bin boundaries, which define the binning scheme. Lemma 4.2 is similar to Lemma 4.1, but with conditioning on all bin boundaries (order statistics) simultaneously. To state it concisely, define $S_{(0)} := 0$ and $S_{(n+1)} := 1$ as fixed hypothetical 'order statistics'.

Lemma 4.2. *Fix any $B-1$ indices $k_1, k_2, \ldots k_{B-1}$ such that $0 = k_0 < k_1 < \ldots < k_B = n+1$. For any $b \in [B]$, the conditional density of the unordered S_i values between the order statistics $S_{(k_{b-1})}, S_{(k_b)}$, $f(S_{\{k_{b-1}+1\}}, \ldots, S_{\{k_b-1\}} \mid S_{(k_0)}, \ldots, S_{(k_B)})$, is identical to the conditional density*

$$\begin{aligned} f(S_1', \ldots, S_{k_b-k_{b-1}-1}' \mid\ & S_{(k_0)}, \ldots, S_{(k_B)}, \\ & \textit{for every } i \in [k_b - k_{b-1} - 1],\ S_{(k_{b-1})} < S_i' < S_{(k_b)})) \end{aligned}$$

of independent random variables $S_i' \sim Q_S$.

4.3.2 Main results

UMD is described in Algorithm 4.1 (in the description, $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denote the floor and ceiling operators respectively). UMD takes input $(g, \mathcal{D}_n)$ and outputs h. There is a small difference between UMD as stated and the proposal by Zadrozny and Elkan (2001). The original version also uses the calibration points that define the bin boundaries for bias estimation — this corresponds to replacing line 12 with

$$\text{line 12: } \widehat{\Pi}_b \leftarrow \text{Mean}(Y_{(l+1)}, \ldots, Y_{(u-1)}, \boldsymbol{Y_{(u)}}), \text{for } b < B.$$

The two algorithms are virtually the same; after stating the calibration guarantee for UMD, we show the result for the original proposal as a corollary.

By construction, every bin defined by UMD has at least $\lfloor n/B \rfloor - 1$ many points for mean estimation. Thus, UMD effectively 'uses' only $B-1$ points for bin formulation using quantile estimation. We prove the following calibration guarantee for UMD in Appendix 4.A.

Algorithm 4.1 UMD: Uniform-mass binning without sample splitting

```
Input: Scoring function $g : \mathcal{X} \to [0, 1]$, #bins $B$, calibration data $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$
Output: Approximately calibrated function $h$
$(S_1, S_2, \ldots, S_n) \leftarrow (g(X_1), g(X_2), \ldots, g(X_n))$
$(S_{(1)}, S_{(2)}, \ldots, S_{(n)}) \leftarrow$ order-stats$(S_1, S_2, \ldots, S_n)$
$(Y_{(1)}, Y_{(2)}, \ldots, Y_{(n)}) \leftarrow (Y_1, Y_2, \ldots, Y_n)$ ordered as per the ordering of $(S_{(1)}, S_{(2)}, \ldots, S_{(n)})$

$\Delta \leftarrow (n + 1)/B$
$\widehat{\Pi} \leftarrow$ empty array of size $B$
$A \leftarrow$ 0-indexed array$([0, \lceil \Delta \rceil, \lceil 2\Delta \rceil, \ldots, n + 1])$
for $b \leftarrow 1$ to $B$ do
    $l \leftarrow A_{b-1}$
    $u \leftarrow A_b$
    $\widehat{\Pi}_b \leftarrow$ Mean$(Y_{(l+1)}, Y_{(l+2)}, \ldots, Y_{(u-1)})$
end for
$(S_{(0)}, S_{(n+1)}) \leftarrow (0, 1)$
$h(\cdot) \leftarrow \sum_{b=1}^{B} \mathbb{1}\left\{S_{(A_{b-1})} \leqslant g(\cdot) < S_{(A_b)}\right\} \widehat{\Pi}_b$
```

Theorem 4.1. *Suppose $g(X)$ is absolutely continuous with respect to the Lebesgue measure and $n \geqslant 2B$. UMD is (ϵ, α)-conditionally calibrated for any $\alpha \in (0, 1)$ and*

$$\epsilon = \sqrt{\frac{\log(2B/\alpha)}{2(\lfloor n/B \rfloor - 1)}}. \tag{4.8}$$

Further, for every distribution P, w.p. $1 - \alpha$ over the calibration data $\mathcal{D}_n$, for all $p \in [1, \infty)$, ℓ_p-ECE$(h) \leqslant \epsilon$.

Note that since UMD is (ϵ, α)-conditionally calibrated, it is also (ϵ', α)-conditionally calibrated for any $\epsilon' \in (\epsilon, 1)$. The absolute continuity requirement for $g(X)$ can be removed with a randomization trick discussed in Section 4.C.1, to make the result fully DF. The proof sketch is as follows. Given the bin boundaries, the scores in each bin are independent, as shown by Lemma 4.2. We use this to conclude that the Y_i values in each bin b are independent and distributed as Bern$(\mathbb{E}[Y \mid \mathcal{B}(X) = b])$. The average of the Y_i values thus concentrates around $\mathbb{E}[Y \mid \mathcal{B}(X) = b]$. Since each bin has at least $(\lfloor n/B \rfloor - 1)$ points, Hoeffding's inequality along with a union bound across bins gives conditional calibration for the value of ϵ in (4.8).

The convenient property that every bin has at least $\lfloor n/B \rfloor - 1$ calibration points for mean estimation is not satisfied deterministically even if we used the true quantiles of $g(X)$. In fact, as long as $B = o(n)$, the ϵ in (4.8) approaches the ϵ we would get if all the data was used for bias estimation, with at least $\lfloor n/B \rfloor$ points in each bin:

$$\text{if } B = o(n), \ \lim_{n \to \infty} \left| \sqrt{\frac{\log(2B/\alpha)}{2(\lfloor n/B \rfloor - 1)}} - \sqrt{\frac{\log(2B/\alpha)}{2(\lfloor n/B \rfloor)}} \right| = 0.$$

In comparison to the clean proof sketch above, UMS requires a tedious multi-step analysis:

1. Suppose the sizes of the two splits are n_1 and n_2. Performing reliable quantile estimation on the first split of the data requires $n_1 = \Omega(B\log(B/\alpha))$ (Kumar et al., 2019, Lemma 4.3)).
2. The estimated quantiles have the guarantee that the *expected* number of points falling into a bin, on the second split is $\geqslant n_2/2B$. A high probability bound is used to lower bound the actual number of points in each bin. This lower bound is $(n_2/2B) - \sqrt{n_2\log(2B/\alpha)/2}$ (Gupta et al., 2020, Theorem 5).

This multi-step analysis leads to a loose bound due to constants stacking up, as discussed in Section 4.2.

A guarantee for the original UMD procedure follows as an immediate corollary of Theorem 4.1. This is because the modification to line 12 can change every estimate $\widehat{\Pi}_b$ by at most $1/(\lfloor n/B \rfloor)$ due to the following fact regarding averages: for any $b \in \mathbb{N}, a \in \{0, 1, \ldots, b\}$,

$$\max\left(\left|\frac{a}{b+1} - \frac{a}{b}\right|, \left|\frac{a+1}{b+1} - \frac{a}{b}\right|\right) \leqslant \frac{1}{b+1}. \tag{4.9}$$

Using (4.9), we prove the following corollary in Appendix 4.A.

Corollary 4.1. *Suppose $g(X)$ is absolutely continuous with respect to the Lebesgue measure and $n \geqslant 2B$. The original UMD algorithm (Zadrozny and Elkan, 2001) is (ϵ, α)-conditionally calibrated for any $\alpha \in (0, 1)$ and*

$$\epsilon = \sqrt{\frac{\log(2B/\alpha)}{2(\lfloor n/B \rfloor - 1)}} + \frac{1}{\lfloor n/B \rfloor}. \tag{4.10}$$

Further, for every distribution P, w.p. $1 - \alpha$ over the calibration data $\mathcal{D}_n$, for all $p \in [1, \infty)$, ℓ_p-ECE$(h) \leqslant \epsilon$.

As claimed in Section 4.2.2, if $(n, \alpha, B) = (2900, 0.1, 10)$, (4.10) gives $\epsilon < 0.1$. The difference between (4.10) and (4.8) is small. For example, we computed that if $\epsilon \leqslant 0.1$, $\alpha \leqslant 0.5$, $B \geqslant 5$, then (4.8) requires $n/B \geqslant 150$, and thus the additional term in (4.10) is at most 0.007. Likewise, in practice, we expect both versions to perform similarly.

At the end of the day, a practitioner may ask: “Given n points for recalibration, how should I use Theorem 4.1 to decide B?” Smaller B gives better bounds on ϵ, but larger B implicitly means that the h learnt is sharper. As n becomes higher, one may like to have higher sharpness (higher B), but at the same time more precise calibration (lower ϵ and thus lower B). We provide a (subjective) discussion on how to balance these two requirements.

First, we suggest fixing a rough domain-dependent probability of failure α. Since the dependence of ϵ on α in (4.8) is $\log(1/\alpha)$, small changes in α do not affect ϵ too much. Typically, 10-20% failure rate is acceptable, so let us set $\alpha = 0.1$. (For a highly sensitive domain, one can set $\alpha = 0.01$.) Then, constraint (4.8) roughly translates to $\epsilon = \sqrt{B\log(20B)/2n}$. For a fixed n, this is a relationship between ϵ and B, that can be plotted as a curve with B as the independent parameter and ϵ as the dependent parameter. Finally, one can eyeball the curve to identify a B.

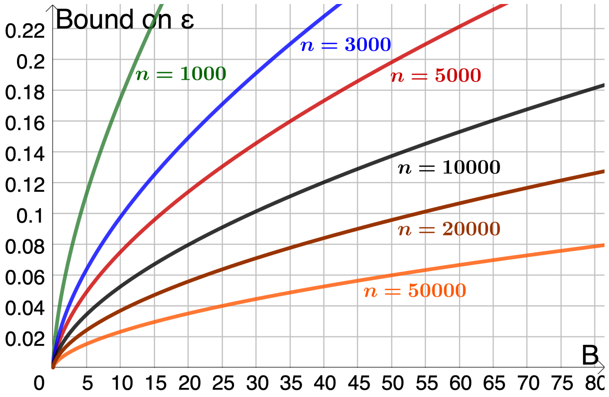


Figure 4.2: Plots displaying the relationship (4.8) between ϵ and B for $\alpha = 0.1$ and different values of n. Some indicative suggestions based on the plot: if $n = 1$K, choose $B = 5$ (gives $\epsilon \leqslant 0.12$); if $n = 5$K, choose $B = 10$ (gives $\epsilon \leqslant 0.08$); if $n = 20$K, choose $B = 22$ (gives $\epsilon \leqslant 0.06$).

We plot such curves in Figure 4.2 for a range of values of n. The caption shows examples of how one can choose B to balance calibration (small ϵ) and sharpness (high B).

While (ϵ, α)-conditional calibration implies (ϵ, α)-marginal calibration, we expect to have marginal calibration with smaller ϵ. Such an improved guarantee can be shown if the bin biases $\widehat{\Pi}_b$ estimated by Algorithm 4.1 are distinct. In Appendix 4.C, we propose a randomized version of UMD (Algorithm 4.2) which guarantees uniqueness of the bin biases. Algorithm 4.2 satisfies the following calibration guarantee (proved in Appendix 4.A).

Theorem 4.2. *Suppose $n \geqslant 2B$ and let $\delta > 0$ be an arbitrarily small randomization parameter. Algorithm 4.2 is (ϵ_1, α)-marginally and (ϵ_2, α)-conditionally calibrated for any $\alpha \in (0, 1)$,*

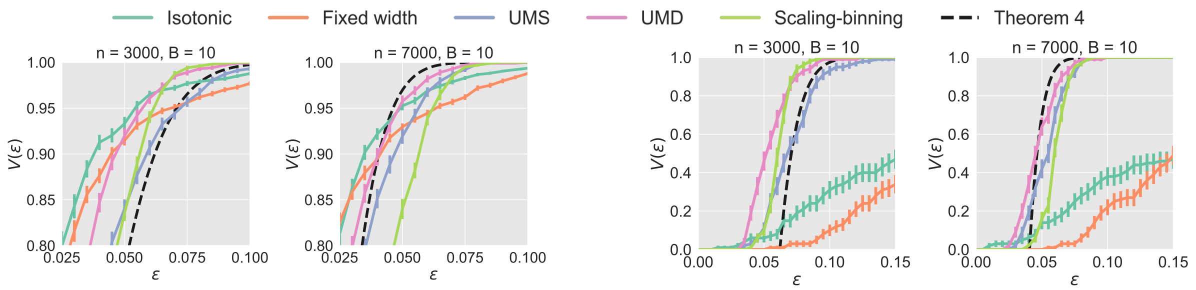
$$\epsilon_1 = \sqrt{\frac{\log(2/\alpha)}{2(\lfloor n/B \rfloor - 1)}} + \delta, \, \epsilon_2 = \sqrt{\frac{\log(2B/\alpha)}{2(\lfloor n/B \rfloor - 1)}} + \delta. \tag{4.11}$$

Further, for every distribution P, (a) w.p. $1 - \alpha$ over the calibration data $\mathcal{D}_n$, for all $p \in [1, \infty)$, ℓ_p-ECE$(h) \leqslant \epsilon_2$, and (b) $\mathbb{E}_{\mathcal{D}_n}[\ell_p\text{-ECE}(h)] \leqslant \sqrt{B/2n} + \delta$ for all $p \in [1, 2]$.

In the proof, we use the law of total expectation to avoid taking a union bound in the marginal calibration result; this gives a $\sqrt{\log(2/\alpha)}$ term in ϵ_1 instead of the $\sqrt{\log(2B/\alpha)}$ in ϵ_2. Theorem 4.2 also does not require absolute continuity of $g(X)$. As claimed in Section 4.2.2, if $(n, \alpha, B) = (1500, 0.1, 10)$, (4.11) gives $\epsilon_1 < 0.1$ (for small enough δ).

4.4 Simulations

We perform illustrative simulations on the CREDIT dataset with two goals: (a) to compare the performance of UMD to other binning methods and (b) to show that the guarantees we have



(a) Marginal validity plots.

(b) Conditional validity plots.

Figure 4.3: UMD performs competitively on the CREDIT dataset. The guarantee of Theorem 4.2 closely matches empirical behavior.

shown are reasonably tight, and thus, practically useful.[4] In addition to validity plots, which assess marginal calibration, we use conditional validity plots, that assess conditional calibration. Let $V : [0, 1] \to [0, 1]$ be given by $V(\epsilon) = \mathbb{P}(\forall r \in \text{Range}(h), |\mathbb{E}[Y \mid h(X) = r] - r| \leqslant \epsilon)$. Given a test set $\mathcal{D}_{\text{test}}$, we first compute $\mathbb{E}_{\widehat{P}}[Y \mid h(X) = h(x)]$ (defined in (4.6)), and then estimate $V(\epsilon)$ as

$$\widehat{V}(\epsilon) = \mathbb{1}\left\{\max_{(X_i,Y_i)\in\mathcal{D}_{\text{test}}} \left|\mathbb{E}_{\widehat{P}}[Y \mid h(X) = h(X_i)] - h(X_i)\right| \leqslant \epsilon\right\}.$$

For a single $\mathcal{D}_n$ and $\mathcal{D}_{\text{test}}$, $\widehat{V}(\epsilon)$ is either 0 or 1. Thus to estimate $V(\epsilon)$, we average $\widehat{V}(\epsilon)$ across multiple calibration and test sets. The mean±std-dev-of-mean of the $\widehat{V}(\epsilon)$ values are plotted as ϵ varies. This gives us a conditional validity plot. It is easy to see that the conditional validity plot is uniformly dominated by the (marginal) validity plot.

The experimental protocol for CREDIT is described in Section 4.2.2. In our experiments, we used the randomized version of UMD (Algorithm 4.2). Figure 4.3 presents validity plots for UMD, UMS, fixed-width binning, isotonic regression, scaling-binning, along with the Theorem 4.2 curve for $n = 3K$ and $n =$ 7K. In Appendix 4.D, we also present plots for $n = 1K$ and $n =$ 5K. Fixed-width binning refers to performing binning with equally spaced bins ($[0, 1/B), \ldots, [1-1/B, 1]$). UMS uses a 50:50 split of the calibration data. We do not rescale in scaling-binning, since it is already done on split B (for all compared procedures) — instead the comparison is between averaging the predictions of the scaling method (as is done in scaling-binning), against averaging the true outputs in each bin (as is done by all other methods). To have a fair comparison, we use double dipping for scaling-binning (thus scaling-binning and UMD are identical except what is being averaged). We make the following observations:

- Isotonic regression and fixed-width binning perform well for marginal calibration, but fail for conditional calibration. This is because both these methods tend to have bins with skewed masses, leading to small ϵ in bins with many points, and high ϵ in bins with few points.
- Scaling-binning is competitive with UMD for $n =$ 3K, $\epsilon > 0.05$. If $n =$ 7K or $\epsilon \leqslant 0.05$,

[4] Relevant code can be found at https://github.com/aigen/df-posthoc-calibration

UMD outperforms scaling-binning. In Appendix 4.D, we show that for $n = 1$K, scaling-binning is nearly the best method.

- UMD always performs better than UMS, and the performance of UMD is almost perfectly explained by the theoretical guarantee. Paradoxically, for $n = 7$K, the theoretical curve *crosses* the validity plot for UMD. This can occur since validity plots are based on a finite sample estimate of $\mathbb{E}\left[Y \mid h(X)\right]$, and the estimation error leads to slight *underestimation* of validity. This phenomenon is the same as the bias of plugin ECE estimators, and is discussed in detail in the last paragraph of Section 4.2.1. The curve-crossing shows that Theorem 4.2 is so precise that 5K test points are insufficient to verify it.

Overall, our experiment indicates that UMD performs competitively in practice and our theoretical guarantee closely explains its performance.

4.5 Conclusion

We used the Markov property of order statistics to prove distribution-free calibration guarantees for the popular uniform-mass binning method of Zadrozny and Elkan (2001). We proposed a novel assessment tool called validity plots, and used this tool to demonstrate that our theoretical bound closely tails empirical performance on a UCI credit default dataset. To the best of our knowledge, we demonstrated for the first time that it is possible to show informative calibration guarantees for binning methods that double dip the data (to both estimate bins and the probability of $Y = 1$ in a bin). Popular calibration methods such as isotonic regression (Zadrozny and Elkan, 2002), probability estimation trees (Provost and Domingos, 2003), random forests (Breiman, 2001) and Bayesian binning (Naeini et al., 2015) perform exactly this style of double dipping. We thus open up the exciting possibility of providing DF calibration guarantees for one or more of these methods.

Another recent line of work for calibration in data-dependent groupings, termed as multicalibration, uses a discretization step similar to fixed-width binning (Hébert-Johnson et al., 2018). Our uniform-mass binning techniques can potentially be extended to multicalibration. A number of non-binned methods for calibrating neural networks have displayed good performance on some tasks (Guo et al., 2017; Kull et al., 2017; Lakshminarayanan et al., 2017). However, the results of Gupta et al. (2020) imply that these methods cannot have DF guarantees. Examining whether they have guarantees under some (weak) distributional assumptions is also interesting future work.

Appendices for Chapter 4

4.A Proofs

4.A.1 Proof of Proposition 4.1

Define the random variables $u(X) = |\mathbb{E}[Y \mid h(X)] - h(X)|^p$ and $v(X) = 1$. Then, by Hölder's inequality for $r = q/p$ and $s = (1 - 1/r)^{-1}$,

$$\begin{aligned}(\ell_p\text{-ECE}(h))^p &= \mathbb{E}[u(X)] \\ &= \mathbb{E}[|u(X)v(X)|] \\ &\leqslant \mathbb{E}[|u(X)|^r]^{1/r}\,\mathbb{E}[|v(X)|^s]^{1/s} \\ &= \mathbb{E}[|u(X)|^r]^{1/r} \\ &= \mathbb{E}[|\mathbb{E}[Y \mid h(X)] - h(X)|^q]^{p/q} \\ &= (\ell_q\text{-ECE}(h))^p,\end{aligned}$$

which proves (4.5). If h satisfies (4.3), then $u(X) \leqslant \epsilon^p$ a.s. Thus $\ell_p\text{-ECE}(h) = \mathbb{E}[u(X)]^{1/p} \leqslant \epsilon$. □

4.A.2 Proof of Lemma 4.1

Let F denote the cdf corresponding to f. The structure of the proof is as follows:

- We first compute the conditional density of the order statistics $S_{(l+1)}, S_{(l+2)}, \dots, S_{(u-1)}$, given $S_{(l)}$ and $S_{(u)}$, in terms of f and F (the expression for this is (4.15)). The basic building block for this computation is a result on the conditional density of order statistics given a single order statistic (equation (4.12)).
- Next, we compute the conditional density of the order statistics of the independent random variables $\{S'_i\}_{i\in[u-l-1]}$, given $S_{(l)}$, $S_{(u)}$, and $S_{(l)} < S'_i < S_{(u)}$ for all $i \in [u-l-1]$ (the expression for this is (4.16)).
- We verify that (4.15) and (4.16) are identical, which shows that the conditional density of the order statistics matches. Finally, we conclude that the unordered random variables must themselves have the same conditional density. This completes the argument.

Let $0 \leqslant s_1 < \ldots < s_{l-1} < a < s_{l+1} < \ldots < s_n \leqslant 1$. The conditional density of all the order statistics given $S_{(l)}$

$$f(S_{(1)} = s_1, S_{(2)} = s_2, \ldots, S_{(l-1)} = s_{l-1}, S_{(l+1)} = s_{l+1}, \ldots, S_{(n)} = s_n \mid S_{(l)} = a)$$

is given by

$$\left((l-1)!\ \Pi_{i=1}^{l-1} \frac{f(s_i)}{F(a)}\right) \cdot \left((n-l)!\ \Pi_{i=l}^{n} \frac{f(s_i)}{1-F(a)}\right).$$

For one derivation, see Ahsanullah et al. (2013, Chapter 5, equation (5.2)). This implies that the order statistics larger than $S_{(l)}$ are independent of the order statistics smaller than $S_{(l)}$ given $S_{(l)}$, and

$$f(S_{(l+1)} = s_{l+1}, \ldots, S_{(n)} = s_n) \mid S_{(l)} = a) = \left((n-l)!\ \Pi_{i=l+1}^{n} \frac{f(s_i)}{1-F(a)}\right). \tag{4.12}$$

Suppose we draw $n-l$ independent samples $T_1, T_2, \ldots, T_{n-l}$ from the distribution whose density is given by

$$g(s) = \begin{cases} \frac{f(s)}{1-F(a)} & \text{if } s \in [a, 1], \\ 0 & \text{otherwise.} \end{cases}$$

(This is the conditional density of S given $S > S_{(l)} = a$ where S is an independent random variable distributed as Q_S.) Consider the order statistics $T_{(1)}, T_{(2)}, \ldots, T_{(n-l)}$ of these $n-l$ samples. It is a standard result — for example, see Arnold et al. (2008, Chapter 2, equation (2.2.3)) — that the density of the order statistics is

$$g(T_{(1)} = s_{l+1}, T_{(2)} = s_{l+2}, \ldots, T_{(n-l)} = s_n) = (n-l)!\ \Pi_{i=1}^{n-l} g(s_{l+1}),$$

which is identical to (4.12). Thus we can see the following fact:

$$\begin{gathered}\text{the density of the order statistics larger than } S_{(l)}\text{, given } S_{(l)} = a, \\ \text{is the same as the density of the order statistics } T_{(1)}, T_{(2)}, \ldots, T_{(n-l)}.\end{gathered} \tag{4.13}$$

Now consider the distribution of the order statistics $T_{(1)}, T_{(2)}, \ldots, T_{(u-l-1)}$ given $T_{(u-l)}$. Let $0 < s_{l+1} < \ldots < s_{u-1} < b \leqslant 1$. Using the same series of steps that led to equation (4.12), we have

$$\begin{aligned} g(T_{(1)} = s_{l+1}, T_{(2)} = s_{l+2}, \ldots, T_{(u-l-1)} = s_{u-1} \mid T_{(u-l)} = b) \\ = (u-l-1)!\ \Pi_{i=1}^{u-l-1} \frac{g(s_{l+i})}{G(b)}, \end{aligned} \tag{4.14}$$

where G is the cdf of g:

$$G(s) = \begin{cases} \frac{F(s)-F(a)}{1-F(a)} & \text{if } s \in [a, 1], \\ 0 & \text{if } s \in (-\infty, a), \\ 1 & \text{if } s \in (1, \infty). \end{cases}$$

Due to fact (4.13), the density of $(T_{(1)}, \ldots, T_{(u-l-1)})$ given $T_{(u-l)} = b$ is the same as the density of $(S_{(l+1)}, \ldots, S_{(u-1)})$ given $S_{(u)} = b$ and $S_{(l)} = a$. Thus,

$$f(S_{(l+1)} = s_{l+1}, \ldots, S_{(u-1)} = s_{u-1} \mid S_{(l)} = a, S_{(u)} = b) = (u-l-1)!\, \Pi_{i=1}^{u-l-1} \frac{g(s_{l+i})}{G(b)}.$$

Writing g and G in terms of f and F, we get

$$f(S_{(l+1)} = s_{l+1}, \ldots, S_{(u-1)} = s_{u-1} \mid S_{(l)} = a, S_{(u)} = b) = (u-l-1)!\, \Pi_{i=1}^{u-l-1} \frac{f(s_{l+i})}{F(b) - F(a)}. \tag{4.15}$$

Now consider the independent random variables $\{Z_i\}_{i=1}^{u-l-1}$, where the density of each Z_i is the same as the conditional density of S'_i, given $S_{(l)} = a < S'_i < b = S_{(u)}$.

Thus the density h of each Z_i is given by

$$h(s) = \begin{cases} \frac{f(s)}{F(b)-F(a)} & \text{if } s \in [a, b]\ , \\ 0 & \text{otherwise.} \end{cases}$$

The density of the order statistics $Z_{(1)}, \ldots, Z_{(u-l-1)}$ is given by

$$h(Z_{(1)} = s_{l+1}, \ldots, Z_{(u-l-1)} = s_{u-1}) = (u-l-1)!\, \Pi_{i=1}^{u-l-1} h(s_{l+i}), \tag{4.16}$$

which exactly matches the right hand side of (4.15). Thus,

$$\begin{aligned}
&f(S_{(l+1)} = s_{l+1}, \ldots, S_{(u-1)} = s_{u-1} \mid S_{(l)} = a, S_{(u)} = b) \\
&= h(Z_{(1)} = s_{l+1}, \ldots, Z_{(u-l-1)} = s_{u-1}) \\
&= f(S'_{(1)} = s_{l+1}, \ldots, S'_{(u-l-1)} = s_{u-1} \mid S_{(l)} = a, S_{(u)} = b, \forall\, i \in [u-l-1], S_{(l)} < S'_i < S_{(u)}).
\end{aligned}$$

Since the conditional densities of the order statistics match, the conditional densities of the unordered random variables must also match. This gives us the claimed result.

□

4.A.3 Proof of Lemma 4.2

The sequence of order statistics $S_{(1)}, S_{(2)}, \ldots, S_{(n)}$ form a Markov chain (Arnold et al., 2008, Theorem 2.4.3). Thus

$$\left(S_{(k_{i-1}+1)}, \ldots, S_{(k_i - 1)} \perp\!\!\!\perp S_{(k_0)}, \ldots, S_{(k_{i-2})}, S_{(k_{i+1})}, \ldots, S_{(k_B)}\right) \mid S_{(k_{i-1})}, S_{(k_i)}.$$

Consequently, for the unordered set of random variables $S_{\{k_{i-1}+1\}}, \ldots, S_{\{k_i - 1\}}$, we have:

$$\left(S_{\{k_{i-1}+1\}}, \ldots, S_{\{k_i - 1\}} \perp\!\!\!\perp S_{(k_0)}, \ldots, S_{(k_{i-2})}, S_{(k_{i+1})}, \ldots, S_{(k_B)}\right) \mid S_{(k_{i-1})}, S_{(k_i)}.$$

Thus,

$$f(S_{\{k_{i-1}+1\}}, \ldots, S_{\{k_i - 1\}} \mid S_{(k_0)}, \ldots, S_{(k_B)}) = f(S_{\{k_{i-1}+1\}}, \ldots, S_{\{k_i - 1\}} \mid S_{(k_{i-1})}, S_{(k_i)}).$$

Using Lemma 4.1, the result follows. □

4.A.4 Proof of Theorem 4.1

For $b \in \{0, 1, \ldots, B\}$, define $k_b = \lceil b(n+1/B) \rceil$. Let $S_{(0)} := 0$ and $S_{(n+1)} := 1$ be fixed hypothetical 'order-statistics'. The rest of this proof is conditional on the observed set $\mathcal{S} := (S_{(k_1)}, S_{(k_2)}, \ldots, S_{(k_{B-1})})$. (Marginalizing over $\mathcal{S}$ gives the theorem result as stated.) Let $\mathcal{B} : \mathcal{X} \to [B]$ be the binning function: for all x, $\mathcal{B}(x) = b \iff S_{(k_{b-1})} \leqslant g(x) < S_{(k_b)}$. Note that given $\mathcal{S}$, the binning function $\mathcal{B}$ is deterministic. In particular, this means that for every $b \in [B]$, $\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right]$ is a fixed number that is not random on the calibration data or (X, Y).

Let us fix some $b \in [B]$ and denote $l = k_{b-1}, u = k_b$. By Lemma 4.2, the scores $S_{\{l+1\}}, \ldots, S_{\{u-1\}}$ are independent and identically distributed given $\mathcal{S}$, and the conditional distribution of each of them equals that of $g(X)$ given $\mathcal{B}(X) = b$. Thus $Y_{\{l+1\}}, Y_{\{l+2\}}, \ldots, Y_{\{u-1\}}$ are independent and identically distributed given $\mathcal{S}$, and the conditional distribution of each of them is Bernoulli($\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right]$). Thus for any $t \in (0, 1)$, by Hoeffding's inequality, with probability at least $1 - t$,

$$\left|\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] - \widehat{\Pi}_b\right| \leqslant \sqrt{\frac{\log(2/t)}{2\lfloor u - l - 1 \rfloor}} \leqslant \sqrt{\frac{\log(2/t)}{2(\lfloor n/B \rfloor - 1)}}. \tag{4.17}$$

The second inequality holds since for any b,

$$\begin{aligned} u - l &= k_b - k_{b-1} \\ &= \lfloor (b+1)(n+1)/B \rfloor - \lfloor b(n+1)/B \rfloor \\ &= \lfloor U + (n+1)/B \rfloor - \lfloor U \rfloor, \text{ where } U = b(n+1)/B, \\ &\geqslant \lfloor (n+1)/B \rfloor \geqslant \lfloor n/B \rfloor. \end{aligned}$$

Next, we set $t = \alpha/B$ in (4.17), and take a union bound over all $b \in B$. Thus, with probability at least $1 - \alpha$, the event

$$E: \qquad \text{for every } b \in [B], \left|\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] - \widehat{\Pi}_b\right| \leqslant \epsilon$$

occurs. To prove the final calibration guarantee, we need to change the conditioning from $\mathcal{B}(X)$ to $h(X)$. Specifically, we have to be careful about the possibility of multiple bins having the same $\widehat{\Pi}$ values, in which case, conditioning on $\mathcal{B}(X)$ and conditioning on $h(X)$ is not the same. Given that E occurs (which happens with probability at least $1 - \alpha$),

$$\begin{aligned} &|\mathbb{E}\left[Y \mid h(X)\right] - h(X)| \\ &= |\mathbb{E}\left[\mathbb{E}\left[Y \mid \mathcal{B}(X), h(X)\right] \mid h(X)\right] - h(X)| && \text{(applying tower rule)} \\ &= |\mathbb{E}\left[\mathbb{E}\left[Y \mid \mathcal{B}(X)\right] \mid h(X)\right] - h(X)| && (\mathbb{E}\left[Y \mid \mathcal{B}(X), h(X)\right] = \mathbb{E}\left[Y \mid \mathcal{B}(X)\right]) \\ &= |\mathbb{E}\left[\mathbb{E}\left[Y \mid \mathcal{B}(X)\right] - h(X) \mid h(X)\right]| \\ &= \left|\mathbb{E}\left[\mathbb{E}\left[Y \mid \mathcal{B}(X)\right] - \widehat{\Pi}_{\mathcal{B}(X)} \mid h(X)\right]\right| && \text{(by definition of } h\text{)} \\ &\leqslant \mathbb{E}\left[\left|\mathbb{E}\left[Y \mid \mathcal{B}(X)\right] - \widehat{\Pi}_{\mathcal{B}(X)}\right| \mid h(X)\right] && \text{(Jensen's inequality)} \\ &\leqslant \epsilon && \text{(since } E \text{ occurs).} \end{aligned}$$

This completes the proof of the conditional calibration guarantee. The ECE bound follows by Proposition 4.1. □

4.A.5 Proof of Corollary 4.1

Conditioned on $\mathcal{S}$ (defined in the proof of Theorem 4.1), for some $b \in [B]$, $l = k_{b-1}$ and $u = k_b$, we showed in the proof of Theorem 4.1 that with probability at least $1 - \alpha/B$,

$$\left|\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] - \text{Mean}(Y_{(l+1)}, Y_{(l+2)}, \ldots, Y_{(u-1)})\right| \leqslant \sqrt{\frac{\log(2B/\alpha)}{2(\lfloor n/B \rfloor - 1)}}.$$

Thus for $b \in [B-1]$,

$$\begin{aligned}\left|\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] - \widehat{\Pi}_b\right| &\leqslant \left|\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] - \text{Mean}(Y_{(l+1)}, Y_{(l+2)}, \ldots, Y_{(u-1)})\right| \\ &\quad + \left|\text{Mean}(Y_{(l+1)}, Y_{(l+2)}, \ldots, Y_{(u-1)}) - \text{Mean}(Y_{(l+1)}, Y_{(l+2)}, \ldots, Y_{(u)})\right| \\ &\leqslant \sqrt{\frac{\log(2B/\alpha)}{2(\lfloor n/B \rfloor - 1)}} + \frac{1}{\lfloor n/B \rfloor} \qquad \text{(by fact (4.9))} \\ &\leqslant \epsilon.\end{aligned}$$

The rest of the argument can be completed exactly as in the proof of Theorem (4.1) after equation (4.17). □

4.A.6 Proof of Theorem 4.2

Let $\{\widehat{\Pi}'_b\}_{b\in[B]}$ denote the the pre-randomization values of $\widehat{\Pi}_b$ as computed in line 13 of Algorithm 4.2. Due to the randomization in line (15), no two $\widehat{\Pi}_b$ values are the same. Formally, consider any two indices $1 \leqslant a \neq b \leqslant B$. Then, $\widehat{\Pi}_a = \widehat{\Pi}_b$ if and only if $\delta(V_a - V_b) = \widehat{\Pi}'_a - \widehat{\Pi}'_b$, which happens with probability zero. Thus for any $1 \leqslant a \neq b \leqslant B$, $\widehat{\Pi}_a \neq \widehat{\Pi}_b$ (with probability one).

The rest of the proof is conditional on $\mathcal{S}$, as defined in the proof of Theorem 4.1. (Marginalizing over $\mathcal{S}$ gives the theorem result as stated.) As noted in that proof, conditioning on $\mathcal{S}$ makes the binning function $\mathcal{B}$ deterministic, which simplifies the proof significantly.

First, we prove a per bin concentration bound for $\widehat{\Pi}_b$ of the form of (4.17). The δ randomization changes this bound as follows. For any $b \in [B], t \in (0,1)$, with probability at least $1 - t$,

$$\begin{aligned}\left|\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] - \widehat{\Pi}_b\right| &\leqslant \left|\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] - \widehat{\Pi}'_b\right| + \left|\widehat{\Pi}_b - \widehat{\Pi}'_b\right| \\ &\leqslant \sqrt{\frac{\log(2/t)}{2(\lfloor n/B \rfloor - 1)}} + \left|(1+\delta)^{-1}(\widehat{\Pi}'_b + \delta) - \widehat{\Pi}'_b\right| \\ &\qquad \text{(Hoeffding's inequaliity (4.17))} \\ &\leqslant \sqrt{\frac{\log(2/t)}{2(\lfloor n/B \rfloor - 1)}} + \delta. \qquad (4.18)\end{aligned}$$

Given this concentration bound for every bin, the (ϵ_2, α)-conditional calibration bound can be shown following the arguments in the proof of Theorem 4.1 after inequality (4.17). We now show the marginal calibration guarantee. Note that since no two $\widehat{\Pi}_b$ values are the same, $\mathcal{B}(X)$ is known given $\widehat{\Pi}_{\mathcal{B}(X)}$, and so $\mathbb{E}\left[Y \mid h(X)\right] = \mathbb{E}\left[Y \mid \mathcal{B}(X)\right]$. Thus,

$$\begin{aligned}
&\mathbb{P}(\left|\mathbb{E}\left[Y \mid h(X)\right] - h(X)\right| \leqslant \epsilon_1) \\
&= \sum_{b=1}^{B} \mathbb{P}(\left|\mathbb{E}\left[Y \mid h(X)\right] - h(X)\right| \leqslant \epsilon_1 \mid \mathcal{B}(X) = b)\, \mathbb{P}(\mathcal{B}(X) = b) && \text{(law of total probability)} \\
&= \sum_{b=1}^{B} \mathbb{P}(\left|\mathbb{E}\left[Y \mid \mathcal{B}(X)\right] - h(X)\right| \leqslant \epsilon_1 \mid \mathcal{B}(X) = b)\, \mathbb{P}(\mathcal{B}(X) = b) && (\mathbb{E}\left[Y \mid h(X)\right] = \mathbb{E}\left[Y \mid \mathcal{B}(X)\right]) \\
&= \sum_{b=1}^{B} \mathbb{P}(\left|\mathbb{E}\left[Y \mid \mathcal{B}(X)\right] - \widehat{\Pi}_{\mathcal{B}(X)}\right| \leqslant \epsilon_1 \mid \mathcal{B}(X) = b)\, \mathbb{P}(\mathcal{B}(X) = b) && \text{(by definition of } h\text{)} \\
&\geqslant \sum_{b=1}^{B} (1-\alpha)\, \mathbb{P}(\mathcal{B}(X) = b) && (t = \alpha \text{ in } (4.18)) \\
&= 1 - \alpha.
\end{aligned}$$

This proves (ϵ_1, α)-marginal calibration.

For the ECE bound, note that for every bin $b \in [B]$, $\widehat{\Pi}'_b$ is the average of at least $\lfloor n/B \rfloor - 1$ Bernoulli random variables with bias $\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right]$. We know the exact form of the variance of averages of Bernoulli random variables with a given bias, giving the following:

$$\mathrm{Var}(\widehat{\Pi}'_b) \leqslant \frac{\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right](1 - \mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right])}{\lfloor n/B \rfloor - 1} \leqslant \frac{1}{4(\lfloor n/B \rfloor - 1)}. \tag{4.19}$$

We now rewrite the expectation of the square of the ℓ_2-ECE in terms of $\mathrm{Var}(\widehat{\Pi}'_b)$. Recall that all expectations and probabilities in the entire proof are conditional on $\mathcal{S}$, so that $\mathcal{B}$ is known; the same is true for all expectations in the forthcoming panel of equations. To aid readability, when we apply the tower law, we are explicit about the remaining randomness in $\mathcal{D}_n$.

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}_n}\left[(\ell_2\text{-ECE}(h))^2\right] &= \mathbb{E}_{\mathcal{D}_n}\left[\mathbb{E}_{(X,Y)}\left[(\mathbb{E}\left[Y \mid h(X)\right] - h(X))^2 \mid \mathcal{D}_n\right]\right] \\
&= \mathbb{E}_{\mathcal{D}_n}\left[\sum_{b=1}^{B} (\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] - \widehat{\Pi}_b)^2 \mathbb{P}(\mathcal{B}(X) = b)\right] \\
&= \sum_{b=1}^{B} \mathbb{E}_{\mathcal{D}_n}\left[(\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] - \widehat{\Pi}_b)^2 \mathbb{P}(\mathcal{B}(X) = b)\right] \\
&= \sum_{b=1}^{B} \mathbb{E}_{\mathcal{D}_n}\left[(\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] - \widehat{\Pi}_b)^2\right] \mathbb{P}(\mathcal{B}(X) = b).
\end{aligned}$$

The first equality is by the tower rule. The second equality uses the same simplifications as the panel of equations used to prove the marginal calibration guarantee (law of total probability, using $\mathbb{E}\left[Y \mid h(X)\right] = \mathbb{E}\left[Y \mid \mathcal{B}(X)\right]$, and the definition of h). The third equality uses linearity of expectation. The fourth equality follows since $\mathcal{B}$ is deterministic given $\mathcal{S}$. Now note that

$$\mathbb{E}_{\mathcal{D}_n}(\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] - \widehat{\Pi}_b)^2 = \mathbb{E}_{\mathcal{D}_n}(\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] - \widehat{\Pi}'_b + \widehat{\Pi}'_b - \widehat{\Pi}_b)^2 \leqslant \text{Var}(\widehat{\Pi}'_b) + \delta^2,$$

since $\mathbb{E}\left[Y \mid \mathcal{B}(X) = b\right] = \mathbb{E}_{\mathcal{D}_n}(\widehat{\Pi}'_b)$ and $\left|\widehat{\Pi}'_b - \widehat{\Pi}_b\right| \leqslant \delta$ deterministically. Thus by bound (4.19),

$$\mathbb{E}_{\mathcal{D}_n}\left[(\ell_2\text{-ECE}(h))^2\right] \leqslant \sum_{b=1}^{B}\left(\frac{1}{4(\lfloor n/B\rfloor - 1)} + \delta^2\right)\mathbb{P}(\mathcal{B}(X) = b) = \frac{1}{4(\lfloor n/B\rfloor - 1)} + \delta^2 \leqslant \frac{B}{2n} + \delta^2.$$

The last inequality holds since $n \geqslant 2B$ implies that $\lfloor n/B\rfloor - 1 \geqslant n/2B$. Jensen's inequality now gives the final result:

$$\begin{aligned}\mathbb{E}_{\mathcal{D}_n}\left[\ell_2\text{-ECE}(h)\right] &\leqslant \sqrt{\mathbb{E}_{\mathcal{D}_n}\left[(\ell_2\text{-ECE}(h))^2\right]} && \text{(Jensen's inequality)}\\ &\leqslant \sqrt{\frac{B}{2n} + \delta^2} \leqslant \sqrt{\frac{B}{2n}} + \delta.\end{aligned}$$

The bound on $\mathbb{E}_{\mathcal{D}_n}\left[\ell_p\text{-ECE}(h)\right]$ for $p \in [1, 2)$ follows by Proposition 4.1. □

4.B Assessing the theoretical guarantee of UMS

We compute the number of calibration points n required to guarantee $(\epsilon, \alpha) = (0.1, 0.1)$-marginal calibration with $B = 10$ bins using UMS, based on Theorem 5 of Gupta et al. (2020). Following their notation, if the minimum number of calibration points in a bin is denoted as $N_{b^\star}$, then the Hoeffding-based bound on ϵ, with probablity of failure δ, is $\sqrt{\log(2B/\delta)/2N_{b^\star}}$. (The original bound is based on empirical-Berstein which is often tighter in practice, but Hoeffding is tighter in the worst case.) Let us set $\delta = \alpha/2 = 0.05$ since the remaining failure budget $\alpha/2$ is for the bin estimation to ensure that $N_{b^\star}$ is lower bounded. Thus, the requirement $\sqrt{\log(2 \cdot 10/0.05)/2N_{b^\star}} \leqslant \epsilon = 0.1$ translates roughly to $N_{b^\star} \geqslant 300$.

To ensure $N_{b^\star} \geqslant 300$, we define the bins to each have roughly $1/B$ fraction of the calibration points in the first split of the data. Lemma 4.3 (Kumar et al., 2019) shows that w.p. $\geqslant 1 - \delta$, the true mass of the estimated bins is at least $1/2B$, as long as the first split of the data has at least $cB\log(10B/\delta)$ points, for a universal constant c. The original proof is for a $c \geqslant 2000$, but let us suppose that with a tighter analysis it can be improved to (say) $c = 100$. Then for $\delta = \alpha/4 = 0.025$, the first split of the data must have at least $100 \cdot 10 \cdot \log(100/0.025) \geqslant 8000$ calibration points. Finally, we use Theorem 5 (Gupta et al., 2020) to bound $N_{b^\star}$. If n' is the cardinality of the second split (denoted as $|\mathcal{D}^2_{\text{cal}}|$ in the original result), then they show that for $\delta = 0.025$, $N_{b^\star} \geqslant n'/2B - \sqrt{n'/\log(2B/\delta)/2} \approx n'/20 - 1.8\sqrt{n'}$. Since we require $N_{b^\star} \geqslant 300$, we must have approximately $n' \geqslant 9500$. Overall, the theoretical guarantee for UMS requires $n \geqslant 17500$ points to guarantee $(0.1, 0.1)$-marginal calibration with 10 bins.

Algorithm 4.2 Randomized UMD

```
1:  **Input:** Scoring function $g : \mathcal{X} \to [0,1]$, #bins $B$, calibration data
    $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$,
    randomization parameter $\delta > 0$ (arbitrarily small)
2:  **Output:** Approximately calibrated function $h$
3:  $(U_1, U_2, \ldots, U_n) \sim \text{Unif}[0,1]^n$
4:  $(S_1, S_2, \ldots, S_n) \leftarrow (1+\delta)^{-1}(g(X_1) + \delta U_1, g(X_2) + \delta U_2, \ldots, g(X_n) + \delta U_n)$
5:  $(S_{(1)}, S_{(2)}, \ldots, S_{(n)}) \leftarrow$ order-stats$(S_1, S_2, \ldots, S_n)$
6:  $(Y_{(1)}, Y_{(2)}, \ldots, Y_{(n)}) \leftarrow (Y_1, Y_2, \ldots, Y_n)$ ordered as per the ordering of $(S_{(1)}, S_{(2)}, \ldots, S_{(n)})$

7:  $\Delta \leftarrow (n+1)/B$
8:  $\widehat{\Pi} \leftarrow$ empty array of size $B$
9:  $A \leftarrow$ 0-indexed array$([0, \lceil\Delta\rceil, \lceil 2\Delta\rceil, \ldots, n+1])$
10: **for** $b \leftarrow 1$ **to** $B$ **do**
11:     $l \leftarrow A_{b-1}$
12:     $u \leftarrow A_b$
13:     $\widehat{\Pi}_b \leftarrow \text{Mean}(Y_{(l+1)}, Y_{(l+2)}, \ldots, Y_{(u-1)})$
14:     $V_b \sim \text{Unif}[0,1]$
15:     $\widehat{\Pi}_b \leftarrow (1+\delta)^{-1}(\widehat{\Pi}_b + \delta V_b)$
16: **end for**
17: $(S_{(0)}, S_{(n+1)}) \leftarrow (0,1)$
18: $h(\cdot) \leftarrow \sum_{b=1}^{B} \mathbb{1}\left\{S_{(A_{b-1})} \leqslant (1+\delta)^{-1}(g(\cdot) + \delta U) < S_{(A_b)}\right\} \widehat{\Pi}_b$, for $U \sim \text{Unif}[0,1]$
```

4.C Randomized UMD

We now describe the randomized version of UMD (Algorithm 4.2) that is nearly identical to the non-randomized version in practice, but for which we are able to show better theoretical properties. In this sense, we view randomized UMD as a theoretical tool rather than a novel algorithm (nevertheless, all experimental results in this chapter use randomized UMD). Algorithm 4.2 takes as input a randomization parameter $\delta > 0$ which can be arbitrarily small, such as 10^{-20}. The specific lines that induce randomization, in comparison to Algorithm 4.1, are lines 3, 4, 14, 15 and 18. This δ perturbation leads to a better theoretical result than the non-randomized version — in comparison to Theorem 4.1, Theorem 4.2 does not require absolute continuity of $g(X)$ and provides an improved marginal calibration guarantee.

4.C.1 Absolute continuity of $g(X)$

In Theorem 4.1, we assumed that $g(X)$ is absolutely continuous with respect to the Lebesgue measure, or equivalently, it has a pdf. This may not always be the case. For example, X may contain atoms, or g may have discrete outputs in $[0,1]$. If $g(X)$ does not have a pdf, a simple randomization trick can be used to ensure that the results hold in full generality (we performed this randomization in our experiments as well).

First, we append the features X with Unif$[0,1]$ random variables U so that $(X, U) \sim P_X \times$ Unif$[0,1]$. Next, for an arbitrarily small value $\delta > 0$, such as 10^{-20}, we define $\widetilde{g} : \mathcal{X} \times [0,1] \to [0,1]$ as $\widetilde{g}(x,u) = (1+\delta)^{-1}(g(x) + \delta u)$. Thus for every x, $\widetilde{g}(x,\cdot)$ is arbitrarily close to $g(x)$, and we do not lose the informativeness of g. However, now $\widetilde{g}(X,U)$ is guaranteed to be absolutely continuous with respect to the Lebesgue measure. The precise implementation details are as follows: (a) to train, draw $(U_i)_{i\in[n]} \sim$ Unif$[0,1]^n$ and call Algorithm 4.1 with $\widetilde{g}, \{((X_i, U_i), Y_i)\}_{i\in[n]}$; (b) to test, draw a new Unif$[0,1]$ random variable for each test point. Algorithm 4.2 packages this randomization into the pseudocode; see lines 3, 4 and 18.

The above process is a technical way of describing the following intuitive methodology: "break ties among the scores arbitrarily but consistently". Lemmas 4.1 and 4.2 fail if two data points have $S_i = S_j$ and one of them is the order statistics we are conditioning on. However, if we fix an arbitrary secondary order through which ties can be broken even if $S_i = S_j$ or $S = S_i$, the lemmas can be made to go through. The noise term δU in $\widetilde{g}$ implicitly provides a strict secondary order.

4.C.2 Improved marginal calibration guarantee

The marginal calibration guarantee of Theorem 4.2 hinges on the bin biases $\widehat{\Pi}_b$ being unique. Lines 14 and 15 in Algorithm 4.2 ensure that this is satisfied almost surely by adding an infinitesimal random perturbation to each $\widehat{\Pi}_b$. This is identical to the technique described in Section 4.C.1. Due to the perturbation, the ϵ required to satisfy calibration as per equation (4.11) has an additional δ term. However the δ can be chosen to be arbitrarily small, and this term is inconsequential.

We make an informal remark that may be relevant to practitioners. In practice, we expect that the bin biases computed using Algorithm 4.1 are unique with high probability without the need for randomization. As long as the bin biases are unique, the marginal calibration and ECE guarantees of Theorem 4.2 apply to Algorithm 4.1 as well. Thus, the $\widehat{\Pi}$-randomization can be skipped if 'simplicity' or 'interpretability' is desired. Note that the $g(X)$ randomization (Section 4.C.1) is still crucial since we envision many practical scenarios where $g(X)$ is not absolutely continuous. In summary, randomized UMD uses a small random perturbation to ensure that (a) the score values and (b) the bin bias estimates, are unique. The particular randomization strategy we proposed is not special; any other strategy that achieves the aforementioned goals is sufficient (for example, using a (truncated) Gaussian random variable instead of uniform).

4.D Additional experiments

We present additional experiments to supplement those presented in the Chapter 4.

In Section 4.4, we compared UMD to other binning methods on the CREDIT dataset, for $n = 3$K and $n = 7$K. Here, we present plots for $n = 1$K and $n = 5$K (for easier comparison, we also show the plots for $n = 3$K and $n = 7$K). The marginal validity plots are in Figure 4.4, and the conditional validity plots are in Figure 4.5. Apart from additional evidence for the same observations made in Section 4.4, we also see some interesting behavior in the low sample case ($n = 1$K). First, the Theorem 4.2 curve does not explain performance as well as the other plots. We tried the Clopper Pearson exact confidence interval (Clopper and Pearson, 1934) instead of Hoeffding and obtained nearly identical results (plots not presented). It would be interesting to explore if a tighter guarantee can be shown for small sample sizes. Second, for $n = 1$K, scaling-binning performs better than UMD in both the marginal and conditional validity plots, and is competitive with isotonic regression in the marginal validity plot. This behavior occurs since in the small sample regime, while all other binning methods attempt to re-estimate the biases of the bins using very little data, scaling-binning relies on the statistical efficiency of the learnt g which was trained on 15K training points. A similar phenomenon was observed by Niculescu-Mizil and Caruana (2005) when comparing Platt scaling and isotonic regression: Platt scaling performs better at small sample sizes since it relies more on the underlying efficiency of g, compared to isotonic regression.

While the experiments considered so far use 10K points for training logistic regression, 5K points for Platt scaling, and between 0.5-10K points for binning, a practically common setting is where most points are used for training the base model, and a small fraction of points are used for recalibration. On recommendation of one of the ICML reviewers, we ran experiments with 14K points for training logistic regression, 1K for Platt scaling, and 1K for binning. The marginal and conditional validity plots for this experiment are displayed in Figure 4.6. We observe that these plots are very similar to the marginal and conditional validity plots in Figures 4.4 and 4.5 for $n = 1$K, and the same conclusions described in the previous paragraph can be drawn.

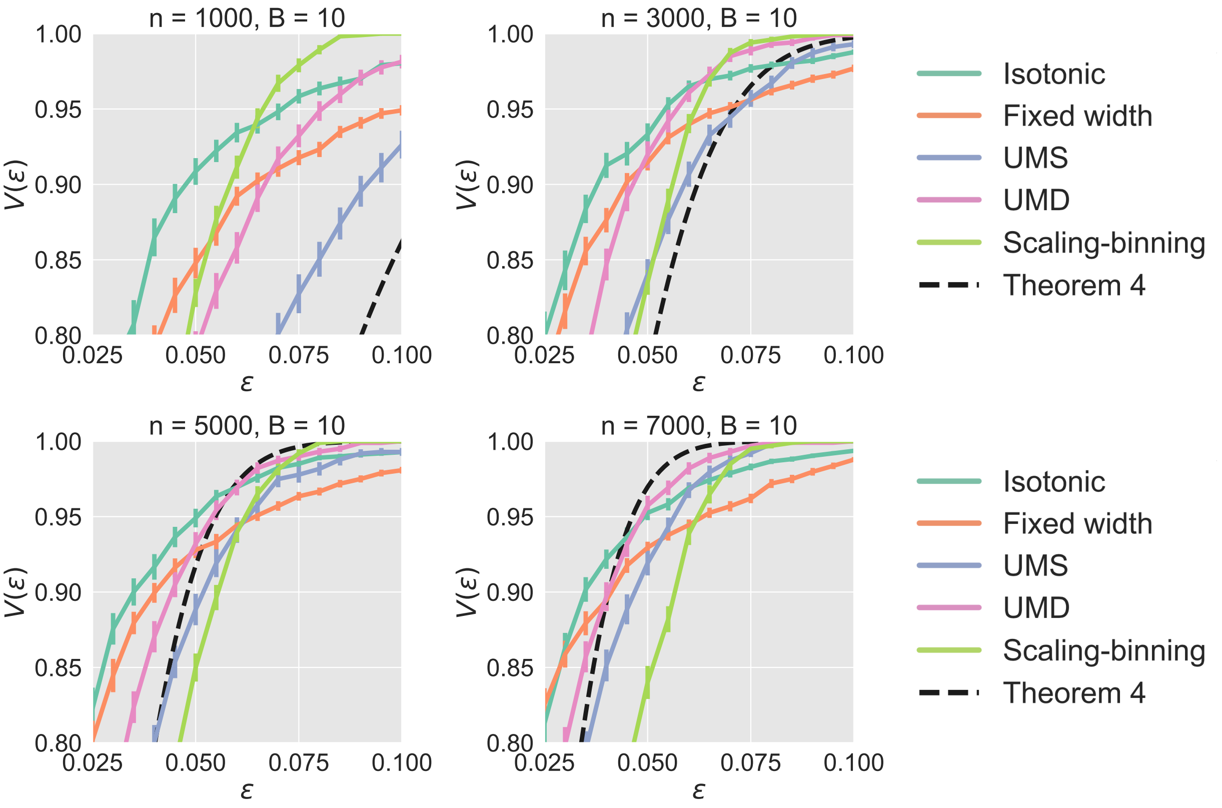


Figure 4.4: Marginal validity plots comparing UMD to other binning methods. The performance of UMD improves at higher values of n and ϵ, and the performance of UMD is closely explained by its theoretical guarantee. Isotonic regression and fixed-width binning perform well at small values of ϵ.

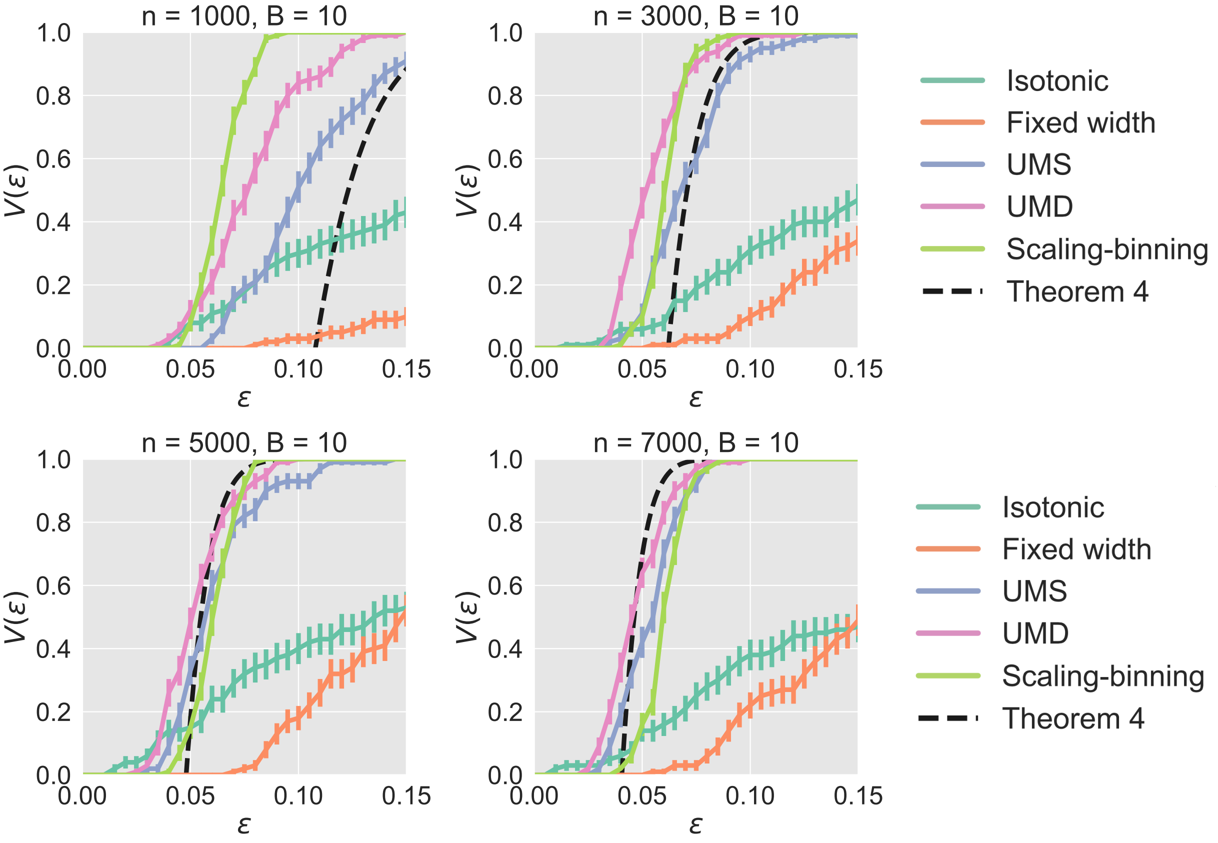
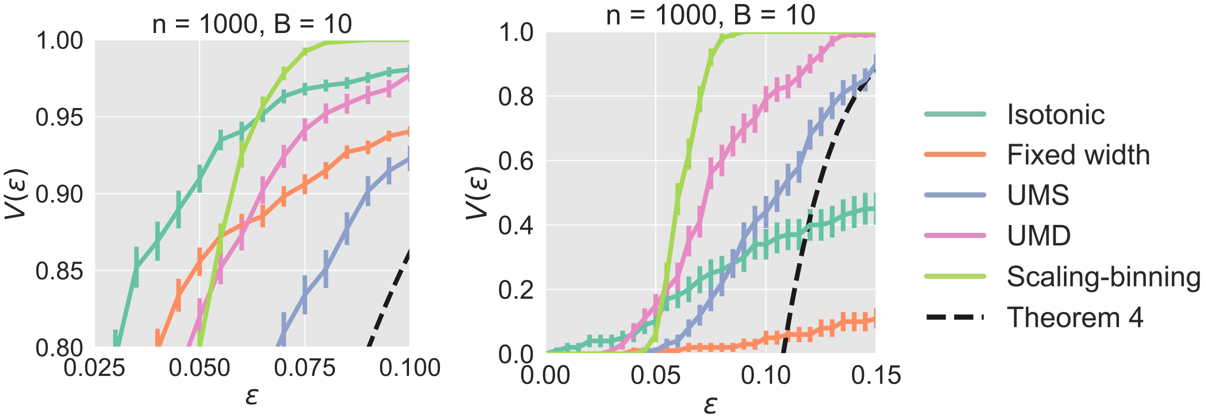


Figure 4.5: Conditional validity plots comparing UMD to other binning methods. UMD and scaling-binning are the best methods for conditional calibration at nearly all values of n, ϵ. Scaling-binning performs slightly better for small n whereas UMD performs slightly better for large n. The performance of UMD is closely explained by its theoretical guarantee.



(a) Marginal validity plot.

(b) Conditional validity plot.

Figure 4.6: Validity plots comparing UMD to other binning methods with fewer points used for recalibration. Namely, 14K points are used for training logistic regression, 1K for Platt scaling, and 1K for binning. Overall, scaling-binning performs quite well, since it relies on the underlying efficiency of logistic regression more than the other methods.

Chapter 5

Top-label calibration and multiclass-to-binary reductions

This chapter is based on Gupta and Ramdas (2022b).

A multiclass classifier is said to be top-label calibrated if the reported probability for the predicted class—the top-label—is calibrated, conditioned on the top-label. This conditioning on the top-label is absent in the closely related and popular notion of confidence calibration, which we argue makes confidence calibration difficult to interpret for decision-making. We propose top-label calibration as a rectification of confidence calibration. Further, we outline a multiclass-to-binary (M2B) reduction framework that unifies confidence, top-label, and class-wise calibration, among others. As its name suggests, M2B works by reducing multiclass calibration to numerous binary calibration problems, each of which can be solved using simple binary calibration routines. We instantiate the M2B framework with the well-studied histogram binning (HB) binary calibrator, and prove that the overall procedure is multiclass calibrated without making any assumptions on the underlying data distribution. In an empirical evaluation with four deep net architectures on CIFAR-10 and CIFAR-100, we find that the M2B + HB procedure achieves lower top-label and class-wise calibration error than other approaches such as temperature scaling. Code for this work is available at https://github.com/aigen/df-posthoc-calibration.

5.1 Introduction

In this chapter, we study calibration for multiclass classification, with $L \geqslant 3$ classes and the label $Y \in [L] := \{1, 2, \ldots, L \geqslant 3\}$. We assume all (training and test) data is drawn i.i.d. from a fixed distribution P, and denote a general point from this distribution as $(X, Y) \sim P$. Consider a typical multiclass predictor, $\mathbf{h} : \mathcal{X} \to \Delta^{L-1}$, whose domain is a feature space $\mathcal{X}$ and range Δ^{L-1} is the probability simplex in $\mathbb{R}^L$. A natural notion of calibration for $\mathbf{h}$, called *canonical calibration* is the following: for every $l \in [L]$, $P(Y = l \mid \mathbf{h}(X) = \mathbf{q}) = q_l$ (q_l denotes the l-th component of $\mathbf{q}$). However, canonical calibration becomes infeasible to achieve or verify once L is even 4 or 5 (Vaicenavicius et al., 2019). Thus, there is interest in studying statistically feasible relaxations of canonical notion, such as confidence calibration (Guo et al., 2017) and class-wise calibration (Kull et al., 2017).

In particular, the notion of confidence calibration (Guo et al., 2017) has been popular recently. A model is confidence calibrated if the following is true: "when the reported confidence for the predicted class is $q \in [0, 1]$, the accuracy is also q". In any practical setting, the confidence q is never reported alone; it is always reported along with the actual class prediction $l \in [L]$. One may expect that if a model is confidence calibrated, the following also holds: "when the class l is predicted with confidence q, the probability of the actual class being l is also q"? Unfortunately, this expectation is rarely met—there exist confidence calibrated classifier for whom the latter statement is grossly violated for all classes (Example 5.1). On the other hand, our proposed notion of top-label calibration enforces the latter statement. It is philosophically more coherent, because it requires conditioning on all relevant reported quantities (both the predicted top label and our confidence in it). In Section 5.2, we argue further that top-label calibration is a simple and practically meaningful replacement of confidence calibration.

In Section 5.3, we unify top-label, confidence, and a number of other popular notions of multiclass calibration into the framework of multiclass-to-binary (M2B) reductions. The M2B framework relies on the simple observation that each of these notions internally verifies binary calibration claims. As a consequence, each M2B notion of calibration can be achieved by solving a number of binary calibration problems. With the M2B framework at our disposal, all of the rich literature on binary calibration can now be used for multiclass calibration. We illustrate this by instantiating the M2B framework with the binary calibration algorithm of histogram binning or HB (Zadrozny and Elkan, 2001; Gupta and Ramdas, 2021). The M2B + HB procedure achieves state-of-the-art results with respect to standard notions of calibration error (Section 5.4). Further, we show that our procedure is provably calibrated for arbitrary data-generating distributions. The formal theorems are delayed to Appendices 5.B, 5.C, but an informal result is presented in Section 5.4.

5.2 Modifying confidence calibration to top-label calibration

Let $c : \mathcal{X} \to [L]$ denote a classifier or top-label predictor and $h : \mathcal{X} \to [0, 1]$ a function that provides a confidence or probability score for the top-label $c(X)$. The predictor (c, h) is said to be confidence calibrated (for the data-generating distribution P) if

$$\mathbb{P}(Y = c(X) \mid h(X)) = h(X). \tag{5.1}$$

In other words, when the reported confidence $h(X)$ equals $p \in [0, 1]$, then the fraction of instances where the predicted label is correct also equals p. Note that for an L-dimensional predictor $\mathbf{h} : \mathcal{X} \to \Delta^{L-1}$, one would use $c(\cdot) = \arg\max_{l \in [L]} h_l(\cdot)$ and $h(\cdot) = h_{c(\cdot)}(\cdot)$; ties are broken arbitrarily. Then $\mathbf{h}$ is confidence calibrated if the corresponding (c, h) satisfies (5.1).

Confidence calibration is most applicable in high-accuracy settings where we trust the label prediction $c(x)$. For instance, if a high-accuracy cancer-grade-prediction model predicts a patient as having "95% grade III, 3% grade II, and 2% grade I", we would suggest the patient to undergo an invasive treatment. However, we may want to know (and control) the number of non-grade-III patients that were given this suggestion incorrectly. In other words, is $\Pr($cancer is not grade III $|$ cancer is predicted to be of grade III with confidence $95\%)$ equal to 5%? It would appear that by focusing on the the probability of the predicted label, confidence calibration enforces such control.

However, as we illustrate next, confidence calibration fails at this goal by providing a guarantee that is neither practically interpretable, nor actionable. Translating the probabilistic statement (5.1) into words, we ascertain that confidence calibration leads to guarantees of the form: "if the confidence $h(X)$ in the top-label is 0.6, then the accuracy (frequency with which Y equals $c(X)$) is 0.6". Such a guarantee is not very useful. Suppose a patient P is informed (based on their symptoms X), that they are most likely to have a certain disease D with probability 0.6. Further patient P is told that this score is confidence calibrated. P can now infer the following: "among all patients who have probability 0.6 of having *some unspecified* disease, the fraction who have *that unspecified* disease is also 0.6." However, P is concerned only about disease D, and not about other diseases. That is, P wants to know the probability of having D *among patients who were predicted to have disease D with confidence 0.6*, not among patients who were predicted to have *some* disease with confidence 0.6. In other words, P cares about the occurrence of D among patients who were told the same thing that P has been told. It is tempting to wish that the confidence calibrated probability 0.6 has any bearing on what P cares about. However, this faith is misguided, as the above reasoning suggests, and further illustrated through the following example.

Example 5.1. Suppose the instance space is $(X, Y) \in \{a, b\} \times \{1, 2, \ldots\}$. ($X$ can be seen as the random patient, and Y as the disease they are suffering from.) Consider a predictor (c, h) and let the values taken by (X, Y, c, h) be as follows:

Feature x	$P(X = x)$	Class prediction $c(x)$	Confidence $h(x)$	$P(Y = c(X) \mid X = x)$
a	0.5	1	0.6	0.2
b	0.5	2	0.6	1.0

The table specifies only the probabilities $P(Y = c(X) \mid X = x)$; the probabilities $P(Y = l \mid X = x)$, $l \neq c(x)$, can be set arbitrarily. We verify that (c, h) is confidence calibrated:

$$\mathbb{P}(Y = c(X) \mid h(X) = 0.6) = 0.5(\mathbb{P}(Y = 1 \mid X = a) + \mathbb{P}(Y = 2 \mid X = b)) = 0.5(0.2 + 1) = 0.6.$$

However, whether the actual instance is $X = a$ or $X = b$, *the probabilistic claim of* 0.6 *bears no correspondence with reality.* If $X = a$, $h(X) = 0.6$ is extremely overconfident since $P(Y = 1 \mid X = a) = 0.2$. Contrarily, if $X = b$, $h(X) = 0.6$ is extremely underconfident. □

The reason for the strange behavior above is that the probability $\mathbb{P}(Y = c(X) \mid h(X))$ is not interpretable from a decision-making perspective. In practice, we never report just the confidence $h(X)$, but also the class prediction $c(X)$ (obviously!). Thus it is more reasonable to talk about the conditional probability of $Y = c(X)$, given what is reported, that is *both* $c(X)$

and $h(X)$. We make a small but critical change to (5.1); we say that (c, h) is *top-label calibrated* if

$$\mathbb{P}(Y = c(X) \mid h(X), c(X)) = h(X). \tag{5.2}$$

(See the disambiguating Remark 5.2 on terminology.) Going back to the patient-disease example, top-label calibration would tell patient P the following: "among all patients, who *(just like you)* are predicted to have disease D with probability 0.6, the fraction who actually have disease D is also 0.6." Philosophically, it makes sense to condition on what is reported—both the top label and its confidence—because that is what is known to the recipient of the information; and there is no apparent justification for *not* conditioning on both.

A commonly used metric for quantifying the miscalibration of a model is the expected-calibration-error (ECE) metric. The ECE associated with confidence calibration is defined as

$$\text{conf-ECE}(c, h) := \mathbb{E}_X \left| \mathbb{P}(Y = c(X) \mid h(X)) - h(X) \right|. \tag{5.3}$$

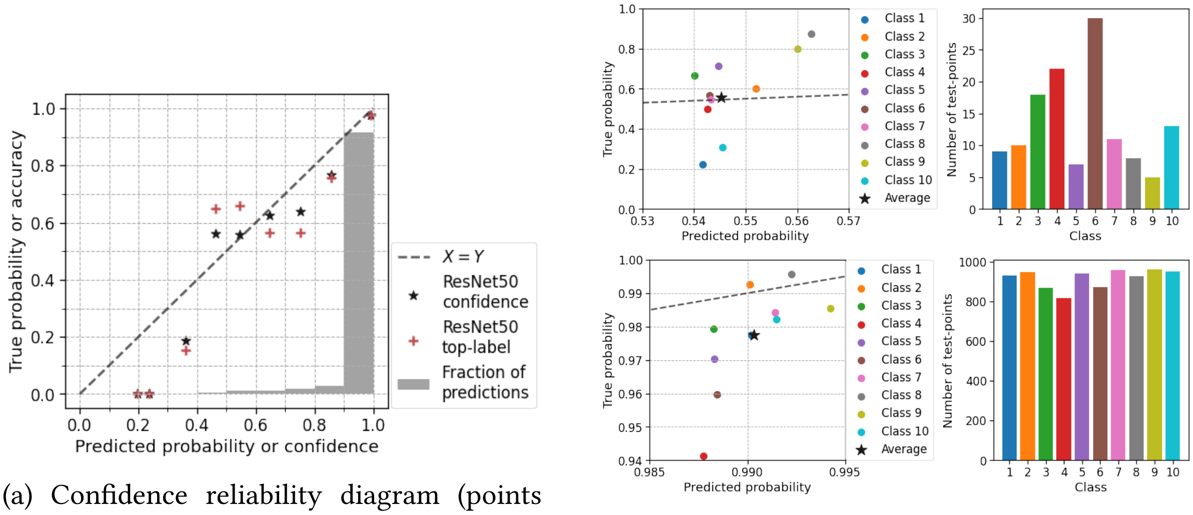
We define top-label-ECE (TL-ECE) in an analogous fashion, but also condition on $c(X)$:

$$\text{TL-ECE}(c, h) := \mathbb{E}_X \left| \mathbb{P}(Y = c(X) \mid c(X), h(X)) - h(X) \right|. \tag{5.4}$$

d Higher values of ECE indicate worse calibration performance. The predictor in Example 5.1 has conf-ECE$(c, h) = 0$. However, it has TL-ECE$(c, h) = 0.4$, revealing its miscalibration. More generally, it can be deduced as a straightforward consequence of Jensen's inequality that conf-ECE(c, h) is always smaller than the TL-ECE(c, h) (see Proposition 5.4 in Appendix 5.H). As illustrated by Example 5.1, the difference can be significant. In the following subsection we illustrate that the difference can be significant on a real dataset as well. First, we make a couple of remarks.

Remark 5.1 (ECE estimation using binning)**.** Estimating the ECE requires estimating probabilities conditional on some prediction such as $h(x)$. A common strategy to do this is to *bin* together nearby values of $h(x)$ using *binning* schemes (Nixon et al., 2020, Section 2.1), and compute a single estimate for the predicted and true probabilities using all the points in a bin. The calibration method we espouse in this work, histogram binning (HB), produces discrete predictions whose ECE can be estimated without further binning. Based on this, we use the following experimental protocol: we report unbinned ECE estimates while assessing HB, and binned ECE estimates for all other compared methods, which are continuous output methods (deep-nets, temperature scaling, etc). It is commonly understood that binning leads to underestimation of the effective ECE (Vaicenavicius et al., 2019; Kumar et al., 2019). Thus, using unbinned ECE estimates for HB gives HB a disadvantage compared to the binned ECE estimates we use for other methods. (This further strengthens our positive results for HB.) The binning scheme we use is equal-width binning, where the interval $[0, 1]$ is divided into B equal-width intervals. Equal-width binning typically leads to lower ECE estimates compared to adaptive-width binning (Nixon et al., 2020).

Remark 5.2 (Terminology)**.** The term conf-ECE was introduced by Kull et al. (2019). Most works refer to conf-ECE as just ECE (Guo et al., 2017; Nixon et al., 2020; Mukhoti et al., 2020; Kumar et al., 2018). However, some papers refer to conf-ECE as top-label-ECE (Kumar et al.,



(a) Confidence reliability diagram (points marked ★) and top-label reliability diagram (points marked +) for a ResNet-50 model on the CIFAR-10 dataset; see further details in points (a) and (b) below. The **gray bars** denote the fraction of predictions in each bin. The confidence reliability diagram (mistakenly) suggests better calibration than the top-label reliability diagram.

(b) Class-wise and zoomed-in version of Figure 5.1a for bin 6 (top) and bin 10 (bottom); see further details in point (c) below. The ★ markers are in the same position as Figure 5.1a, and denote the average predicted and true probabilities. The colored points denote the predicted and true probabilities when seen class-wise. The histograms on the right show the number of test points per class within bins 6 and 10.

Figure 5.1: Confidence reliability diagrams misrepresent the effective miscalibration.

2019; Zhang et al., 2020), resulting in two different terms for the same concept. We call the older notion as conf-ECE, and *our definition of top-label calibration/ECE* (5.4) *is different from previous ones.*

5.2.1 An illustrative experiment with ResNet-50 on CIFAR-10

We now compare confidence and top-label calibration using ECE estimates and reliability diagrams (Niculescu-Mizil and Caruana, 2005). This experiment can be seen as a less malignant version of Example 5.1. Here, confidence calibration is not completely meaningless, but can nevertheless be misleading. Figure 5.1 illustrates the (test-time) calibration performance of a ResNet-50 model (He et al., 2016) on the CIFAR-10 dataset (Krizhevsky, 2009). In the following summarizing points, the (c, h) correspond to the ResNet-50 model.

(a) The ★ markers in Figure 5.1a form the **confidence reliability diagram** (Guo et al., 2017), constructed as follows. First, the $h(x)$ values on the test set are binned into one of $B = 10$ bins, $[0, 0.1), [0.1, 0.2), \ldots, [0.9, 1]$, depending on the interval to which $h(x)$ belongs. The gray bars in Figure 5.1a indicate the fraction of $h(x)$ values in each bin—nearly 92% points belong to bin $[0.9, 1]$ and no points belong to bin $[0, 0.1)$. Next, for every bin b, we plot ★ $= (\text{conf}_b, \text{acc}_b)$, which are the plugin estimates of $\mathbb{E}\left[h(X) \mid h(X) \in \text{Bin } b\right]$ and $\mathbb{P}(Y = c(X) \mid h(X) \in \text{Bin } b)$ respectively. The dashed $X = Y$ line indicates perfect confidence calibration.

(b) The + markers in Figure 5.1a form the **top-label reliability diagram**. Unlike the confidence reliability diagram, the top-label reliability diagram shows the average *miscalibration* across classes in a given bin. For a given class l and bin b, define

$$\Delta_{b,l} := |\widehat{\mathbb{P}}(Y = c(X) \mid c(X) = l, h(X) \in \text{Bin } b) - \widehat{\mathbb{E}}\left[h(X) \mid c(X) = l, h(X) \in \text{Bin } b\right]|,$$

where $\widehat{\mathbb{P}}, \widehat{\mathbb{E}}$ denote empirical estimates based on the test data. The overall miscalibration is then

$$\Delta_b := \text{Weighted-average}(\Delta_{b,l}) = \textstyle\sum_{l \in [L]} \widehat{\mathbb{P}}(c(X) = l \mid h(X) \in \text{Bin } b)\ \Delta_{b,l}.$$

Note that Δ_b is always non-negative and does not indicate whether the overall miscalibration occurs due to under- or over-confidence; also, if the absolute-values were dropped from $\Delta_{b,l}$, then Δ_b would simply equal $\text{acc}_b - \text{conf}_b$. In order to plot Δ_b in a reliability diagram, we obtain the direction for the corresponding point from the confidence reliability diagram. Thus for every $\bigstar = (\text{conf}_b, \text{acc}_b)$, we plot $+ = (\text{conf}_b, \text{conf}_b + \Delta_b)$ if $\text{acc}_b > \text{conf}_b$ and $+ = (\text{conf}_b, \text{conf}_b - \Delta_b)$ otherwise, for every b. This scatter plot of the +'s gives us the top-label reliability diagram.

Figure 5.1a shows that there is a **visible increase in miscalibration when going from confidence calibration to top-label calibration**. To understand why this change occurs, Figure 5.1b zooms into the sixth bin ($h(X) \in [0.5, 0.6)$) and bin 10 ($h(X) \in [0.9, 1.0]$), as described next.

(c) Figure 5.1b displays the **class-wise top-label reliability diagrams** for bins 6 and 10. Note that for bin 6, the ★ marker is nearly on the $X = Y$ line, indicating that the overall accuracy matches the overall confidence of 0.545. However, the true accuracy when class 1 was predicted is ≈ 0.2 and the true accuracy when class 8 was predicted is ≈ 0.9 (a very similar scenario to Example 5.1). For bin 10, the ★ marker indicates a miscalibration of ≈ 0.01; however, when class 4 was predicted (roughly 8% of all test-points) the miscalibration is ≈ 0.05.

Figure 5.2 displays the aggregate effect of the above phenomenon (across bins and classes) through estimates of the conf-ECE and TL-ECE. The precise experimental setup is described in Section 5.4. These plots display the ECE estimates of the base model, as well as the base model when recalibrated using temperature scaling (Guo et al., 2017) and our upcoming formulation of top-label histogram binning (Section 5.3). Since ECE estimates depend on the number of bins B used (see Roelofs et al. (2022) for empirical work around this), we plot the ECE estimate for every value $B \in [5, 25]$ in order to obtain clear and unambiguous results. We find that the TL-ECE is significantly higher than the conf-ECE for most values of B, the architectures, and the pre- and post- recalibration models. This figure also previews the performance of our forthcoming top-label histogram binning algorithm. Top-label HB has smaller estimated TL-ECE than temperature scaling for most values of B and the architectures. Except for ResNet-50, the conf-ECE estimates are also better.

To summarize, top-label calibration captures the intuition of confidence calibration by focusing on the predicted class. However, top-label calibration also conditions on the predicted class,

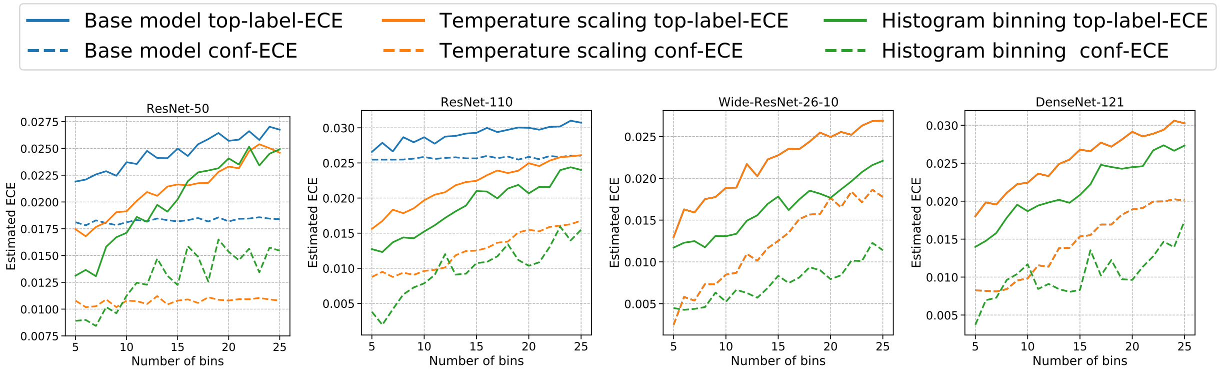


Figure 5.2: Conf-ECE (dashed lines) and TL-ECE (solid lines) of four deep-net architectures on CIFAR-10, as well as with recalibration using histogram binning and temperature scaling. The TL-ECE is often 2-3 times the conf-ECE, depending on the number of bins used to estimate ECE, and the architecture. Top-label histogram binning typically performs better than temperature scaling.

Calibration notion	Quantifier	Prediction (pred(X))	Binary calibration statement
Confidence	-	$h(X)$	$P(Y = c(X) \mid \text{pred}(X)) = h(X)$
Top-label	-	$c(X), h(X)$	$P(Y = c(X) \mid \text{pred}(X)) = h(X)$
Class-wise	$\forall l \in [L]$	$h_l(X)$	$P(Y = l \mid \text{pred}(X)) = h_l(X)$
Top-K-confidence	$\forall k \in [K]$	$h^{(k)}(X)$	$P(Y = c^{(k)}(X) \mid \text{pred}(X)) = h^{(k)}(X)$
Top-K-label	$\forall k \in [K]$	$c^{(k)}(X), h^{(k)}(X)$	$P(Y = c^{(k)}(X) \mid \text{pred}(X)) = h^{(k)}(X)$

Table 5.1: Multiclass-to-binary (M2B) notions internally verify one or more binary calibration statements/claims. The statements in the rightmost column are required to hold almost surely.

which is always part of the prediction in any practical setting. Further, TL-ECE estimates can be substantially different from conf-ECE estimates. Thus, while it is common to compare predictors based on the conf-ECE, the TL-ECE comparison is more meaningful, and can potentially be different.

5.3 Calibration algorithms from calibration metrics

In this section, we unify a number of notions of multiclass calibration as multiclass-to-binary (or M2B) notions, and propose a general-purpose calibration algorithm that achieves the corresponding M2B notion of calibration. The M2B framework yields multiple novel post-hoc calibration algorithms, each of which is tuned to a specific M2B notion of calibration.

5.3.1 Multiclass-to-binary (M2B) notions of calibration

In Section 5.2, we defined confidence calibration (5.1) and top-label calibration (5.2). These notions verify calibration claims for the highest predicted probability. Other popular notions of calibration verify calibration claims for other entries in the full L-dimensional prediction vector. A predictor $\mathbf{h} = (h_1, h_2, \ldots, h_L)$ is said to be class-wise calibrated (Kull et al., 2017) if

$$\text{(class-wise calibration)} \quad \forall l \in [L], \mathbb{P}(Y = l \mid h_l(X)) = h_l(X). \tag{5.5}$$

Another recently proposed notion is top-K confidence calibration (Gupta et al., 2021). For some $l \in [L]$, let $c^{(l)} : \mathcal{X} \to [L]$ denote the l-th highest class prediction, and let $h^{(l)} : \mathcal{X} \to [L]$ denote the confidence associated with it ($c = c^{(1)}$ and $h = h^{(1)}$ are special cases). For a given $K \leqslant L$,

$$\text{(top-}K\text{-confidence calibration)} \quad \forall k \in [K], \mathbb{P}(Y = c^{(k)}(X) \mid h^{(k)}(X)) = h^{(k)}(X). \tag{5.6}$$

As we did in Section 5.2 for confidence→top-label, top-K-confidence calibration can be modified to the more interpretable top-K-label calibration by further conditioning on the predicted labels:

$$\text{(top-}K\text{-label calibration)} \quad \forall k \in [K], \mathbb{P}(Y = c^{(k)}(X) \mid h^{(k)}(X), c^{(k)}(X)) = h^{(k)}(X). \tag{5.7}$$

Each of these notions reduce multiclass calibration to one or more binary calibration requirements, where each binary calibration requirement corresponds to **verifying if the distribution of Y, conditioned on some prediction pred(X), satisfies a single binary calibration claim associated with pred(X)**. Table 5.1 illustrates how the calibration notions discussed so far internally verify a number of binary calibration claims, making them M2B notions. For example, for class-wise calibration, for every $l \in [L]$, the conditioning is on pred$(X) = h_l(X)$, and a single binary calibration statement is verified: $P(Y = l \mid \text{pred}(X)) = h_l(X)$. Based on this property, we call each of these notions multiclass-to-binary or M2B notions.

The notion of canonical calibration mentioned in the introduction is *not* an M2B notion. Canonical calibration is discussed in detail in Appendix 5.G. Due to the conditioning on a multi-dimensional prediction, non-M2B notions of calibration are harder to achieve or verify. For the same reason, it is possibly easier for humans to interpret binary calibration claims when taking decisions/actions.

5.3.2 Achieving M2B notions of calibration using M2B calibrators

The M2B framework illustrates how multiclass calibration can typically be viewed via a reduction to binary calibration. The immediate consequence of this reduction is that one can now solve multiclass calibration problems by leveraging the well-developed methodology for binary calibration.

The upcoming M2B calibrators belong to the standard recalibration or post-hoc calibration setting. In this setting, one starts with a fixed pre-learnt base model $\mathbf{g} : \mathcal{X} \to \Delta^{L-1}$. The base model $\mathbf{g}$ can correspond to a deep-net, a random forest, or any 1-v-all (one-versus-all) binary classification model such as logistic regression. The base model is typically optimized

M2B calibrators. Input in each case: Binary calibrator $\mathcal{A}_{\{0,1\}} : [0,1]^{\mathcal{X}} \times (\mathcal{X} \times \{0,1\})^\star \to [0,1]^{\mathcal{X}}$, base multiclass predictor $\mathbf{g} : \mathcal{X} \to \Delta^{L-1}$, calibration data $\mathcal{D} = (X_1, Y_1), \ldots, (X_n, Y_n)$.

Algorithm 5.1 Confidence calibrator

```
$c \leftarrow$ classifier or top-class based on $\mathbf{g}$
$g \leftarrow$ top-class-probability based on $\mathbf{g}$
$\mathcal{D}' \leftarrow \{(X_i, \mathbb{1}\{Y_i = c(X_i)\}) : i \in [n]\}$
$h \leftarrow \mathcal{A}_{\{0,1\}}(g, \mathcal{D}')$
return $(c, h)$
```

Algorithm 5.2 Top-label calibrator

```
$c \leftarrow$ classifier or top-class based on $\mathbf{g}$
$g \leftarrow$ top-class-probability based on $\mathbf{g}$
for $l \leftarrow 1$ to $L$ do
    $\mathcal{D}_l \leftarrow \{(X_i, \mathbb{1}\{Y_i = l\}) : c(X_i) = l)\}$
    $h_l \leftarrow \mathcal{A}_{\{0,1\}}(g, \mathcal{D}_l)$
end for
$h(\cdot) \leftarrow h_{c(\cdot)}(\cdot)$ (predict $h_l(x)$ if $c(x) = l$)
return $(c, h)$
```

Algorithm 5.3 Class-wise calibrator

```
Write $\mathbf{g} = (g_1, g_2, \ldots, g_L)$
for $l \leftarrow 1$ to $L$ do
    $\mathcal{D}_l \leftarrow \{(X_i, \mathbb{1}\{Y_i = l\}) : i \in [n]\}$
    $h_l \leftarrow \mathcal{A}_{\{0,1\}}(g_l, \mathcal{D}_l)$
end for
return $(h_1, h_2, \ldots, h_L)$
```

Algorithm 5.4 Normalized calibrator

```
Write $\mathbf{g} = (g_1, g_2, \ldots, g_L)$
for $l \leftarrow 1$ to $L$ do
    $\mathcal{D}_l \leftarrow \{(X_i, \mathbb{1}\{Y_i = l\}) : i \in [n]\}$
    $\widetilde{h}_l \leftarrow \mathcal{A}_{\{0,1\}}(g_l, \mathcal{D}_l)$
end for
Normalize: for every $l \in [L]$, $h_l(\cdot) := \widetilde{h}_l(\cdot) / \sum_{k=1}^{L} \widetilde{h}_k(\cdot)$
return $(h_1, h_2, \ldots, h_L)$
```

for classification accuracy and may not be calibrated. The goal of post-hoc calibration is to use some given *calibration data* $\mathcal{D} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n) \in (\mathcal{X} \times [L])^n$, typically data on which $\mathbf{g}$ was not learnt, to recalibrate $\mathbf{g}$. In practice, the calibration data is usually the same as the validation data.

To motivate M2B calibrators, suppose we want to verify if $\mathbf{g}$ is calibrated on a certain test set, based on a given M2B notion of calibration. Then, the verifying process will split the test data into a number of sub-datasets, each of which will verify one of the binary calibration claims. In Appendix 5.A.2, we argue that the calibration data can also be viewed as a test set, and every step in the verification process can be used to provide a signal for improving calibration.

M2B calibrators take the form of *wrapper* methods that work on top of a given binary calibrator. Denote an arbitrary black-box binary calibrator as $\mathcal{A}_{\{0,1\}} : [0,1]^{\mathcal{X}} \times (\mathcal{X} \times \{0,1\})^\star \to [0,1]^{\mathcal{X}}$, where the first argument is a mapping $\mathcal{X} \to [0,1]$ that denotes a (miscalibrated) binary predicor, and the second argument is a calibration data sequence of arbitrary length. The output is a (better calibrated) binary predictor. Examples of $\mathcal{A}_{\{0,1\}}$ are histogram binning (Zadrozny and Elkan, 2001), isotonic regression (Zadrozny and Elkan, 2002), and Platt scaling (Platt, 1999). In the upcoming descriptions, we use the indicator function $\mathbb{1}\{a = b\} \in \{0,1\}$ which takes the value 1 if $a = b$, and 0 if $a \neq b$.

The general formulation of our M2B calibrator is delayed to Appendix 5.A since the description is a bit involved. To ease readability, in the main part of the chapter we describe the calibrators corresponding to top-label, class-wise, and confidence calibration (Algorithms 5.1–5.3). Each of these calibrators are different from the *classical* M2B calibrator (Algorithm 5.4) that has been used by Zadrozny and Elkan (2002), Guo et al. (2017), Kull et al. (2019), and most other papers we are aware of, with the most similar one being Algorithm 5.3. Top-K-label and top-K-confidence calibrators are also explicitly described in Appendix 5.A (Algorithms 5.6 and 5.7).

Top-label calibration requires that for every class $l \in [L]$, $\mathbb{P}(Y = l \mid c(X) = l, h(X)) = h(X)$. Thus, to achieve top-label calibration, we must solve L calibration problems. Algorithm 5.2 constructs L datasets $\{\mathcal{D}_l : l \in [L]\}$ (line 4). The features in $\mathcal{D}_l$ are the X_i's for which $c(X_i) = l$, and the labels are $\mathbb{1}\{Y_i = l\}$. Now for every $l \in [L]$, we calibrate g to $h_l : \mathcal{X} \to [0, 1]$ using $\mathcal{D}_l$ and any binary calibrator. The final probabilistic predictor is $h(\cdot) = h_{c(\cdot)}(\cdot)$ (that is, it predicts $h_l(x)$ if $c(x) = l$). The top-label predictor c does not change in this process. Thus the accuracy of (c, h) is the same as the accuracy of $\mathbf{g}$ irrespective of which $\mathcal{A}_{\{0,1\}}$ is used. Unlike the top-label calibrator, the confidence calibrator merges all classes together into a single dataset $\mathcal{D}' = \bigcup_{l\in[L]} \mathcal{D}_l$.

To achieve class-wise calibration, Algorithm 5.3 also solves L calibration problems, but these correspond to satisfying $\mathbb{P}(Y = l \mid h_l(X)) = h_l(X)$. Unlike top-label calibration, the dataset D_l for class-wise calibration contains all the X_i's (even if $c(X_i) \neq l$), and h_l is passed to $\mathcal{A}_{\{0,1\}}$ instead of h. Also, unlike confidence calibration, Y_i is replaced with $\mathbb{1}\{Y_i = l\}$ instead of $\mathbb{1}\{Y_i = c(X_i)\}$. The overall process is similar to reducing multiclass classification to L 1-v-all binary classification problem, but our motivation is intricately tied to the notion of class-wise calibration.

Most popular empirical works that have discussed binary calibrators for multiclass calibration have done so using the normalized calibrator, Algorithm 5.4. This is almost identical to Algorithm 5.3, except that there is an additional normalization step (line 6 of Algorithm 5.4). This normalization was first proposed by Zadrozny and Elkan (2002, Section 5.2), and has been used unaltered by most other works[1] where the goal has been to simply compare direct multiclass calibrators such as temperature scaling, Dirichlet scaling, etc., to a calibrator based on binary methods (for instance, see Section 4.2 of Guo et al. (2017)). In contrast to these papers, we investigate multiple M2B reductions in an effort to identify the right reduction of multiclass calibration to binary calibration.

To summarize, the M2B characterization immediately yields a novel and different calibrator for every M2B notion. In the following section, we instantiate M2B calibrators on the binary calibrator of histogram binning (HB), leading to two new algorithms: top-label-HB and class-wise-HB, that achieve strong empirical results and satisfy distribution-free calibration guarantees.

Metric	Dataset	Architecture	Base	TS	VS	DS	N-HB	TL-HB
Top-label-ECE	CIFAR-10	ResNet-50	0.025	0.022	0.020	0.019	**0.018**	0.020
		ResNet-110	0.029	0.022	0.021	0.021	**0.020**	0.021
		WRN-26-10	0.023	0.023	0.019	0.021	**0.012**	0.018
		DenseNet-121	0.027	0.027	0.020	0.020	**0.019**	0.021
	CIFAR-100	ResNet-50	0.118	0.114	0.113	0.322	**0.081**	0.143
		ResNet-110	0.127	0.121	0.115	0.353	**0.093**	0.145
		WRN-26-10	0.103	0.103	0.100	0.304	**0.070**	0.129
		DenseNet-121	0.110	0.110	0.109	0.322	**0.086**	0.139
Top-label-MCE	CIFAR-10	ResNet-50	0.315	0.305	0.773	0.282	0.411	**0.107**
		ResNet-110	0.275	0.227	0.264	0.392	0.195	**0.077**
		WRN-26-10	0.771	0.771	0.498	0.325	0.140	**0.071**
		DenseNet-121	0.289	0.289	0.734	0.294	0.345	**0.087**
	CIFAR-100	ResNet-50	0.436	0.300	**0.251**	0.619	0.397	0.291
		ResNet-110	0.313	**0.255**	0.277	0.557	0.266	0.257
		WRN-26-10	0.273	**0.255**	0.256	0.625	0.287	0.280
		DenseNet-121	0.279	**0.231**	0.235	0.600	0.320	0.289

Table 5.2: Top-label-ECE and top-label-MCE for deep-net models (above: 'Base') and various post-hoc calibrators: temperature-scaling (TS), vector-scaling (VS), Dirichlet-scaling (DS), top-label-HB (TL-HB), and normalized-HB (N-HB). Best performing method in each row is in **bold**.

Metric	Dataset	Architecture	Base	TS	VS	DS	N-HB	CW-HB
Class-wise-ECE $\times 10^2$	CIFAR-10	ResNet-50	0.46	0.42	0.35	0.35	0.50	**0.28**
		ResNet-110	0.59	0.50	0.42	0.38	0.53	**0.27**
		WRN-26-10	0.44	0.44	0.35	0.39	0.39	**0.28**
		DenseNet-121	0.46	0.46	**0.36**	**0.36**	0.48	**0.36**
	CIFAR-100	ResNet-50	0.22	0.20	0.20	0.66	0.23	**0.16**
		ResNet-110	0.24	0.23	0.21	0.72	0.24	**0.16**
		WRN-26-10	0.19	0.19	0.18	0.61	0.20	**0.14**
		DenseNet-121	0.20	0.21	0.19	0.66	0.24	**0.16**

Table 5.3: Class-wise-ECE for deep-net models and various post-hoc calibrators. All methods are same as Table 5.2, except TL-HB is replaced with class-wise-HB (CW-HB).

5.4 Experiments: M2B calibration with histogram binning

Histogram binning or HB was proposed by Zadrozny and Elkan (2001) with strong empirical results for binary calibration. In HB, a base binary calibration model $g : \mathcal{X} \to [0, 1]$ is used to partition the calibration data into a number of bins so that each bin has roughly the same

[1] the only exception we are aware of is the recent work of Patel et al. (2021) who also suggest skipping normalization (see their Appendix A1); however they use a common I-Max binning scheme across classes, whereas in Algorithm 5.3 the predictor h_l for each class is learnt completely independently of other classes

number of points. Then, for each bin, the probability of $Y = 1$ is estimated using the empirical distribution on the calibration data. This estimate forms the new calibrated prediction for that bin. Recently, Gupta and Ramdas (2021) showed that HB satisfies strong distribution-free calibration guarantees, which are otherwise impossible for scaling methods (Gupta et al., 2020).

Despite these results for binary calibration, studies for multiclass calibration have reported that HB typically performs worse than scaling methods such as temperature scaling (TS), vector scaling (VS), and Dirichlet scaling (DS) (Kull et al., 2019; Roelofs et al., 2022; Guo et al., 2017). In our experiments, we find that the issue is not HB but the M2B wrapper used to produce the HB baseline. With the right M2B wrapper, HB beats TS, VS, and DS. A number of calibrators have been proposed recently (Zhang et al., 2020; Rahimi et al., 2020; Patel et al., 2021; Gupta et al., 2021), but VS and DS continue to remain strong baselines which are often close to the best in these papers. We do not compare to each of these calibrators; our focus is on the M2B reduction and the message that the baselines dramatically improve with the right M2B wrapper.

We use three metrics for comparison: the first is top-label-ECE or TL-ECE (defined in (5.4)), which we argued leads to a more meaningful comparison compared to conf-ECE. Second, we consider the more stringent maximum-calibration-error (MCE) metric that assesses the worst calibration across predictions (see more details in Appendix 5.E.3). For top-label calibration MCE is given by

$$\text{TL-MCE}(c, h) := \max_{l \in [L]} \sup_{r \in \text{Range}(h)} \left| \mathbb{P}(Y = l \mid c(X) = l, h(X) = r) - r \right|.$$

To assess class-wise calibration, we use class-wise-ECE defined as the average calibration error across classes:

$$\text{CW-ECE}(c, \mathbf{h}) := L^{-1} \textstyle\sum_{l=1}^{L} \mathbb{E}_X \left| \mathbb{P}(Y = l \mid h_l(X)) - h_l(X) \right|.$$

All ECE/MCE estimation is performed as described in Remark 5.1. For further details, see Appendix 5.E.2.

Formal algorithm and theoretical guarantees. Top-label-HB (TL-HB) and class-wise-HB (CW-HB) are explicitly stated in Appendices 5.B and 5.C respectively; these are instantiations of the top-label calibrator and class-wise calibrator with HB. N-HB is the the normalized calibrator (Algorithm 5.4) with HB, which is the same as CW-HB, but with an added normalization step. In the Appendix, we extend the binary calibration guarantees of Gupta and Ramdas (2021) to TL-HB and CW-HB (Theorems 5.1 and 5.2). We informally summarize one of the results here: if there are at least k calibration points-per-bin, then the expected-ECE is bounded as: $\mathbb{E}$ [(TL-) or (CW-) ECE] $\leqslant \sqrt{1/2k}$, for TL-HB and CW-HB respectively. The outer $\mathbb{E}$ above is an expectation over the calibration data, and corresponds to the randomness in the predictor learnt on the calibration data. Note that the ECE itself is an expected error over an unseen i.i.d. test-point $(X, Y) \sim P$.

Experimental details. We experimented on the CIFAR-10 and CIFAR-100 datasets, which have 10 and 100 classes each. The base models are deep-nets with the following architectures: ResNet-50, Resnet-110, Wide-ResNet-26-10 (WRN) (Zagoruyko and Komodakis, 2016), and DenseNet-121 (Huang et al., 2017). Both CIFAR datasets consist of 60K (60,000) points, which are split as 45K/5K/10K to form the train/validation/test sets. The validation set was used for post-hoc calibration and the test set was used for evaluation through ECE/MCE estimates. Instead of training new models, we used the pre-trained models of Mukhoti et al. (2020). We then ask: *"which post-hoc calibrator improves the calibration the most?"* We used their Brier score and focal loss models in our experiments (Mukhoti et al. (2020) report that these are the empirically best performing loss functions). *All results in the main chaoter are with Brier score, and results with focal loss are in Appendix 5.E.4.* Implementation details for TS, VS, and DS are in Appendix 5.E.

Findings. In Table 5.2, we report the binned ECE and MCE estimates when $B = 15$ bins are used by HB, and for ECE estimation. We make the following observations:

(a) For TL-ECE, N-HB is the best performing method for both CIFAR-10 and CIFAR-100. While most methods perform similarly across architectures for CIFAR-10, there is high variation in CIFAR-100. DS is the worst performing method on CIFAR-100, but TL-HB also performs poorly. We believe that this could be because the data splitting scheme of the TL-calibrator (line 4 of Algorithm 5.2) splits datasets across the predicted classes, and some classes in CIFAR-100 occur very rarely. This is further discussed in Appendix 5.E.6.

(b) For TL-MCE, TL-HB is the best performing method on CIFAR-10, by a huge margin. For CIFAR-100, TS or VS perform slightly better than TL-HB. Since HB ensures that each bin gets roughly the same number of points, the predictions are well calibrated across bins, leading to smaller TL-MCE. A similar observation was also made by Gupta and Ramdas (2021).

(c) For CW-ECE, CW-HB is the best performing method across the two datasets and all four architectures. The N-HB method which has been used in many CW-ECE baseline experiments performs terribly. In other words, skipping the normalization step leads to a large improvement in CW-ECE. **This observation is one of our most striking findings.** To shed further light on this, we note that the distribution-free calibration guarantees for CW-HB shown in Appendix 5.C no longer hold post-normalization. Thus, both our theory and experiments indicate that skipping normalization improves CW-ECE performance.

Additional experiments in the Appendix. In Appendix 5.E.5, we report each of the results in Tables 5.2 and 5.3 with the number of bins taking every value in the range $[5, 25]$. Most observations remain the same under this expanded study. In Appendix 5.B.2, we consider top-label calibration for the class imbalanced COVTYPE-7 dataset, and show that TL-HB adapts to tail/infrequent classes.

5.5 Conclusion

We make two contributions to the study of multiclass calibration: (i) defining the new notion of top-label calibration which enforces a natural minimal requirement on a multiclass predictor—the probability score for the top-label should be calibrated conditioned on the reported top-label; (ii) developing a multiclass-to-binary (M2B) framework which posits that various notions of multiclass calibration can be achieved via reduction to binary calibration, balancing practical utility with statistically tractability. Since it is important to identify appropriate notions of calibration in any structured output space (Kuleshov et al., 2018; Gneiting et al., 2007), we anticipate that the philosophy behind the M2B framework could find applications in other structured spaces.

5.6 Reproducibility

Some reproducibility desiderata, such as external code and libraries that were used are summarized in Appendix 5.E.1. Most of our experiments can be reproduced using the code at https://github.com/aigen/df-posthoc-calibration. Our base models were pre-trained deep-net models generated by Mukhoti et al. (2020), obtained from www.robots.ox.ac.uk/~viveka/focal_calibration/ (corresponding to 'brier_score' and 'focal_loss_adaptive_53' at the above link). By avoiding training of new deep-net models with multiple hyperparameters, we also consequently avoided selection biases that inevitably creep in due to test-data-peeking. The predictions of the pre-trained models were obtained using the code at https://github.com/torrvision/focal_calibration.

Appendices for Chapter 5

5.A Addendum to Section 5.3 “Calibration algorithms from calibration metrics”

In Section 5.3, we introduced the concept of M2B calibration, and showed that popular calibration notions are in fact M2B notions (Table 5.1). We showed how the calibration notions of top-label, class-wise, and confidence calibration can be achieved using a corresponding M2B calibrator. In the following subsection, we present the general-purpose *wrapper* Algorithm 5.5 that can be used to derive an M2B calibrator from any given M2B calibration notion that follows the rubric specified by Table 5.1. In Appendix 5.A.2, we illustrate the philosophy of M2B calibration using a simple example with a dataset that contains 6 points. This example also illustrates the top-label-calibrator, the class-wise-calibrator, and the confidence-calibrator.

5.A.1 General-purpose M2B calibrator

Denote some M2B notion of calibration as $\mathcal{C}$. Suppose $\mathcal{C}$ corresponds to K binary calibration claims. The outer for-loop in Algorithm 5.5, runs over each such claim in $\mathcal{C}$. For example, for class-wise calibration, $K = L$ and for confidence and top-label calibration, $K = 1$. Corresponding to each claim, there is a probability-predictor that the conditioning is to be done on, such as g or g_l or $g_{(k)}$. Additionally, there may be conditioning on the label predictor such as c or $c_{(k)}$. These are denoted as $(\widetilde{c}, \widetilde{g})$ in Algorithm 5.5. For confidence and top-label calibration, $\widetilde{c} = c$, the top-label-confidence. For class-wise calibration, when $\widetilde{g} = g_l$, we have $\widetilde{c}(\cdot) = l$.

If there is no label conditioning in the calibration notion, such as in confidence, top-K-confidence, and class-wise calibration, then we enter the if-condition inside the for-loop. Here h_k is learnt using a single calibration dataset and a single call to $\mathcal{A}_{\{0,1\}}$. Otherwise, if there is label conditioning, such as in top-label and top-K-label calibration, we enter the else-condition, where we learn a separate $h_{k,l}$ for every $l \in [L]$, using a different part of the dataset $\mathcal{D}_l$ in each case. Then $h_k(x)$ equals $h_{k,l}(x)$ if $\widetilde{c}(x) = l$.

Finally, since $\mathcal{C}$ is verifying a sequence of claims, the output of Algorithm 5.5 is a sequence of predictors. Each original prediction $(\widetilde{c}, \widetilde{g})$ corresponding to the $\mathcal{C}$ is replaced with $(\widetilde{c}, h_k)$. This is the output of the M2B calibrator. Note that the $\widetilde{c}$ values are not changed. This output appears abstract, but normally, it can be represented in an interpretable way. For example, for class-wise calibration, the output is just a sequence of predictors, one for each class: $(h_1, h_2, \ldots, h_L)$.

Algorithm 5.5 Post-hoc calibrator for a given M2B calibration notion $\mathcal{C}$

```
Require: Base (uncalibrated) multiclass predictor $\mathbf{g}$, calibration data $\mathcal{D} = (X_1, Y_1), \ldots, (X_n, Y_n)$, binary calibrator $\mathcal{A}_{\{0,1\}} : [0,1]^{\mathcal{X}} \times (\mathcal{X} \times \{0,1\})^\star \to [0,1]^{\mathcal{X}}$
$K \leftarrow$ number of distinct calibration claims that $\mathcal{C}$ verifies
for each claim $k \in [K]$ do
    From $\mathbf{g}$, infer $(\widetilde{c}, \widetilde{g}) \leftarrow$ (label-predictor, probability-predictor) corresponding to claim $k$
    $\mathcal{D}_k \leftarrow \{(X_i, Z_i)\}$, where $Z_i \leftarrow \mathbb{1}\{Y_i = \widetilde{c}(X_i)\}$
    if conditioning does not include class prediction $\widetilde{c}$ then
        — (confidence, top-$K$-confidence, and class-wise calibration) —
        $h_k \leftarrow \mathcal{A}_{\{0,1\}}(\widetilde{g}, \mathcal{D}_k)$
    else
        — (top-label and top-$K$-label calibration) —
        for $l \in [L]$ do
            $\mathcal{D}_{k,l} \leftarrow \{(X_i, Z_i) \in \mathcal{D}_k : \widetilde{c}(X_i) = l)\}$
            $h_{k,l} \leftarrow \mathcal{A}_{\{0,1\}}(\widetilde{g}, \mathcal{D}_{k,l})$
        end for
        $h_k(\cdot) \leftarrow h_{k,\widetilde{c}(\cdot)}(\cdot)$  ($h_k$ predicts $h_{k,l}(x)$ if $\widetilde{c}(x) = l$)
    end if
end for
— (the new predictor replaces each $\widetilde{g}$ with the corresponding $h_k$) —
return (label-predictor, $h_k$) corresponding to each claim $k \in [K]$
```

This general-purpose M2B calibrators can be used to achieve any M2B calibration notion: top-label calibration (Algorithm 5.2), class-wise calibration (Algorithm 5.3), confidence calibration (Algorithm 5.1), top-K-label calibration (Algorithm 5.6), and top-K-confidence calibration (Algorithm 5.7).

5.A.2 An example to illustrate the philosophy of M2B calibration

Figure 5.3a shows the predictions of a given base model $\mathbf{g}$ on a given dataset $\mathcal{D}$. Suppose $\mathcal{D}$ is a *test set*, and we are testing confidence calibration. Then the only predictions that matter are the top-predictions corresponding to the shaded values. These are stripped out and shown in Figure 5.3b, in the $g(\cdot)$ row. Note that the indicator $\mathbb{1}\{Y = c(\cdot)\}$ is sufficient to test confidence calibration and given this, the $c(X)$ are not needed. Thus the second row in Figure 5.3b only shows these indicators. Verifying top-label calibration is similar (Figure 5.3c), but in addition to the predictions $g(\cdot)$, we also retain the values of $c(\cdot)$. Thus the $g(\cdot)$ and $\mathbb{1}\{Y = c(\cdot)\}$ are shown, but split across the 4 classes. Class-wise calibration requires access to all the predictions, however, each class is considered separately as indicated by Figure 5.3d. Canonical calibration looks at the full prediction vector in each case. However, in doing so, it becomes unlikely that $\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{y})$ for any $\mathbf{x}, \mathbf{y}$ since the number of values that $\mathbf{g}$ can take is now exponential.

Let us turn this around and suppose that $\mathcal{D}$ were a calibration set instead of a test set. We

Algorithm 5.6 Top-K-label calibrator

```
Require: Base multiclass predictor $\mathbf{g}$, calibration data $\mathcal{D} = (X_1, Y_1), \ldots, (X_n, Y_n)$
For every $k \in [K]$, infer from $\mathbf{g}$ the $k$-th largest class predictor $c^{(k)}$ and the associated probability $g^{(k)}$
for $k \leftarrow 1$ to $K$ do
    for $l \leftarrow 1$ to $L$ do
        $\mathcal{D}_{k,l} \leftarrow \{(X_i, \mathbb{1}\{Y_i = l\}) : c^{(k)}(X_i) = l\}$
        $h^{(k,l)} \leftarrow \mathcal{A}_{\{0,1\}}(g^{(k)}, \mathcal{D}_{k,l})$
    end for
    $h^{(k)} \leftarrow h^{(k,c^{(k)}(\cdot))}(\cdot)$
end for
return $(h^{(1)}, h^{(2)}, \ldots, h^{(K)})$
```

Algorithm 5.7 Top-K-confidence calibrator

```
Require: Base multiclass predictor $\mathbf{g}$, calibration data $\mathcal{D} = (X_1, Y_1), \ldots, (X_n, Y_n)$
For every $k \in [K]$, infer from $\mathbf{g}$ the $k$-th largest class predictor $c^{(k)}$ and the associated probability $g^{(k)}$
for $k \leftarrow 1$ to $K$ do
    $\mathcal{D}_k \leftarrow \{(X_i, \mathbb{1}\{Y_i = l\}) : i \in [n]\}$
    $h^{(k)} \leftarrow \mathcal{A}_{\{0,1\}}(g^{(k)}, \mathcal{D}_k)$
end for
return $(h^{(1)}, h^{(2)}, \ldots, h^{(K)})$
```

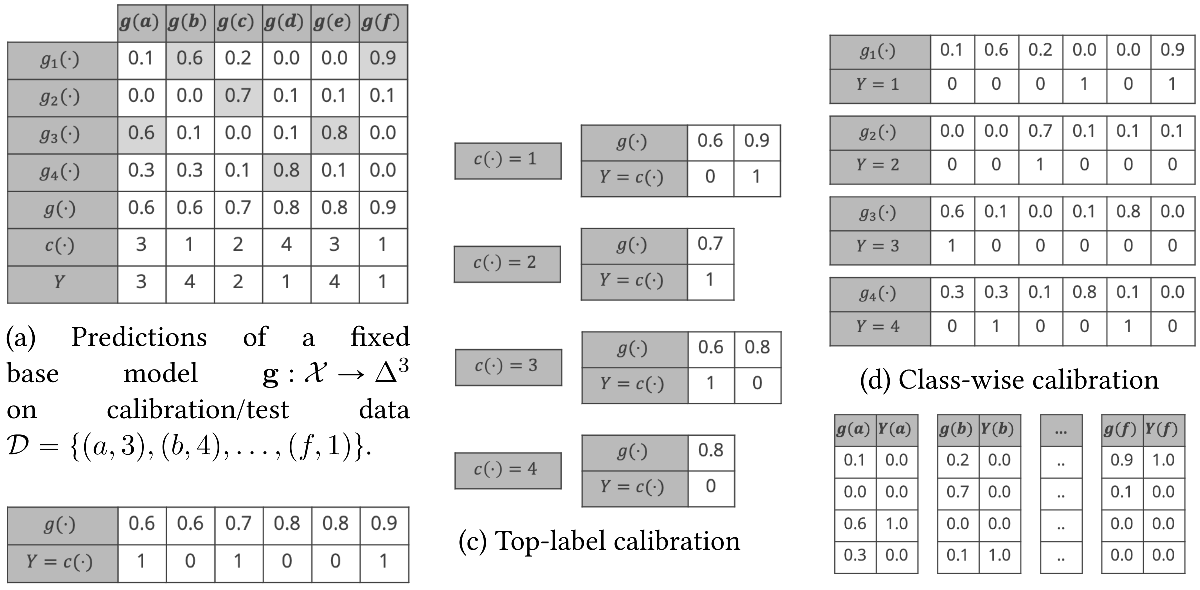
argue that $\mathcal{D}$ should be used in the *same way, whether testing or calibrating*. Thus, if confidence calibration is to be achieved, we should focus on the $(g, \mathbb{1}\{Y = c(\cdot)\})$ corresponding to $\mathbf{g}$. If top-label calibration is to be achieved, we should use the (c, g) values. If class-wise calibration is to be achieved, we should look at each g_l separately and solve L different problems. Finally, for canonical calibration, we must look at the entire $\mathbf{g}$ vector as a single unit. This is the core philosophy behind M2B calibrators: if binary claims are being verified, solve binary calibration problems.

5.B Distribution-free top-label calibration using histogram binning

In this section, we formally describe histogram binning (HB) with the top-label-calibrator (Algorithm 5.2) and provide methodological insights through theory and experiments.

5.B.1 Formal algorithm and theoretical guarantees

Algorithm 5.8 describes the top-label calibrator formally using HB as the binary calibration algorithm. The function called in line 6 is Algorithm 2 of Gupta and Ramdas (2021). The first argument in the call is the top-label confidence predictor, the second argument is the dataset to be used, the third argument is the number of bins to be used, and the fourth argument is a tie-breaking parameter (described shortly). While previous empirical works on HB fixed the *number of bins per class*, the analysis of Gupta and Ramdas (2021) suggests that a more principled way of choosing the number of bins is to fix the *number of points per bin*. This is parameter k of



	$g(a)$	$g(b)$	$g(c)$	$g(d)$	$g(e)$	$g(f)$
$g_1(\cdot)$	0.1	0.6	0.2	0.0	0.0	0.9
$g_2(\cdot)$	0.0	0.0	0.7	0.1	0.1	0.1
$g_3(\cdot)$	0.6	0.1	0.0	0.1	0.8	0.0
$g_4(\cdot)$	0.3	0.3	0.1	0.8	0.1	0.0
$g(\cdot)$	0.6	0.6	0.7	0.8	0.8	0.9
$c(\cdot)$	3	1	2	4	3	1
Y	3	4	2	1	4	1

(a) Predictions of a fixed base model $\mathbf{g} : \mathcal{X} \to \Delta^3$ on calibration/test data $\mathcal{D} = \{(a, 3), (b, 4), \ldots, (f, 1)\}$.

$g(\cdot)$	0.6	0.6	0.7	0.8	0.8	0.9
$Y = c(\cdot)$	1	0	1	0	0	1

(b) Confidence calibration

$c(\cdot) = 1$

$g(\cdot)$	0.6	0.9
$Y = c(\cdot)$	0	1

$c(\cdot) = 2$

$g(\cdot)$	0.7
$Y = c(\cdot)$	1

$c(\cdot) = 3$

$g(\cdot)$	0.6	0.8
$Y = c(\cdot)$	1	0

$c(\cdot) = 4$

$g(\cdot)$	0.8
$Y = c(\cdot)$	0

(c) Top-label calibration

$g_1(\cdot)$	0.1	0.6	0.2	0.0	0.0	0.9
$Y = 1$	0	0	0	1	0	1

$g_2(\cdot)$	0.0	0.0	0.7	0.1	0.1	0.1
$Y = 2$	0	0	1	0	0	0

$g_3(\cdot)$	0.6	0.1	0.0	0.1	0.8	0.0
$Y = 3$	1	0	0	0	0	0

$g_4(\cdot)$	0.3	0.3	0.1	0.8	0.1	0.0
$Y = 4$	0	1	0	0	1	0

(d) Class-wise calibration

$g(a)$	$Y(a)$
0.1	0.0
0.0	0.0
0.6	1.0
0.3	0.0

$g(b)$	$Y(b)$
0.2	0.0
0.7	0.0
0.0	0.0
0.1	1.0

...
..
..
..
..

$g(f)$	$Y(f)$
0.9	1.0
0.1	0.0
0.0	0.0
0.0	0.0

(e) Canonical calibration

Figure 5.3: Illustrative example for Section 5.A.2. The numbers in plot (a) correspond to the predictions made by g on a dataset $\mathcal{D}$. If $\mathcal{D}$ were a test set, plots (b–e) show how it should be used to verify if g satisfies the corresponding notion of calibration. Consequently, we argue that if $\mathcal{D}$ were a calibration set, and we want to achieve one of the notions (b–e), then the data shown in the corresponding plots should be the data used to calibrate g as well.

Algorithm 5.8 Top-label histogram binning

```
Require: Base multiclass predictor g, calibration data $\mathcal{D} = (X_1, Y_1), \ldots, (X_n, Y_n)$
Require: # points per bin $k \in \mathbb{N}$ (say 50), tie-breaking parameter $\delta > 0$ (say $10^{-10}$)
Ensure: Top-label calibrated predictor $(c, h)$
$c \leftarrow$ classifier or top-class based on g
$g \leftarrow$ top-class-probability based on g
for $l \leftarrow 1$ to $L$ do
    $\mathcal{D}_l \leftarrow \{(X_i, \mathbb{1}\{Y_i = l\}) : c(X_i) = l\}$ and
    $n_l \leftarrow |\mathcal{D}_l|$
    $h_l \leftarrow$ Binary-histogram-binning$(g, \mathcal{D}_l, \lfloor n_l/k \rfloor, \delta)$
end for
$h(\cdot) \leftarrow h_{c(\cdot)}(\cdot)$
return $(c, h)$
```

Algorithm 5.8. Given k, the number of bins is decided separately for every class as $\lfloor n_l/k \rfloor$ where n_l is the number of points predicted as class l. This choice is particularly relevant for top-label calibration since n_l can be highly non-uniform (we illustrate this empirically in Section 5.B.2). The tie-breaking parameter δ can be arbitrarily small (like 10^{-10}), and its significance is mostly theoretical—it is used to ensure that outputs of different bins are not exactly identical by chance, so that conditioning on a calibrated probability output is equivalent to conditioning on a bin;

this leads to a cleaner theoretical guarantee.

HB recalibrates g to a piecewise constant function h that takes one value per bin. Consider a specific bin b; the h value for this bin is computed as the average of the indicators $\{\mathbb{1}\{Y_i = c(X_i)\} : X_i \in \text{Bin } b\}$. This is an estimate of the *bias* of the bin $\mathbb{P}(Y = c(X) \mid X \in \text{Bin } b)$. A concentration inequality can then be used to bound the deviation between the estimate and the true bias to prove distribution-free calibration guarantees. In the forthcoming Theorem 5.1, we show high-probability and in-expectation bounds on the the TL-ECE of HB. Additionally, we show marginal and conditional top-label calibration bounds, defined next. These notions were proposed in the binary calibration setting by Gupta et al. (2020) and Gupta and Ramdas (2021). In the definition below, $\mathcal{A}$ refers to any algorithm that takes as input calibration data $\mathcal{D}$ and an initial classifier $\mathbf{g}$ to produce a top-label predictor c and an associated probability map h. Algorithm 5.8 is an example of $\mathcal{A}$.

Definition 5.1 (Marginal and conditional top-label calibration)**.** Let $\epsilon, \alpha \in (0, 1)$ be some given levels of approximation and failure respectively. An algorithm $\mathcal{A} : (\mathbf{g}, \mathcal{D}) \mapsto (c, h)$ is

(a) (ϵ, α)-marginally top-label calibrated if for every distribution P over $\mathcal{X} \times [L]$,

$$\mathbb{P}\Big(|\mathbb{P}(Y = c(X) \mid c(X), h(X)) - h(X)| \leqslant \epsilon \Big) \geqslant 1 - \alpha. \tag{5.8}$$

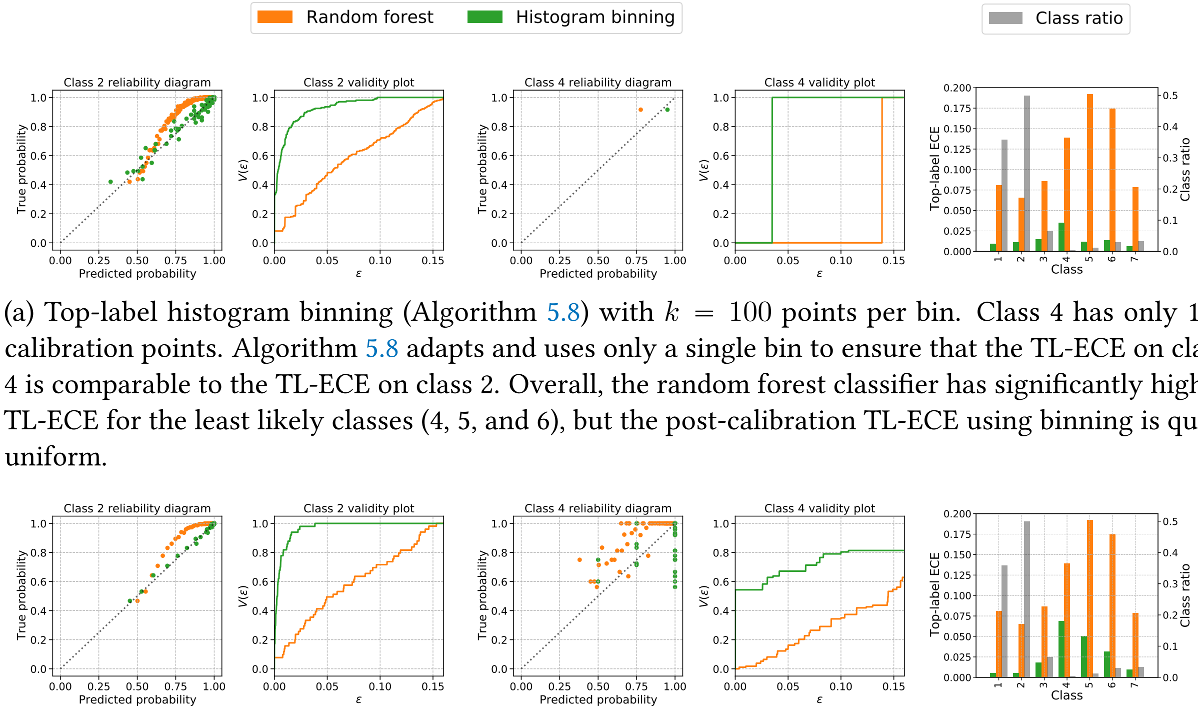
(b) (ϵ, α)-conditionally top-label calibrated if for every distribution P over $\mathcal{X} \times [L]$,

$$\mathbb{P}\Big(\forall\, l \in [L], r \in \text{Range}(h), |\mathbb{P}(Y = c(X) \mid c(X) = l, h(X) = r) - r| \leqslant \epsilon\Big) \geqslant 1 - \alpha. \tag{5.9}$$

To clarify, all probabilities are taken over the test point $(X, Y) \sim P$, the calibration data $\mathcal{D} \sim P^n$, and any other inherent algorithmic randomness in $\mathcal{A}$; these are all implicit in $(c, h) = \mathcal{A}(\mathcal{D}, \mathbf{g})$. Marginal calibration asserts that with high probability, on average over the distribution of $\mathcal{D}, X, \mathbb{P}(Y = c(X) \mid c(X), h(X))$ is at most ϵ away from $h(X)$. In comparison, TL-ECE is the average of these deviations over X. Marginal calibration may be a more appropriate metric for calibration than TL-ECE if we are somewhat agnostic to probabilistic errors less than some fixed threshold ϵ (like 0.05). Conditional calibration is a strictly stronger definition that requires the deviation to be at most ϵ for every possible prediction (l, r), including rare ones, not just on average over predictions. This may be relevant in medical settings where we want the prediction on every patient to be reasonably calibrated. Algorithm 5.8 satisfies the following calibration guarantees.

Theorem 5.1. *Fix hyperparameters $\delta > 0$ (arbitrarily small) and points per bin $k \geqslant 2$, and assume $n_l \geqslant k$ for every $l \in [L]$. Then, for any $\alpha \in (0, 1)$, Algorithm 5.8 is (ϵ_1, α)-marginally and (ϵ_2, α)-conditionally top-label calibrated for*

$$\epsilon_1 = \sqrt{\frac{\log(2/\alpha)}{2(k-1)}} + \delta, \qquad \textit{and} \qquad \epsilon_2 = \sqrt{\frac{\log(2n/k\alpha)}{2(k-1)}} + \delta. \tag{5.10}$$



(a) Top-label histogram binning (Algorithm 5.8) with $k = 100$ points per bin. Class 4 has only 183 calibration points. Algorithm 5.8 adapts and uses only a single bin to ensure that the TL-ECE on class 4 is comparable to the TL-ECE on class 2. Overall, the random forest classifier has significantly higher TL-ECE for the least likely classes (4, 5, and 6), but the post-calibration TL-ECE using binning is quite uniform.

(b) Histogram binning with $B = 50$ bins for every class. Compared to Figure 5.4a, the post-calibration TL-ECE for the most likely classes decreases while the TL-ECE for the least likely classes increases.

Figure 5.4: Recalibration of a random forest using histogram binning on the class imbalanced COVTYPE-7 dataset (class 2 is roughly 100 times likelier than class 4). By ensuring a fixed number of calibration points per bin, Algorithm 5.8 obtains relatively uniform top-label calibration across classes (Figure 5.4a). In comparison, if a fixed number of bins are chosen for all classes, the performance deteriorates for the least likely classes (Figure 5.4b).

Further, for any distribution P over $\mathcal{X} \times [L]$, we have $P(\text{TL-ECE}(c, h) \leqslant \epsilon_2) \geqslant 1 - \alpha$, and $\mathbb{E}\left[\text{TL-ECE}(c, h)\right] \leqslant \sqrt{1/2k} + \delta$.

The proof in Appendix 5.H is a multiclass top-label adaption of the guarantee in the binary setting by Gupta and Ramdas (2021). The $\widetilde{O}(1/\sqrt{k})$ dependence of the bound relies on Algorithm 5.8 delegating at least k points to every bin. Since δ can be chosen to be arbitrarily small, setting $k = 50$ gives roughly $\mathbb{E}_{\mathcal{D}}\left[\text{TL-ECE}(h)\right] \leqslant 0.1$. Base on this, we suggest setting $k \in [50, 150]$ in practice.

5.B.2 Top-label histogram binning adapts to class imbalanced datasets

The principled methodology of fixing the number of points per bin reaps practical benefits. Figure 5.4 illustrates this through the performance of HB for the class imbalanced COVTYPE-7 dataset (Blackard and Dean, 1999) with class ratio approximately 36% for class 1 and 49% for class 2. The entire dataset has 581012 points which is divided into train-test in the ratio 70:30. Then, 10% of the training points are held out for calibration ($n = |\mathcal{D}| = 40671$). The base classifier is a random forest (RF) trained on the remaining training points (it achieves around 95% test accuracy). The RF is then recalibrated using HB. The top-label reliability diagrams in Figure 5.4a illustrate that the original RF (in orange) is *underconfident* on both the most likely and least likely classes. Additional figures in Appendix 5.F show that the RF is always underconfident no matter which class is predicted as the top-label. HB (in green) recalibrates the RF effectively across all classes. Validity plots (Gupta and Ramdas, 2021) estimate how the LHS of condition (5.8), denoted as $V(\epsilon)$, varies with ϵ. We observe that for all ϵ, $V(\epsilon)$ is higher for HB. The rightmost barplot compares the estimated TL-ECE for all classes, and also shows the class proportions. While the original RF is significantly miscalibrated for the less likely classes, HB has a more uniform miscalibration across classes. Figure 5.4b considers a slightly different HB algorithm where the number of points per class is not adapted to the number of times the class is predicted, but is fixed beforehand (this corresponds to replacing $\lfloor n_l/k \rfloor$ in line 6 of Algorithm 5.8 with a fixed $B \in \mathbb{N}$). While even in this setting there is a drop in the TL-ECE compared to the RF model, the final profile is less uniform compared to fixing the number of points per bin.

The validity plots and top-label reliability diagrams for all the 7 classes are reported in Figure 5.9 in Appendix 5.F, along with some additional observations.

5.C Distribution-free class-wise calibration using histogram binning

In this section, we formally describe histogram binning (HB) with the class-wise-calibrator (Algorithm 5.3) and provide theoretical guarantees for it. The overall procedure is called class-wise-HB. Further details and background on HB are contained in Appendix 5.B, where top-label-HB is described.

5.C.1 Formal algorithm

To achieve class-wise calibration using binary routines, we learn each component function h_l in a 1-v-all fashion as described in Algorithm 5.3. Algorithm 5.9 contains the pseudocode with the underlying routine as binary HB. To learn h_l, we use a dataset $\mathcal{D}_l$, which unlike top-label HB (Algorithm 5.8), contains X_i even if $c(X_i) \neq l$. However the Y_i is replaced with $\mathbb{1}\{Y_i = l\}$. The number of points per bin k_l can be different for different classes, but generally one would set $k_1 = \ldots = k_L = k \in \mathbb{N}$. Larger values of k_l will lead to smaller ϵ_l and δ_l in the guarantees, at loss of sharpness since the number of bins $\lfloor n/k_l \rfloor$ would be smaller.

5.C.2 Calibration guarantees

A general algorithm $\mathcal{A}$ for class-wise calibration takes as input calibration data $\mathcal{D}$ and an initial classifier $\mathbf{g}$ to produce an approximately class-wise calibrated predictor $\mathbf{h} : \mathcal{X} \to [0,1]^L$. Define the notation $\boldsymbol{\varepsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_L) \in (0,1)^L$ and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_L) \in (0,1)^L$.

Algorithm 5.9 Class-wise histogram binning

```
Require: Base multiclass predictor $\mathbf{g} : \mathcal{X} \to \Delta^{L-1}$, calibration data $\mathcal{D} = (X_1, Y_1), \dots, (X_n, Y_n)$
Require: # points per bin $k_1, k_2, \dots, k_l \in \mathbb{N}^L$ (say each $k_l = 50$), tie-breaking parameter $\delta > 0$
    (say $10^{-10}$)
Ensure: $L$ class-wise calibrated predictors $h_1, h_2, \dots, h_L$
for $l \leftarrow 1$ to $L$ do
    $\mathcal{D}_l \leftarrow \{(X_i, \mathbb{1}\{Y_i = l\}) : i \in [n]\}$
    $h_l \leftarrow$ Binary-histogram-binning$(g_l, \mathcal{D}_l, \lfloor n/k_l \rfloor, \delta)$
end for
return $(h_1, h_2, \dots, h_L)$
```

Definition 5.2 (Marginal and conditional class-wise calibration)**.** Let $\boldsymbol{\varepsilon}, \boldsymbol{\alpha} \in (0,1)^L$ be some given levels of approximation and failure respectively. An algorithm $\mathcal{A} : (\mathbf{g}, \mathcal{D}) \mapsto \mathbf{h}$ is

(a) $(\boldsymbol{\varepsilon}, \boldsymbol{\alpha})$-marginally class-wise calibrated if for every distribution P over $\mathcal{X} \times [L]$ and for every $l \in [L]$

$$\mathbb{P}\Big(\left|\mathbb{P}(Y = l \mid h_l(X)) - h_l(X)\right| \leqslant \epsilon_l \Big) \geqslant 1 - \alpha_l. \tag{5.11}$$

(b) $(\boldsymbol{\varepsilon}, \boldsymbol{\alpha})$-conditionally class-wise calibrated if for every distribution P over $\mathcal{X} \times [L]$ and for every $l \in [L]$,

$$\mathbb{P}\Big(\forall r \in \text{Range}(h_l), \left|\mathbb{P}(Y = l \mid h_l(X) = r) - r\right| \leqslant \epsilon_l \Big) \geqslant 1 - \alpha_l. \tag{5.12}$$

Definition 5.2 requires that each h_l is (ϵ_l, α_l) calibrated in the binary senses defined by Gupta et al. (2021, Definitions 1 and 2). From Definition 5.2, we can also *uniform* bounds that hold simultaneously over every $l \in [L]$. Let $\alpha = \sum_{l=1}^{L} \alpha_l$ and $\epsilon = \max_{l \in [L]} \epsilon_l$. Then (5.11) implies

$$\mathbb{P}\Big(\forall l \in [L], \left|\mathbb{P}(Y = l \mid h_l(X)) - h_l(X)\right| \leqslant \epsilon \Big) \geqslant 1 - \alpha, \tag{5.13}$$

and (5.12) implies

$$\mathbb{P}\Big(\forall l \in [L], r \in \text{Range}(h_l), \left|\mathbb{P}(Y = l \mid h_l(X) = r) - r\right| \leqslant \epsilon \Big) \geqslant 1 - \alpha. \tag{5.14}$$

The choice of not including the uniformity over L in Definition 5.2 reveals the nature of our class-wise HB algorithm and the upcoming theoretical guarantees: (a) we learn the h_l's separately for each l and do not combine the learnt functions in any way (such as normalization), (b) we do not combine the calibration inequalities for different $[L]$ in any other way other than a union bound. Thus the only way we can show (5.13) (or (5.14)) is by using a union bound over (5.11) (or (5.12)).

We now state the distribution-free calibration guarantees satisfied by Algorithm 5.9.

Theorem 5.2. *Fix hyperparameters $\delta > 0$ (arbitrarily small) and points per bin $k_1, k_2, \dots, k_l \geqslant 2$, and assume $n_l \geqslant k_l$ for every $l \in [L]$. Then, for every $l \in [L]$, for any $\alpha_l \in (0,1)$, Algorithm 5.9 is*

$(\boldsymbol{\varepsilon}^{(1)}, \boldsymbol{\alpha})$*-marginally and* $(\boldsymbol{\varepsilon}^{(2)}, \boldsymbol{\alpha})$*-conditionally class-wise calibrated with*

$$\epsilon_l^{(1)} = \sqrt{\frac{\log(2/\alpha_l)}{2(k_l - 1)}} + \delta, \qquad \textit{and} \qquad \epsilon_l^{(2)} = \sqrt{\frac{\log(2n/k_l\alpha_l)}{2(k_l - 1)}} + \delta. \tag{5.15}$$

Further, for any distribution P *over* $\mathcal{X} \times [L]$*,*

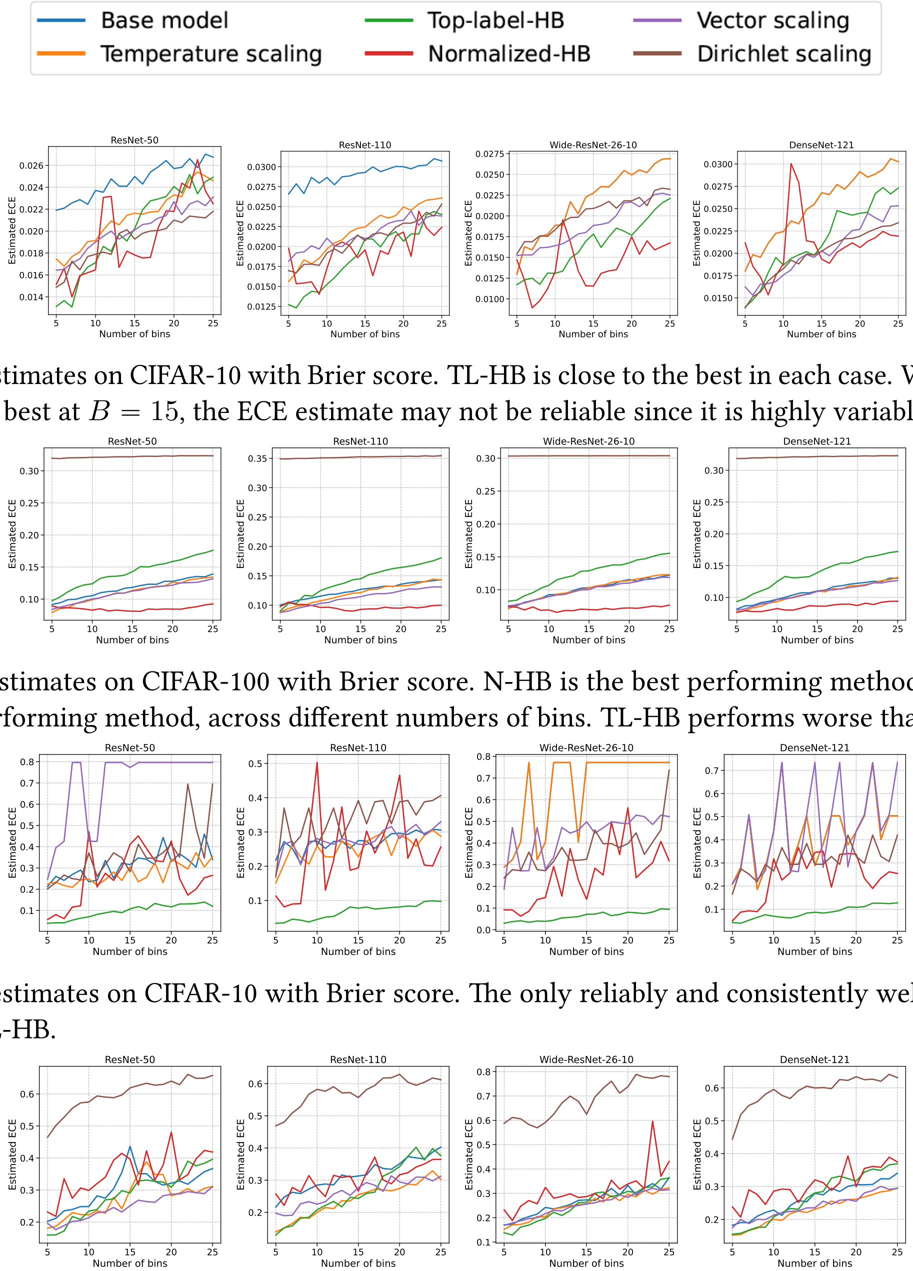
- *(a)* $P(\textit{CW-ECE}(c, h) \leqslant \max_{l\in[L]} \epsilon_l^{(2)}) \geqslant 1 - \sum_{l\in[L]} \alpha_l$*, and*
- *(b)* $\mathbb{E}\left[\textit{CW-ECE}(c, h)\right] \leqslant \max_{l\in[L]} \sqrt{1/2k_l} + \delta.$

Theorem 5.2 is proved in Appendix 5.H. The proof follows by using the result of Gupta and Ramdas (2021, Theorem 2), derived in the binary calibration setting, for each h_l separately. Gupta and Ramdas (2021) proved a more general result for general ℓ_p-ECE bounds. Similar results can also be derived for the suitably defined ℓ_p-CW-ECE.

As discussed in Section 5.3.2, unlike previous works (Zadrozny and Elkan, 2002; Guo et al., 2017; Kull et al., 2019), Algorithm 5.9 does not normalize the h_l's. We do not know how to derive Theorem 5.2 style results for a normalized version of Algorithm 5.9.

5.D Figures for Appendix 5.E

Appendix 5.E begins on page 119. The relevant figures for Appendix 5.E are displayed on the following pages.



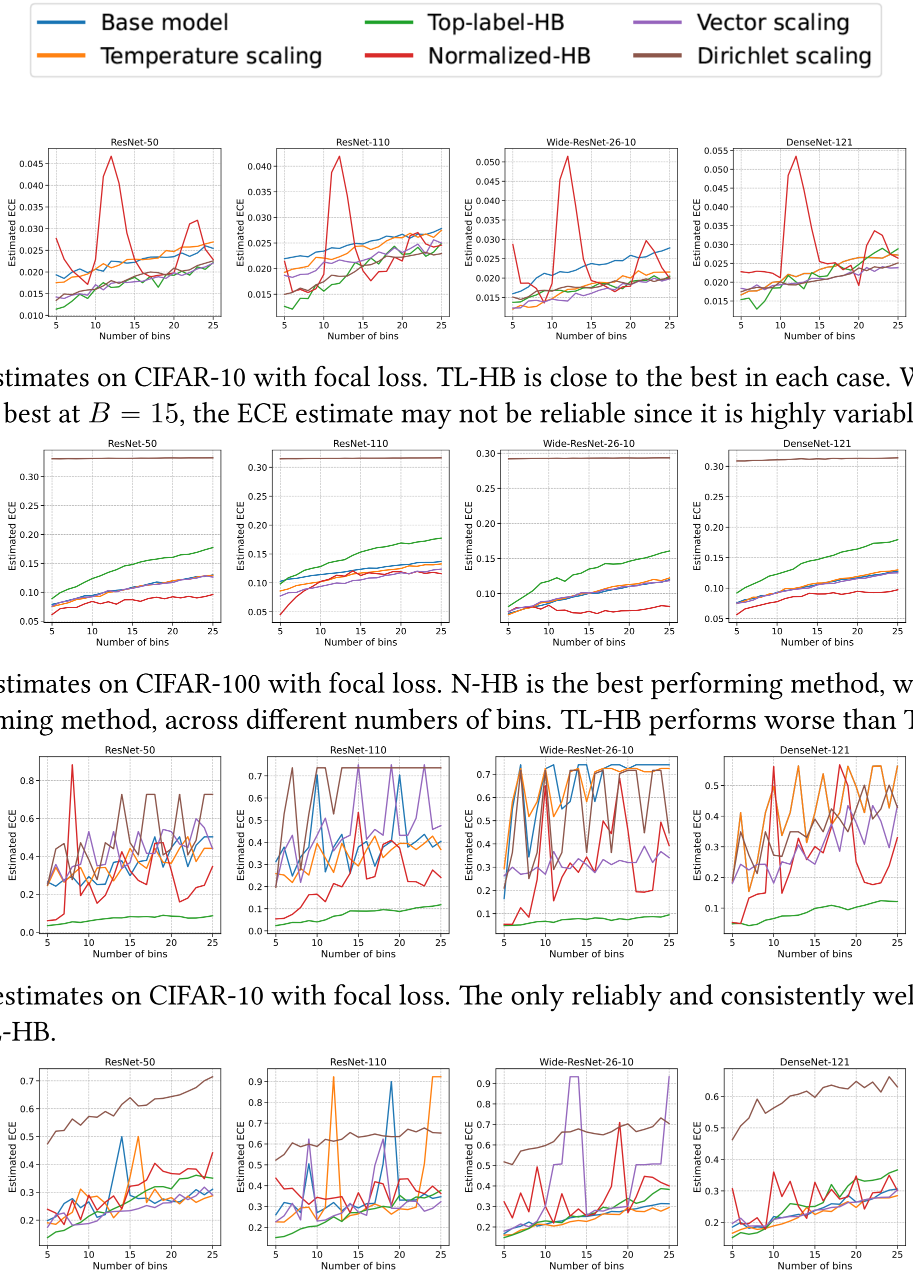
(a) TL-ECE estimates on CIFAR-10 with Brier score. TL-HB is close to the best in each case. While CW-HB performs the best at $B = 15$, the ECE estimate may not be reliable since it is highly variable across bins.

(b) TL-ECE estimates on CIFAR-100 with Brier score. N-HB is the best performing method, while DS is the worst performing method, across different numbers of bins. TL-HB performs worse than TS and VS.

(c) TL-MCE estimates on CIFAR-10 with Brier score. The only reliably and consistently well-performing method is TL-HB.

(d) TL-MCE estimates on CIFAR-100 with Brier score. DS is the worst performing method. Other methods perform across different values of B.

Figure 5.5: Table 5.2 style results with the number of bins varied as $B \in [5, 25]$. See Appendix 5.E.5 for further details. The captions summarize the findings in each case. In most cases, the findings are similar to those with $B = 15$. The notable exception is that performance of N-HB on CIFAR-10 for TL-ECE while very good at $B = 15$, is quite inconsistent when seen across different bins. In some cases, the blue base model line and the orange temperature scaling line coincide. This occurs since the optimal temperature on the calibration data was learnt to be $T = 1$, which corresponds to not changing the base model at all.



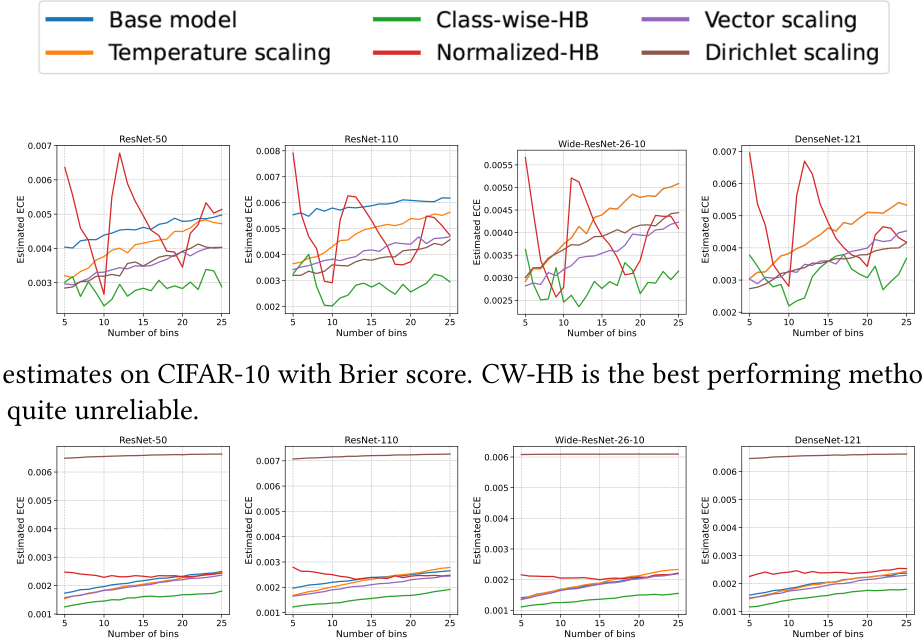
(a) TL-ECE estimates on CIFAR-10 with focal loss. TL-HB is close to the best in each case. While CW-HB performs the best at $B = 15$, the ECE estimate may not be reliable since it is highly variable across bins.

(b) TL-ECE estimates on CIFAR-100 with focal loss. N-HB is the best performing method, while DS is the worst performing method, across different numbers of bins. TL-HB performs worse than TS and VS.

(c) TL-MCE estimates on CIFAR-10 with focal loss. The only reliably and consistently well-performing method is TL-HB.

(d) TL-MCE estimates on CIFAR-100 with focal loss. DS is the worst performing method. Other methods perform across different values of B.

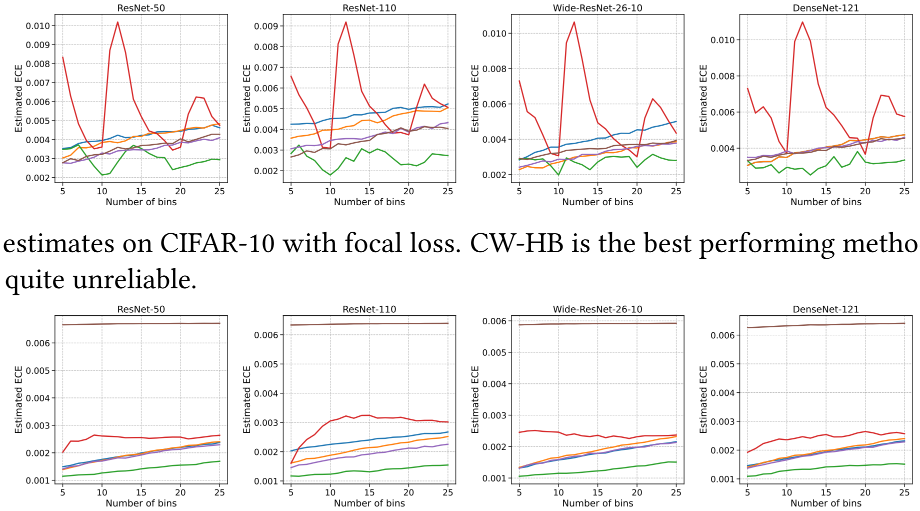
Figure 5.6: Table 5.4 style results with the number of bins varied as $B \in [5, 25]$. See Appendix 5.E.5 for further details. The captions summarize the findings in each case. In most cases, the findings are similar to those with $B = 15$. In some cases, the blue base model line and the orange temperature scaling line coincide. This occurs since the optimal temperature on the calibration data was learnt to be $T = 1$, which corresponds to not changing the base model at all.



(a) CW-ECE estimates on CIFAR-10 with Brier score. CW-HB is the best performing method across bins, and N-HB is quite unreliable.

(b) CW-ECE estimates on CIFAR-100 with Brier score. CW-HB is the best performing method. DS and N-HB are the worst performing methods.

Figure 5.7: Table 5.3 style results with the number of bins varied as $B \in [5, 25]$. The captions summarize the findings in each case, which are consistent with those in the table. See Appendix 5.E.5 for further details.



(a) CW-ECE estimates on CIFAR-10 with focal loss. CW-HB is the best performing method across bins, and N-HB is quite unreliable.

(b) CW-ECE estimates on CIFAR-100 with focal loss. CW-HB is the best performing method. DS and N-HB are the worst performing methods.

Figure 5.8: Table 5.5 style results with the number of bins varied as $B \in [5, 25]$. The captions summarize the findings in each case, which are consistent with those in the table. See Appendix 5.E.5 for further details.

5.E Additional experimental details and results for CIFAR-10 and CIFAR-100

We present additional details and results to supplement the experiments with CIFAR-10 and CIFAR-100 in Sections 5.2 and 5.4 of the main chapter.

5.E.1 External libraries used

All our base models were pre-trained deep-net models generated by Mukhoti et al. (2020), obtained from `www.robots.ox.ac.uk/~viveka/focal_calibration/` and used along with the code at `https://github.com/torrvision/focal_calibration` to obtain base predictions. We focused on the models trained with Brier score and focal loss, since it was found to perform the best for calibration. All results in the main chapter are with the Brier score; in Appendix 5.E.4, we report corresponding results with focal loss.

We also used the code at `https://github.com/torrvision/focal_calibration` for temperature scaling (TS). For vector scaling (VS) and Dirichlet scaling (DS), we used the code of Kull et al. (2019), hosted at `https://github.com/dirichletcal/dirichlet_python`. For VS, we used the file `dirichletcal/calib/vectorscaling.py`, and for DS, we used the file `dirichletcal/calib/fulldirichlet.py`. No hyperparameter tuning was performed in any of our histogram binning experiments or baseline experiments; default settings were used in every case. The random seed was fixed so that every run of the experiment gives the same result. In particular, by relying on pre-trained models, we avoid training new deep-net models with multiple hyperparameters, thus avoiding any selection biases that may arise due to test-data peeking across multiple settings.

5.E.2 Further comments on binning for ECE estimation

As mentioned in Remark 5.1, ECE estimates for all methods except TL-HB and CW-HB was done using fixed-width bins $[0, 1/B), [1/B, 2/B), \ldots [1-1/B, 1]$ for various values of $B \in [5, 25]$. For TL-HB and CW-HB, B is the number of bins used for each call to binary HB. For TL-HB, note that we actually proposed that the number of bins-per-class should be fixed; see Section 5.B.2. However, for ease of comparison to other methods, we simply set the number of bins to B for each call to binary HB. That is, in line 6, we replace $\lfloor n_l/k \rfloor$ with B. For CW-HB, we described Algorithm 5.9 with different values of k_l corresponding to the number of bins per class. For the CIFAR-10 and CIFAR-100 comparisons, we set each $k_1 = k_2 = \ldots = k_L = k$, where $k \in \mathbb{N}$ satisfies $\lfloor n/k \rfloor = B$.

Tables 5.2,5.3, 5.4, and 5.5 report estimates with $B = 15$, which has been commonly used in many works (Guo et al., 2017; Kull et al., 2019; Mukhoti et al., 2020). Corresponding to each table, we have a figure where ECE estimates with varying B are reported to strengthen conclusions: these are Figure 5.5,5.7, 5.6, and 5.8 respectively. Plugin estimates of the ECE were used, same as Guo et al. (2017). Further binning was not done for TL-HB and CW-HB since the output is already discrete and sufficiently many points take each of the predicted values. Note that due to Jensen's inequality, any further binning will only decrease the ECE estimate (Kumar et al., 2019). Thus, using unbinned estimates may give TL-HB and CW-HB a disadvantage.

5.E.3 Some remarks on maximum-calibration-error (MCE)

Guo et al. (2017) defined MCE with respect to confidence calibration, as follows:

$$\text{conf-MCE}(c, h) := \sup_{r \in \text{Range}(h)} \left| \mathbb{P}(Y = c(X) \mid h(X) = r) - r \right|. \tag{5.16}$$

Conf-MCE suffers from the same issue illustrated in Figure 5.2 for conf-ECE. In Figure 5.1b, we looked at the reliability diagram within two bins. These indicate two of the values over which the supremum is taken in equation (5.16): these are the Y-axis distances between the ★ markers and the $X = Y$ line for bins 6 and 10 (both are less than 0.02). On the other hand, the effective *maximum* miscalibration for bin 6 is roughly 0.15 (for class 1), and roughly 0.045 (for class 4), and the maximum should be taken with respect to these values across all bins. To remedy the underestimation of the effective MCE, we can consider the top-label-MCE, defined as

$$\text{TL-MCE}(c, h) := \max_{l \in [L]} \sup_{r \in \text{Range}(h)} \left| \mathbb{P}(Y = l \mid c(X) = l, h(X) = r) - r \right|. \tag{5.17}$$

Interpreted in words, the TL-MCE assesses the maximum deviation between the predicted and true probabilities across all predictions and all classes. Following the same argument as in the proof of Proposition 5.4, it can be shown that for any c, h, conf-MCE$(c, h) \leqslant$ TL-MCE(c, h). The TL-MCE is closely related to conditional top-label calibration (Definition 5.1b). Clearly, an algorithm is (ϵ, α)-conditionally top-label calibrated if and only if for every distribution P, $P(\text{TL-MCE}(c, h) \leqslant \epsilon) \geqslant 1 - \alpha$. Thus the conditional top-label calibration guarantee of Theorem 5.1 implies a high probability bound on the TL-MCE as well.

5.E.4 Table 5.2 and 5.3 style results with focal loss

Results for top-label-ECE and top-label-MCE with the base deep net model being trained using focal loss are reported in Table 5.4. Corresponding results for class-wise-ECE are reported in Table 5.5. The observations are similar to the ones reported for Brier score:

1. For TL-ECE, TL-HB is either the best or close to the best performing method on CIFAR-10, but suffers on CIFAR-100. This phenomenon is discussed further in Appendix 5.E.6. N-HB is the best or close to the best for both CIFAR-10 and CIFAR-100.
2. For TL-MCE, TL-HB is the best performing method on CIFAR-10, by a huge margin. For CIFAR-100, TS or VS perform better than TL-HB, but not by a huge margin.
3. For CW-ECE, CW-HB is the best performing method across the two datasets and all four architectures.

Metric	Dataset	Architecture	Base	TS	VS	DS	N-HB	TL-HB
Top-label-ECE	CIFAR-10	ResNet-50	0.022	0.023	**0.018**	0.019	0.023	0.019
		ResNet-110	0.025	0.024	0.022	0.021	**0.020**	**0.020**
		WRN-26-10	0.024	0.019	**0.016**	0.017	0.019	0.018
		DenseNet-121	0.023	0.023	**0.021**	0.021	0.025	**0.021**
	CIFAR-100	ResNet-50	0.109	0.107	0.107	0.332	**0.086**	0.148
		ResNet-110	0.124	0.117	**0.105**	0.316	0.115	0.153
		WRN-26-10	0.100	0.100	0.101	0.293	**0.074**	0.135
		DenseNet-121	0.106	0.108	0.105	0.312	**0.091**	0.147
Top-label-MCE	CIFAR-10	ResNet-50	0.298	0.443	0.368	0.472	0.325	**0.082**
		ResNet-110	0.378	0.293	0.750	0.736	0.535	**0.089**
		WRN-26-10	0.741	0.582	0.311	0.363	0.344	**0.075**
		DenseNet-121	0.411	0.411	0.243	0.391	0.301	**0.099**
	CIFAR-100	ResNet-50	0.289	0.355	**0.234**	0.640	0.322	0.273
		ResNet-110	0.293	**0.265**	0.274	0.633	0.366	0.272
		WRN-26-10	0.251	**0.227**	0.256	0.663	0.229	0.270
		DenseNet-121	0.237	**0.225**	0.239	0.597	0.327	0.248

Table 5.4: Top-label-ECE and top-label-MCE for deep-net models and various post-hoc calibrators. All methods are same as Table 5.2. Best performing method in each row is in **bold**.

Metric	Dataset	Architecture	Base	TS	VS	DS	N-HB	CW-HB
Class-wise-ECE $\times 10^2$	CIFAR-10	ResNet-50	0.42	0.42	**0.35**	0.37	0.52	**0.35**
		ResNet-110	0.48	0.44	0.36	0.35	0.51	**0.29**
		WRN-26-10	0.41	0.31	0.31	0.35	0.49	**0.27**
		DenseNet-121	0.41	0.41	0.40	0.39	0.63	**0.30**
	CIFAR-100	ResNet-50	0.22	0.20	0.20	0.66	0.23	**0.16**
		ResNet-110	0.24	0.23	0.21	0.72	0.24	**0.16**
		WRN-26-10	0.19	0.19	0.18	0.61	0.20	**0.14**
		DenseNet-121	0.20	0.21	0.19	0.66	0.24	**0.16**

Table 5.5: Class-wise-ECE for deep-net models and various post-hoc calibrators. All methods are same as Table 5.2, except top-label-HB is replaced with class-wise-HB or Algorithm 5.3 (CW-HB). Best performing method in each row is in **bold**.

5.E.5 ECE and MCE estimates with varying number of bins

Corresponding to each entry in Tables 5.2 and 5.4, we perform an ablation study with the number of bins varying as $B \in [5, 25]$. This is in keeping with the findings of Roelofs et al. (2022) that the ECE/MCE estimate can vary with different numbers of bins, along with the relative performance of the various models.

The results are reported in Figure 5.5 (ablation of Table 5.2) and Figure 5.7 (ablation of Table 5.3). The captions of these figures contain further details on the findings. Most findings are similar to those in the main chapter, but the findings in the tables are strengthened through this ablation.

The same ablations are performed for focal loss as well. The results are reported in Figure 5.6 (ablation of Table 5.4) and Figure 5.8 (ablation of Table 5.5). The captions of these figures contain further details on the findings. The ablation results in the figures support those in the tables.

5.E.6 Analyzing the poor performance of TL-HB on CIFAR-100

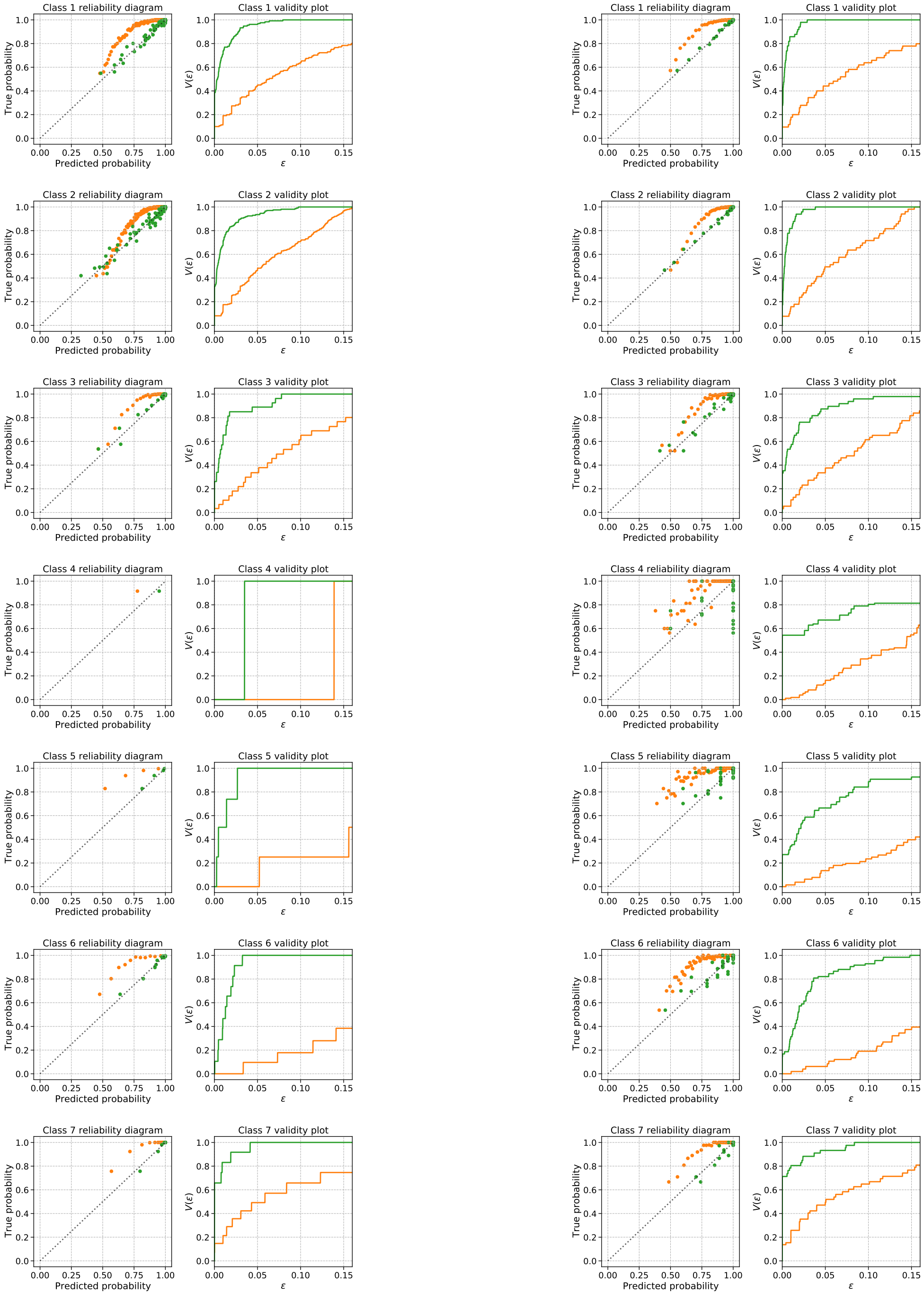
CIFAR-100 is an imbalanced dataset with 100 classes and 5000 points for validation/calibration (as per the default splits). Due to random subsampling, the validation split we used had one of the classes predicted as the top-label only 31 times. Thus, based on Theorem 5.1, we do not expect HB to have small TL-ECE. This is confirmed by the empirical results presented in Tables 5.2/5.4, and Figures 5.5b/5.6b. We observe that HB has higher estimated TL-ECE than all methods except DS, for most values of the number of bins. The performance of TL-HB for TL-MCE however is much much closer to the other methods since HB uses the same number of points per bin, ensuring that the predictions are somewhat equally calibrated across bins (Figures 5.5d/5.6d). In comparison, for CW-ECE, CW-HB is the best performing method. This is because in the class-wise setting, 5000 points are available for recalibration irrespective of the class, which is sufficient for HB.

The deterioration in performance of HB when few calibration points are available was also observed in the binary setting by Gupta and Ramdas (2021, Appendix C). Niculescu-Mizil and Caruana (2005) noted in the conclusion of their paper that Platt scaling (Platt, 1999), which is closely related to TS, performs well when the data is small, but another nonparametric binning method, isotonic regression (Zadrozny and Elkan, 2002) performs better when enough data is available. Kull et al. (2019, Section 4.1) compared HB to other calibration techniques for class-wise calibration on 21 UCI datasets, and found that HB performs the worst. On inspecting the UCI repository, we found that most of the datasets they used had fewer than 5000 (total) data points, and many contain fewer than 500.

Overall, comparing our results to previous empirical studies, we believe that if sufficiently many points are available for recalibration, or the number of classes is small, then HB performs quite well. To be more precise, we expect HB to be competitive if at least 200 points per class can be held out for recalibration, and the number of points per bin is at least $k \geqslant 20$.

5.F Additional experimental details and results for COVTYPE-7

We present additional details and results for the top-label HB experiment of Section 5.B.2. The base classifier is an RF learnt using the `sklearn.ensemble` module, with default parameters. The base RF is a nearly continuous base model since most predictions are unique. Thus, we need to use binning to make reliability diagrams, validity plots, and perform ECE estimation, for the base model. To have a fair comparison, instead of having a fixed binning scheme to assess the base model, the binning scheme was decided based on the unique predictions of top-label HB. Thus for every l, and $r \in \text{Range}(h_l)$, the bins are defined as $\{x : c(x) = l, h_l(x) = r\}$. Due to



(a) Top-label HB with $k = 100$ points per bin.

(b) Top-label HB with $B = 50$ bins per class.

Figure 5.9: Top-label histogram binning (HB) calibrates a miscalibrated random-forest on the class imbalanced COVTYPE-7 dataset. For the less likely classes (4, 5, and 6), the left column is better calibrated than the right column. Similar observations are made on other datasets, and so we recommend adaptively choosing a different number of bins per class, as Algorithm 5.8 does.

this, while the base model in Figures 5.4a and 5.4b are the same, the reliability diagrams and validity plots in orange are different. As can be seen in the bar plots in Figure 5.4, the ECE estimation is not affected significantly.

When $k = 100$, the total number of bins chosen by Algorithm 5.8 was 403, which is roughly 57.6 bins per class. The choice of $B = 50$ for the fixed bins per class experiment was made on this basis.

Figure 5.9 supplements Figure 5.4 in the main chapter by presenting reliability diagrams and validity plots of top-label HB for all classes. Figure 5.9a presents the plots with adaptive number of bins per class (Algorithm 5.8), and Figure 5.9b presents these for fixed number of bins per class. We make the following observations.

(a) For every class $l \in [L]$, the RF is overconfident. This may seem surprising at first since we generally expect that models may be overconfident for certain classes and underconfident for others. However, note that all our plots assess top-label calibration, that is, we are assessing the predicted and true probabilities of only the predicted class. It is possible that a model is overconfident for every class whenever that class is predicted to be the top-label.

(b) For the most likely classes, namely classes 1 and 2, the number of bins in the adaptive case is higher than 50. Fewer bins leads to better calibration (at the cost of sharpness). This can be verified through the validity plots for classes 1 and 2—the validity plots in the fixed bins case is slightly *above* the validity plot in the adaptive bin case. However both validity plots are quite similar.

(c) The opposite is true for the least likely classes, namely classes 4, 5, 6. The validity plot in the fixed bins case is *below* the validity plot in the adaptive bins case, indicating higher TL-ECE in the fixed bins case. The difference between the validity plots is high. Thus if a fixed number of bins per class is pre-decided, the performance for the least likely classes significantly suffers.

Based on these observations, we recommend adaptively choosing the number of bins per class, as done by Algorithm 5.8.

5.G Binning-based calibrators for canonical multiclass calibration

Canonical calibration is a notion of calibration that does not fall in the M2B category. To define canonical calibration, we use $\mathbf{Y}$ to denote the output as a 1-hot vector. That is, $\mathbf{Y}_i = \mathbf{e}_{Y_i} \in \Delta^{L-1}$, where e_l corresponds to the l-th canonical basis vector in $\mathbb{R}^d$. Recall that a predictor $\mathbf{h} = (h_1, h_2, \dots, h_L)$ is said to be canonically calibrated if $\mathbb{P}(Y = l \mid \mathbf{h}(X)) = h_l(X)$ for every $l \in [L]$. Equivalently, this can be stated as $\mathbb{E}\left[\mathbf{Y} \mid \mathbf{h}(X)\right] = \mathbf{h}(X)$. Canonical calibration implies class-wise calibration:

Proposition 5.1. *If $\mathbb{E}\left[\mathbf{Y} \mid \mathbf{h}(X)\right] = \mathbf{h}(X)$, then for every $l \in [L]$, $\mathbb{P}(Y = l \mid h_l(X)) = h_l(X)$.*

The proof in Appendix 5.H is straightforward, but the statement above is illuminating, because there exist predictors that are class-wise calibrated but not canonically calibrated (Vaicenavicius et al., 2019, Example 1).

Canonical calibration is not an M2B notion since the conditioning occurs on the L-dimensional prediction vector $\text{pred}(X) = \mathbf{h}(X)$, and after this conditioning, each of the L statements $P(Y = l \mid \text{pred}(X)) = h_l(X)$ should *simultaneously* be true. On the other hand, M2B notions verify only individual binary calibration claims for every such conditioning. Since canonical calibration does not fall in the M2B category, Algorithm 5.5 does not lead to a calibrator for canonical calibration. In this section, we discuss alternative binning-based approaches to achieving canonical calibration.

For binary calibration, there is a complete ordering on the interval $[0, 1]$, and this ordering is leveraged by binning based calibration algorithms. However, Δ^{L-1}, for $L \geqslant 3$ does not have such a natural ordering. Hence, binning algorithms do not obviously extend for multiclass classification. In this section, we briefly discuss some binning-based calibrators for canonical calibration. Our descriptions are for general $L \geqslant 3$, but we anticipate these algorithms to work reasonably only for small L, say if $L \leqslant 5$.

As usual, denote $\mathbf{g} : \mathcal{X} \to \Delta^{L-1}$ as the base model and $\mathbf{h} : \mathcal{X} \to \Delta^{L-1}$ as the model learnt using some post-hoc canonical calibrator. For canonical calibration, we can surmise binning schemes that directly learn $\mathbf{h}$ by partitioning the prediction space Δ^{L-1} into bins and estimating the distribution of $\mathbf{Y}$ in each bin. A canonical calibration guarantee can be showed for such a binning scheme using multinomial concentration (Podkopaev and Ramdas, 2021, Section 3.1). However, since $\text{Vol}(\Delta^{L-1}) = 2^{\Theta(L)}$, there will either be a bin whose volume is $2^{\Omega(L)}$ (meaning that $\mathbf{h}$ would not be sharp), or the number of bins will be $2^{\Omega(L)}$, entailing $2^{\Omega(L)}$ requirements on the sample complexity—a *curse of dimensionality*. Nevertheless, let us consider some binning schemes that could work if L is small.

Formally, a binning scheme corresponds to a partitioning of Δ^{L-1} into $B \geqslant 1$ bins. We denote this binning scheme as $\mathcal{B} : \Delta^{L-1} \to [B]$, where $\mathcal{B}(\mathbf{s})$ corresponds to the bin to which $\mathbf{s} \in \Delta^{L-1}$ belongs. To learn $\mathbf{h}$, the calibration data is binned to get sets of data-point indices that belong to each bin, depending on the $\mathbf{g}(X_i)$ values:

$$\text{for every } b \in [B],\ T_b := \{i : \mathcal{B}(\mathbf{g}(X_i)) = b\}, n_b = |T_b|\,.$$

We then compute the following estimates for the label probabilities in each bin:

$$\text{for every } (l, b) \in [L] \times [B],\ \widehat{\Pi}_{l,b} := \frac{\sum_{i \in T_b} \mathbb{1}\left\{Y_i = l\right\}}{n_b} \text{ if } n_b > 0 \text{ else } \widehat{\Pi}_{l,b} = 1/B.$$

The binning predictor $\mathbf{h} : \mathcal{X} \to \Delta^{L-1}$ is now defined component-wise as follows:

$$\text{for every } l \in [L],\ h_l(x) = \widehat{\Pi}_{l,\mathcal{B}(x)}.$$

In words, for every bin $b \in [B]$, $\mathbf{h}$ predicts the empirical distribution of the Y values in bin b.

Using a multinomial concentration inequality (Devroye et al., 1983; Qian et al., 2020; Weissman et al., 2003), calibration guarantees can be shown for the learnt $\mathbf{h}$. Podkopaev and Ramdas

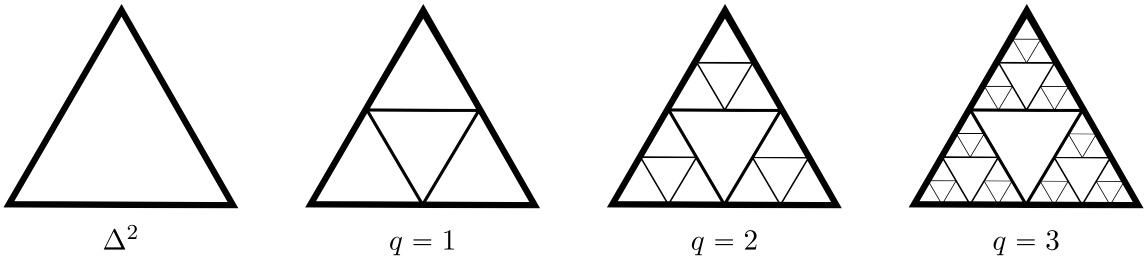


Figure 5.10: Sierpinski binning for $L = 3$. The leftmost triangle represents the probability simplex Δ^2. Sierpinski binning divides Δ^2 recursively based on a depth parameter $q \in \mathbb{N}$.

(2021, Theorem 3) show such a result using the Bretagnolle-Huber-Carol inequality. All of these concentration inequality give bounds that depend inversely on n_b or $\sqrt{n_b}$.

In the following subsections, we describe some binning schemes which can be used for canonical calibration based on the setup illustrated above. First we describe fixed schemes that are not adaptive to the distribution of the data: Sierpinski binning (Appendix 5.G.1) and grid-style binning (Appendix 5.G.2). These are analogous to fixed-width binning for $L = 2$. Fixed binning schemes are not adapted to the calibration data and may have highly imbalanced bins leading to poor estimation of the distribution of $\mathbf{Y}$ in bins with small n_b. In the binary case, this issue is remedied using histogram binning to ensure that each bin has nearly the same number of calibration points (Gupta and Ramdas, 2021). While histogram binning uses the order of the scalar $g(X_i)$ values, there is no obvious ordering for the multi-dimensional $\mathbf{g}(X_i)$ values. In Appendix 5.G.3 we describe a projection based histogram binning scheme that circumvents this issue and ensures that each n_b is reasonably large. In Appendix 5.G.4, we present some preliminary experimental results using our proposed binning schemes.

Certain asymptotic consistency results different from calibration have been established for histogram regression and classification in the nonparametric statistics literature (Nobel, 1996; Lugosi and Nobel, 1996; Gordon and Olshen, 1984; Breiman, 2017; Devroye, 1988); further extensive references can be found within these works. The methodology of histogram regression and classification relies on binning and is very similar to the one we propose here. The main difference is that these works consider binning the feature space $\mathcal{X}$ directly, unlike the post-hoc setting where we are essentially interested in binning Δ^{L-1}. In terms of theory, the results these works target are asymptotic consistency for the (Bayes) optimal classification and regression functions, instead of canonical calibration. It would be interesting to consider the (finite-sample) canonical calibration properties of the various algorithms proposed in the context of histogram classification. However, such a study is beyond the scope of our work.

5.G.1 Sierpinski binning

First, we describe Sierpinski binning for $L = 3$. The probability simplex for $L = 3$, Δ^2, is a triangle with vertices $\mathbf{e}_1 = (1, 0, 0)$, $\mathbf{e}_2 = (0, 1, 0)$, and $\mathbf{e}_3 = (0, 0, 1)$. Sierpinski binning is a recursive partitioning of this triangle based on the fractal popularly known as the Sierpinski triangles. Some Sierpinski bins for $L = 3$ are shown in Figure 5.10. Formally, we define Sierpinski binning recursively based on a depth parameter $q \in \mathbb{N}$. Given an $x \in \mathcal{X}$, let $\mathbf{s} = \mathbf{g}(x)$. For $q = 1$, the number of bins is $B = 4$, and the binning scheme $\mathcal{B}$ is defined as:

$$\mathcal{B}(\mathbf{s}) = \begin{cases} 1 & \text{if } s_1 > 0.5 \\ 2 & \text{if } s_2 > 0.5 \\ 3 & \text{if } s_3 > 0.5 \\ 4 & \text{otherwise.} \end{cases} \tag{5.18}$$

Note that since $s_1 + s_2 + s_3 = 1$, only one of the above conditions can be true. It can be verified that each of the bins have volume equal to $(1/4)$-th the volume of the probability simplex Δ^2. If a finer resolution of Δ^2 is desired, B can be increased by further dividing the partitions above. Note that each partition is itself a triangle; thus each triangle can be mapped to Δ^2 to recursively define the sub-partitioning. For $i \in [4]$, define the bins $b_i = \{\mathbf{s} : \mathcal{B}(\mathbf{s}) = i\}$. Consider the bin b_1. Let us *reparameterize* it as $(t_1, t_2, t_3) = (2s_1 - 1, 2s_2, 2s_3)$. It can be verified that

$$b_1 = \{(t_1, t_2, t_3) : s_1 > 0.5\} = \{(t_1, t_2, t_3) : t_1 + t_2 + t_3 = 1, t_1 \in (0, 1], t_2 \in [0, 1), t_3 \in [0, 1)\}.$$

Based on this reparameterization, we can recursively sub-partition b_1 as per the scheme (5.18), replacing s with t. Such reparameterizations can be defined for each of the bins defined in (5.18):

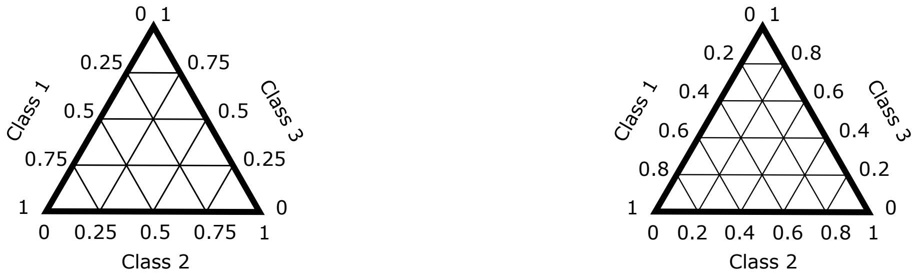
$$\begin{aligned} b_2 &= \{(s_1, s_2, s_3) : s_2 > 0.5\} : (t_1, t_2, t_3) = (2s_1, 2s_2 - 1, 2s_3), \\ b_3 &= \{(s_1, s_2, s_3) : s_3 > 0.5\} : (t_1, t_2, t_3) = (2s_1, 2s_2, 2s_3 - 1), \\ b_4 &= \{(s_1, s_2, s_3) : s_i \leqslant 0.5 \text{ for all } i\} : (t_1, t_2, t_3) = (1 - 2s_1, 1 - 2s_2, 1 - 2s_3), \end{aligned}$$

and thus every bin can be recursively sub-partitioned as per (5.18). As illustrated in Figure 5.10, for Sierpinski binning, we sub-partition only the bins b_1, b_2, b_3 since the bin b_4 corresponds to low confidence for all labels, where finer calibration may not be needed. (Also, in the $L > 3$ case described shortly, the corresponding version of b_4 is geometrically different from Δ^{L-1}, and the recursive partitioning cannot be defined for it.) If at every depth, we sub-partition all bins except the corresponding b_4 bins, then it can be shown using simple algebra that the total number of bins is $(3^{q+1} - 1)/2$. For example, in Figure 5.10, when $q = 2$, the number of bins is $B = 14$, and when $q = 3$, the number of bins is $B = 40$.

As in the case of $L = 3$, Sierpinski binning for general L is defined through a partitioning function of Δ^{L-1} into $L + 1$ bins, and a reparameterization of the partitions so that they can be further sub-partitioned. The $L + 1$ bins at depth $q = 1$ are defined as

$$\mathcal{B}(\mathbf{s}) = \begin{cases} l & \text{if } s_l > 0.5, \\ L + 1 & \text{otherwise.} \end{cases} \tag{5.19}$$

While this looks similar to the partitioning (5.18), the main difference is that the bin b_{L+1} has a larger volume than other bins. Indeed for $l \in [L]$, $\text{vol}(b_l) = \text{vol}(\Delta^{L-1})/2^{L-1}$, while $\text{vol}(b_{L+1}) = \text{vol}(\Delta^{L-1})(1 - L/2^{L-1}) \geqslant \text{vol}(\Delta^{L-1})/2^{L-1}$, with equality only occuring for $L = 3$. Thus the bin b_{L+1} is larger than the other bins. If $\mathbf{g}(x) \in b_{L+1}$, then the prediction for x may be not be very sharp, compared to if $\mathbf{g}(x)$ were in any of the other bins. On the other hand, if



$K = 4, \tau = 0.25$ $K = 5, \tau = 0.2$

Figure 5.11: Grid-style binning for $L = 3$.

$\mathbf{g}(x) \in b_{L+1}$, the score for every class is smaller than 0.5, and sharp calibration may often not be desired in this region.

In keeping with this understanding, we only reparameterize the bins $b_1, b_2, \ldots, b_L$ so that they can be further divided:

$$b_l = \{(s_1, s_2, \ldots, s_L) : s_l > 0.5\} : (t_1, t_2, \ldots, t_L) = (2s_1, \ldots, 2s_l - 1, \ldots, 2s_L).$$

For every $l \in [L]$, under the reparameterization above, it is straightforward to verify that

$$\{(t_1, t_2, \ldots, t_L) : s_l > 0.5\} = \{(t_1, t_2, \ldots, t_L) : \sum_{u \in [L]} t_u = 1, t_l \in (0, 1], t_u \in [0, 1)\ \forall u \neq l\}.$$

Thus every bin can be recursively sub-partitioned following (5.19). For Sierpinski binning with L labels, the number of bins at depth q is given by $(L^{q+1} - 1)/(L - 1)$.

5.G.2 Grid-style binning

Grid-style binning is motivated from the 2D reliability diagrams of Widmann et al. (2019, Figure 1), where they partitioned Δ^2 into multiple equi-volume bins in order to assess canonical calibration on a 3-class version of CIFAR-10. For $L = 3$, Δ^2 can be divided as shown in Figure 5.11. This corresponds to *gridding* the space Δ^2, just the way we think of *gridding* the real hyperplane. However, the mathematical description of this grid for general L is not apparent from Figure 5.11. We describe grid-style binning formally for general $L \geqslant 3$.

Consider some $\tau > 0$ such that $K := 1/\tau \in \mathbb{N}$. For every tuple $\mathbf{k} = (k_1, k_2, \ldots, k_L)$ in the set

$$I = \{\mathbf{k} \in \mathbb{N}^L : \max(L, K + 1) \leqslant \sum_{l \in [L]} k_l \leqslant K + (L - 1)\}, \tag{5.20}$$

define the bins

$$b_{\mathbf{k}} := \{\mathbf{s} \in \Delta^{L-1} : \text{ for every } l \in [L], s_l K \in [k_l - 1, k_l]. \tag{5.21}$$

These bins are not mutually disjoint, but intersections can only occur at the edges. That is, for every $\mathbf{s}$ that belongs to more than one bin, at least one component s_l satisies $s_l K \in \mathbb{N}$. In order to identify a single bin s, ties can be broken arbitrarily. One strategy is to use some ordering on $\mathbb{N}^L$; say for $\mathbf{k}_1 \neq \mathbf{k}_2 \in \mathbb{N}^L$, $\mathbf{k}_1 < \mathbf{k}_2$ if and only if for the first element of $\mathbf{k}_1$ that is unequal to the corresponding element of $\mathbf{k}_2$ the one corresponding to $\mathbf{k}_1$ is smaller. Then define the binning function $\mathcal{B} : \Delta^{L-1} \to |I|$ as $\mathcal{B}(\mathbf{s}) = \min\{\mathbf{k} : \mathbf{s} \in b_{\mathbf{k}}\}$. The following propositions prove that a) each $\mathbf{s}$ belongs to at least one bin, and b) that every bin is an $L-1$ dimensional object (and thus a meaningful partition of Δ^{L-1}).

Proposition 5.2. *The bins* $\{b_{\mathbf{k}} : \mathbf{k} \in I\}$ *defined by* (5.21) *mutually exhaust* Δ^{L-1}.

Proof. Consider any $\mathbf{s} \in \Delta^{L-1}$. For $s_l K \notin \mathbb{N} = \{1, 2, \ldots\}$, set $k_l = \max(1, \lceil s_l K \rceil) > s_l K$. Consider the condition

$$C: \text{ for all } l \text{ such that } s_l K \notin \mathbb{N}, s_l K = 0.$$

If C is true, then for l such that $s_l K \in \mathbb{N}$, set $k_l = s_l K$. If C is not true, then for l such that $s_l K \in \mathbb{N}$, set exactly one $k_l = s_l K + 1$, and for the rest set $k_l = s_l K$. Based on this setting of $\mathbf{k}$, it can be verified that $\mathbf{s} \in b_{\mathbf{k}}$.

Further, note that for every l, $k_l \geqslant s_l K$, and there exists at least one l such that $k_l > s_l K$. Thus we have:

$$\begin{aligned}\sum_{l=1}^{L} k_l &> \sum_{l=1}^{L} s_l K \\ &= K.\end{aligned}$$

Since $\sum_{l=1}^{L} k_l \in \mathbb{N}$, we must have $\sum_{l=1}^{L} k_l \geqslant K + 1$. Further since each $k_l \in \mathbb{N}$, $\sum_{l=1}^{L} k_l \geqslant L$.

Next, note that for every l, $k_l \leqslant s_l K + 1$. If C is true, then there is at least one l such that $s_l K \in \mathbb{N}$, and for this l, we have set $k_l = s_l K < s_l K + 1$. If C is not true, then either there exists at least one l such that $s_l K \notin \mathbb{N} \cup \{0\}$ for which $k_l = \lceil s_l K \rceil < s_l K + 1$, or every $s_l K \in \mathbb{N}$, in which case we have set $k_l = s_l K$ for one of them. In all cases, note that there exists an l such that $k_l < s_l K + 1$. Thus,

$$\begin{aligned}\sum_{l=1}^{L} k_l &< \sum_{l=1}^{L} (s_l K + 1) \\ &= K + L.\end{aligned}$$

Since $\sum_{l=1}^{L} k_l \in \mathbb{N}$, we must have $\sum_{l=1}^{L} k_l \leqslant K + L - 1$.

□

Next, we show that each bin indexed by $\mathbf{k} \in I$ contains a non-zero volume subset of Δ^{L-1}, where volume is defined with respect to the Lebesgue measure in $\mathbb{R}^{L-1}$. This can be shown by arguing that $b_{\mathbf{k}}$ contains a scaled and translated version of Δ^{L-1}.

Proposition 5.3. *For every* $\mathbf{k} \in I$*, there exists some* $\mathbf{u} \in \mathbb{R}^L$ *and* $v > 0$ *such that* $\mathbf{u} + v\Delta^{L-1} \subseteq b_{\mathbf{k}}$.

Proof. Fix some $\mathbf{k} \in I$. By condition (5.20), $\sum_{l=1}^{L} k_l \in [\max(L, K+1), K+L-1]$. Based on this, we claim that there exists a $\tau \in [0, 1)$ such that

$$\sum_{l=1}^{L}(k_l - 1) + \tau L + (1-\tau) = K. \tag{5.22}$$

Indeed, note that for $\tau = 0$, $\sum_{l=1}^{L}(k_l - 1) + \tau L + (1-\tau) \leqslant (K-1)+1 = K$ and for $\tau = 1$, $\sum_{l=1}^{L}(k_l - 1) + \tau L + (1-\tau) = \sum_{l=1}^{L} k_l > K$. Thus, there exists a τ that satisfies (5.22) by the intermediate value theorem.

Next, define $\mathbf{u} = K^{-1}(\mathbf{k} + (\tau - 1)\mathbf{1}_L)$ and $v = K^{-1}(1-\tau) > 0$, where $\mathbf{1}_L$ denotes the vector in $\mathbb{R}^L$ with each component equal to 1. Consider any $\mathbf{s} \in \mathbf{u} + v\Delta^{L-1}$. Note that for every $l \in [L]$, $s_l K \in [k_l - 1, k_l]$ and by property (5.22),

$$\sum_{l=1}^{L} s_l K = \left(\sum_{l=1}^{L}(k_l + (\tau - 1))\right) + v = \sum_{l=1}^{L}(k_l - 1) + \tau L + (1-\tau) = K.$$

Thus, $\mathbf{s} \in \Delta^{L-1}$ and by the definition of $b_{\mathbf{k}}$, $\mathbf{s} \in b_{\mathbf{k}}$. This completes the proof.

□

The previous two propositions imply that $\mathcal{B}$ satisfies the property we require of a reasonable binning scheme. For $L = 3$, grid-style binning gives equi-volume bins as illustrated in Figure 5.11; however this is not true for $L > 3$. We now describe a histogram binning based partitioning scheme.

5.G.3 Projection based histogram binning for canonical calibration

Some of the bins defined by Sierpinski binning and grid-style binning may have very few calibration points n_b, leading to poor estimation of $\widehat{\Pi}$. In the binary calibration case, this can be remedied using histogram binning which strongly relies on the scoring function g taking values in a fully ordered space $[0, 1]$. To ensure that each bin contains $\Omega(n/B)$ points, we estimate the quantiles of $g(X)$ and created the bins as per these quantiles. However, there are no natural quantiles for unordered prediction spaces such as Δ^{L-1} ($L \geqslant 3$). In this section, we develop a histogram binning scheme for Δ^{L-1} that is semantically interpretable and has desirable statistical properties.

Our algorithm takes as input a prescribed number of bins B and an arbitrary sequence of vectors $q_1, q_2, \ldots, q_{B-1} \in \mathbb{R}^L$ with unit ℓ_2-norm: $\|q_i\|_2 = 1$. Each of these vectors represents a direction on which we will project Δ^{L-1} in order to induce a full order on Δ^{L-1}. Then, for each direction, we will use an order statistics on the induced full order to identify a bin with exactly $\lfloor (n+1)/B \rfloor - 1$ calibration points (except the last bin, which may have more points). The formal algorithm is described in Algorithm 5.10. It uses the following new notation: given m vectors $v_1, v_2, \ldots, v_m \in \mathbb{R}^L$, a unit vector u, and an index $j \in [m]$, let order-statistics$(\{v_1, v_2, \ldots, v_m\}, u, j)$ denote the j-th order-statistics of $\{v_1^T u, v_2^T u, \ldots, v_m^T u\}$.

Algorithm 5.10 Projection histogram binning for canonical calibration

```
Require: Base multiclass predictor $\mathbf{g} : \mathcal{X} \to \Delta^{L-1}$, calibration data $\mathcal{D} = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$
Ensure: Approximately calibrated scoring function $\mathbf{h}$
  $S \leftarrow \{\mathbf{g}(X_1), \mathbf{g}(X_2), \dots, \mathbf{g}(X_n)\}$
  $T \leftarrow$ empty array of size $B$
  $c \leftarrow \lfloor \frac{n+1}{B} \rfloor$
  for $b \leftarrow 1$ to $B - 1$ do
      $T_b \leftarrow$ order-statistics$(S, q_b, c)$
      $S \leftarrow S \backslash \{v \in S : v^T q_b \leqslant T_b\}$
  end for
  $T_B \leftarrow 1.01$
  $\mathcal{B}(\mathbf{g}(\cdot)) \leftarrow \min\{b \in [B] : \mathbf{g}(\cdot)^T q_b < T_b\}$
  $\widehat{\Pi} \leftarrow$ empty matrix of size $B \times L$
  for $b \leftarrow 1$ to $B$ do
      for $l \leftarrow 1$ to $L$ do
          $\widehat{\Pi}_{b,l} \leftarrow \text{Mean}\{\mathbb{1}\{Y_i = l\} : \mathcal{B}(\mathbf{g}(X_i)) = b \text{ and } \forall s \in [B],\ \mathbf{g}(X_i)^T q_s \neq T_s\}$
      end for
  end for
  for $l \leftarrow 1$ to $L$ do
      $h_l(\cdot) \leftarrow \widehat{\Pi}_{\mathcal{B}(\mathbf{g}(\cdot)),l}$
  end for
  return $\mathbf{h}$
```

We now briefly describe some values computed by Algorithm 5.10 in words to build intuition. The array T, which is learnt on the data, represents the identified thresholds for the directions given by q. Each (q_b, T_b) pair corresponds to a hyperplane that *cuts* Δ^{L-1} into two subsets given by $\{x \in \Delta^{L-1} : x^T q_b < T_b\}$ and $\{x \in \Delta^{L-1} : x^T q_b \geqslant T_b\}$. The overall partitioning of Δ^{L-1} is created by merging these cuts sequentially. This defines the binning function $\mathcal{B}$. By construction, the binning function is such that each bin contains at least $\lfloor \frac{n+1}{B} \rfloor - 1$ many points in its interior. As suggested by Gupta and Ramdas (2021), we do not include the points that lie on the boundary, that is, points X_i that satisfy $\mathbf{g}(X_i)^T q_s = T_s$ for some $s \in [B]$. The interior points bins are then used to estimate the bin biases $\widehat{\Pi}$.

No matter how the q-vectors are chosen, the bins created by Algorithm 5.10 have at least $\lfloor \frac{n}{B} \rfloor - 1$ points for bias estimation. However, we discuss some simple heuristics for setting q that are semantically meaningful. For some intuition, note that the binary version of histogram binning Gupta and Ramdas (2021, Algorithm 1) is essentially recovered by Algorithm 5.10 if $L = 2$ by setting each q_b as $\mathbf{e}_2$ (the vector $[0, 1]$). Equivalently, we can set each q_b to $-\mathbf{e}_1$ since both are equivalent for creating a projection-based order on Δ_2. Thus for $L \geqslant 3$, a natural strategy for the q-vectors is to rotate between the canonical basis vectors: $q_1 = -\mathbf{e}_1, q_2 = -\mathbf{e}_2, \dots, q_L = -\mathbf{e}_L, q_{L+1} = -\mathbf{e}_1, \dots,$ and so on. Projecting with respect to $-\mathbf{e}_l$ focuses on the class l by forming a bin corresponding to the largest values of $g_l(X_i)$ among the remaining X_i's which have not yet been binned. (On the other hand, projecting with respect to $\mathbf{e}_l$ will correspond to forming a bin with the smallest values of $g_l(X_i)$.)

The q-vectors can also be set adaptively based on the training data (without seeing the calibration data). For instance, if most points belong to a specific class $l \in [L]$, we may want more sharpness for this particular class. In that case, we can choose a higher ratio of the q-vectors to be $-\mathbf{e}_l$.

5.G.4 Experiments with the COVTYPE dataset

In Figure 5.12 we illustrate the binning schemes proposed in this section on a 3-class version of the COVTYPE-7 dataset considered in Section 5.B.2. As noted before, this is an imbalanced dataset where classes 1 and 2 dominate. We created a 3 class problem with the classes 1, 2, and other (as class 3). The entire dataset has 581012 points and the ratio of the classes is approximately 36%, 49%, 15% respectively. The dataset was split into train-test in the ratio 70:30. The training data was further split into modeling-calibration in the ratio 90:10. A logistic regression model $\mathbf{g}$ using `sklearn.linear_model.LogisticRegression` was learnt on the modeling data, and $\mathbf{g}$ was recalibrated on the calibration data.

The plots on the right in Figure 5.12 are *recalibration diagrams*. The base predictions $\mathbf{g}(X)$ on the test-data are displayed as a scatter plot on Δ^2. Points in different bins are colored using one of 10 different colors (since the number of bins is larger than 10, some colors correspond to more than one bin). For each bin, an arrow is drawn, where the tail of the arrow corresponds to the average $\mathbf{g}(X)$ predictions in the bin and the head of the arrow corresponds to the recalibrated $\mathbf{h}(X)$ prediction for the bin. For bins that contained very few points, the arrows are suppressed for visual clarity.

The plots on the left in Figure 5.12 are validity plots (Gupta and Ramdas, 2021). Validity plots display estimates of

$$V(\epsilon) = \mathbb{P}_{\text{test-data}} \left(\left\lVert \mathbb{E}\left[\mathbf{Y} \mid \mathbf{g}(X) \right] - \mathbf{g}(X) \right\rVert_1 \leqslant \epsilon \right),$$

as ϵ varies ($\mathbf{g}$ corresponds to the validity plot for logistic regression; replacing $\mathbf{g}$ with $\mathbf{h}$ above gives plots for the binning based classifier $\mathbf{h}$). For logistic regression, the same binning scheme as the one provided by $\mathbf{h}$ is used to estimate $V(\epsilon)$.

Overall, Figure 5.12 shows that all of the binning approaches improve the calibration of the original logistic regression model across different ϵ. However, the recalibration does not change the original model significantly. Comparing the different binning methods to each other, we find that they all perform quite similarly. It would be interesting to further study these and other binning methods for post-hoc canonical calibration.

5.H Proofs

Proofs appear in separate subsections, in the same order as the corresponding results appear in the main chapter and Appendix. Proposition 5.4 was stated informally, so we state it formally as well.

5.H.1 Statement and proof of Proposition 5.4

Proposition 5.4. *For any predictor* (c, h)*, conf-ECE*$(c, h) \leqslant$ *TL-ECE*(c, h)*.*

Proof. To avoid confusion between the the conditioning operator and the absolute value operator $|\cdot|$, we use $\text{abs}\,(\cdot)$ to denote absolute values below. Note that,

$$\begin{aligned}
\text{abs}\left(\mathbb{P}(Y = c(X) \mid h(X)) - h(X)\right) &= \text{abs}\left(\mathbb{E}\left[\mathbb{1}\left\{Y = c(X)\right\} \mid h(X)\right] - h(X)\right) \\
&= \text{abs}\left(\mathbb{E}\left[\mathbb{1}\left\{Y = c(X)\right\} - h(X) \mid h(X)\right]\right) \\
&= \text{abs}\left(\mathbb{E}\left[\mathbb{E}\left[\mathbb{1}\left\{Y = c(X)\right\} - h(X) \mid h(X), c(X)\right] \mid h(X)\right]\right) \\
&\leqslant \mathbb{E}\left[\text{abs}\left(\mathbb{E}\left[\mathbb{1}\left\{Y = c(X)\right\} - h(X) \mid h(X), c(X)\right]\right) \mid h(X)\right] \\
&\qquad\qquad \text{(by Jensen's inequality)} \\
&= \mathbb{E}\left[\text{abs}\left(\mathbb{P}(Y = c(X) \mid h(X), c(X)) - h(X)\right) \mid h(X)\right].
\end{aligned}$$

Thus,

$$\begin{aligned}
\text{conf-ECE}(c, h) &= \mathbb{E}\left[\text{abs}\left(\mathbb{P}(Y = c(X) \mid h(X)) - h(X)\right)\right] \\
&\leqslant \mathbb{E}\left[\mathbb{E}\left[\text{abs}\left(\mathbb{P}(Y = c(X) \mid h(X), c(X)) - h(X)\right) \mid h(X)\right]\right] \\
&= \mathbb{E}\left[\text{abs}\left(\mathbb{P}(Y = c(X) \mid h(X), c(X)) - h(X)\right)\right] \\
&= \text{TL-ECE}(c, h).
\end{aligned}$$

□

5.H.2 Proof of Theorem 5.1

The proof strategy is as follows. First, we use the bound of Gupta and Ramdas (2021, Theorem 4) (henceforth called the GR21 bound), derived in the binary calibration setting, to conclude marginal, conditional, and ECE guarantees for each h_l. Then, we show that the binary guarantees for the individual h_l's leads to a top-label guarantee for the overall predictor (c, h).

Consider any $l \in [L]$. Let P_l denote the conditional distribution of $(X, \mathbb{1}\left\{Y = l\right\})$ given $c(X) = l$. Clearly, D_l is a set of n_l i.i.d. samples from P_l, and h_l is learning a binary calibrator with respect to P_l using binary histogram binning. The number of data-points is n_l and the number of bins is $B_l = \lfloor n_l/k \rfloor$ bins. We now apply the GR21 bounds on h_l. First, we verify that the condition they require is satisfied:

$$n_l \geqslant k \left\lfloor n_l/k \right\rfloor \geqslant 2B_l.$$

Thus their marginal calibration bound for h_l gives,

$$P\left(|P(Y = l \mid c(X) = l, h_l(X)) - h_l(X)| \leqslant \delta + \sqrt{\frac{\log(2/\alpha)}{2(\lfloor n_l/B_l \rfloor - 1)}} \mid c(X) = l\right) \geqslant 1 - \alpha.$$

Note that since $\lfloor n_l/B_l \rfloor = \lfloor n_l/\lfloor n_l/k \rfloor \rfloor \geqslant k$,

$$\epsilon_1 = \delta + \sqrt{\frac{\log(2/\alpha)}{2(k-1)}} \geqslant \delta + \sqrt{\frac{\log(2/\alpha)}{2(\lfloor n_l/B_l \rfloor - 1)}}.$$

Thus we have

$$P\left(\left|P(Y = l \mid c(X) = l, h_l(X)) - h_l(X)\right| \leqslant \epsilon_1 \mid c(X) = l\right) \geqslant 1 - \alpha.$$

This is satisfied for every l. Using the law of total probability gives us the top-label marginal calibration guarantee for (c, h):

$$\begin{aligned}
&P(|P(Y = c(X) \mid c(X), h(X)) - h(X)| \leqslant \epsilon_1) \\
&= \sum_{l=1}^{L} P(c(X) = l) P(|P(Y = c(X) \mid c(X), h(X)) - h(X)| \leqslant \epsilon_1 \mid c(X) = l) \\
&\qquad\qquad \text{(law of total probability)} \\
&= \sum_{l=1}^{L} P(c(X) = l) P(|P(Y = l \mid c(X) = l, h_l(X)) - h_l(X)| \leqslant \epsilon_1 \mid c(X) = l) \\
&\qquad\qquad \text{(by construction, if } c(x) = l,\ h(x) = h_l(x)\text{)} \\
&\geqslant \sum_{l=1}^{L} P(c(X) = l)(1 - \alpha) \\
&= 1 - \alpha.
\end{aligned}$$

Similarly, the in-expectation ECE bound of GR21, for $p = 1$, gives for every l,

$$\begin{aligned}
\mathbb{E}\left|P(Y = l \mid c(X) = l, h_l(X)) - h_l(X) \mid c(X) = l\right| &\leqslant \sqrt{B_l/2n_l} + \delta \\
&= \sqrt{\lfloor n_l/k \rfloor / 2n_l} + \delta \\
&\leqslant \sqrt{1/2k} + \delta.
\end{aligned}$$

Thus,

$$\begin{aligned}
&\mathbb{E}|P(Y = c(X) \mid c(X), h_l(X)) - h(X)| \\
&= \sum_{l=1}^{L} P(c(X) = l) \mathbb{E}|P(Y = l \mid c(X) = l, h_l(X)) - h_l(X)| \mid c(X) = l \\
&\leqslant \sum_{l=1}^{L} P(c(X) = l)(\sqrt{1/2k} + \delta) \\
&= \sqrt{1/2k} + \delta.
\end{aligned}$$

Next, we show the top-label conditional calibration bound. Let $B = \sum_{l=1}^{L} B_l$ and $\alpha_l = \alpha B_l / B$. Note that $B \leqslant \sum_{l=1}^{L} n_l/k = n/k$. The binary conditional calibration bound of GR21 gives

$$P\left(\forall r \in \text{Range}(h_l), |P(Y = l \mid c(X) = l, h_l(X) = r) - r| \leqslant \delta + \sqrt{\frac{\log(2B_l/\alpha_l)}{2(\lfloor n_l/B_l \rfloor - 1)}} \mid c(X) = l\right) \geqslant 1 - \alpha_l.$$

Note that

$$\begin{aligned}\sqrt{\frac{\log(2B_l/\alpha_l)}{2(\lfloor n_l/B_l \rfloor - 1)}} &= \sqrt{\frac{\log(2B/\alpha)}{2(\lfloor n_l/B_l \rfloor - 1)}} \\ &\leqslant \sqrt{\frac{\log(2n/k\alpha)}{2(\lfloor n_l/B_l \rfloor - 1)}} && \text{(since } B \leqslant n/k\text{)} \\ &\leqslant \sqrt{\frac{\log(2n/k\alpha)}{2(k-1)}} && \text{(since } k \leqslant \lfloor n_l/B_l \rfloor\text{).}\end{aligned}$$

Thus for every $l \in [L]$,

$$P(\forall r \in \text{Range}(h_l)),\ |P(Y = l \mid c(X) = l, h_l(X) = r) - r| \leqslant \epsilon_2) \geqslant 1 - \alpha_l.$$

By construction of h, conditioning on $c(X) = l$ and $h_l(X) = r$ is the same as conditioning on $c(X) = l$ and $h(X) = r$. Taking a union bound over all L gives

$$\begin{aligned}P(\forall l \in [L], r \in \text{Range}(h), |P(Y = c(X) \mid c(X) = l, h(X) = r) - r|) \leqslant \epsilon_2) \\ \geqslant 1 - \sum_{l=1}^{L} \alpha_l = 1 - \alpha,\end{aligned}$$

proving the conditional calibration result. Finally, note that if for every $l \in [L], r \in \text{Range}(h)$,

$$|P(Y = c(X) \mid c(X) = l, h(X) = r) - r| \leqslant \epsilon_2,$$

then also

$$\text{TL-ECE}(c, h) = \mathbb{E}|\mathbb{P}(Y = c(X) \mid h(X), c(X)) - h(X)| \leqslant \epsilon_2.$$

This proves the high-probability bound for the TL-ECE. □

Remark 5.3. Gupta and Ramdas (2021) proved a more general result for general ℓ_p-ECE bounds. Similar results can also be derived for the suitably defined ℓ_p-TL-ECE. Additionally, it can be shown that with probability $1 - \alpha$, the TL-MCE of (c, h) is bounded by ϵ_2. (TL-MCE is defined in Appendix 5.E, equation (5.17).)

5.H.3 Proof of Proposition 5.1

Consider a specific $l \in [L]$. We use h_l to denote the l-th component function of $\mathbf{h}$ and $Y_l = \mathbb{1}\{Y = l\}$. Canonical calibration implies

$$\mathbb{P}(Y = l \mid \mathbf{h}(X)) = \mathbb{E}[Y_l \mid \mathbf{h}(X)] = h_l(X). \tag{5.23}$$

We can then use the law of iterated expectations (or tower rule) to get the final result:

$$\begin{aligned}
\mathbb{E}\left[Y_l \mid h_l(X)\right] &= \mathbb{E}\left[\mathbb{E}\left[Y_l \mid \mathbf{h}(X)\right] \mid h_l(X)\right] \\
&= \mathbb{E}\left[h_l(X) \mid h_l(X)\right] && \text{(by the canonical calibration property (5.23))} \\
&= h_l(X).
\end{aligned}$$

□

5.H.4 Proof of Theorem 5.2

We use the bounds of Gupta and Ramdas (2021, Theorem 4) (henceforth called the GR21 bounds), derived in the binary calibration setting, to conclude marginal, conditional, and ECE guarantees for each h_l. This leads to the class-wise results as well.

Consider any $l \in [L]$. Let P_l denote the distribution of $(X, \mathbb{1}\{Y = l\})$. Clearly, D_l is a set of n i.i.d. samples from P_l, and h_l is learning a binary calibrator with respect to P_l using binary histogram binning. The number of data-points is n and the number of bins is $B_l = \lfloor n/k_l \rfloor$ bins. We now apply the GR21 bounds on h_l. First, we verify that the condition they require is satisfied:

$$n \geqslant k_l \lfloor n/k_l \rfloor \geqslant 2B_l.$$

Thus the GR21 marginal calibration bound gives that for every $l \in [L]$, h_l satisfies

$$P\left(\left|P(Y = l \mid h_l(X)) - h_l(X)\right| \leqslant \delta + \sqrt{\frac{\log(2/\alpha_l)}{2(\lfloor n/B_l \rfloor - 1)}}\right) \geqslant 1 - \alpha_l.$$

The class-wise marginal calibration bound of Theorem 5.2 follows since $\lfloor n/B_l \rfloor = \lfloor n/\lfloor n/k_l \rfloor \rfloor \geqslant k_l$, and so

$$\epsilon_l^{(1)} \geqslant \delta + \sqrt{\frac{\log(2/\alpha_l)}{2(\lfloor n/B_l \rfloor - 1)}}.$$

Next, the GR21 conditional calibration bound gives for every $l \in [L]$, h_l satisfies

$$P\left(\forall r \in \text{Range}(h_l), \left|P(Y = l \mid h_l(X) = r) - r\right| \leqslant \delta + \sqrt{\frac{\log(2B_l/\alpha_l)}{2(\lfloor n/B_l \rfloor - 1)}}\right) \geqslant 1 - \alpha_l.$$

The class-wise marginal calibration bound of Theorem 5.2 follows since $B_l = \lfloor n/k_l \rfloor \leqslant n/k_l$ and $\lfloor n/B_l \rfloor = \lfloor n/\lfloor n/k_l \rfloor \rfloor \geqslant k_l$, and so

$$\epsilon_l^{(2)} \geqslant \delta + \sqrt{\frac{\log(2B_l/\alpha_l)}{2(\lfloor n/B_l \rfloor - 1)}}.$$

Let $k = \min_{l \in [L]} k_l$. The in-expectation ECE bound of GR21, for $p = 1$, gives for every l,

$$\mathbb{E}\left[\text{binary-ECE-for-class-}l\ (h_l)\right] \leqslant \sqrt{B_l/2n_l} + \delta$$

$$
\begin{aligned}
&= \sqrt{\lfloor n/k_l \rfloor /2n} + \delta \\
&\leqslant \sqrt{1/2k_l} + \delta \\
&\leqslant \sqrt{1/2k} + \delta.
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\mathbb{E}\left[\text{CW-ECE}(c,h)\right] &= \mathbb{E}\left[L^{-1}\sum_{l=1}^{L}\text{binary-ECE-for-class-}l\ (h_l)\right] \\
&\leqslant L^{-1}\sum_{l=1}^{L}(\sqrt{1/2k}+\delta) \\
&= \sqrt{1/2k}+\delta,
\end{aligned}
$$

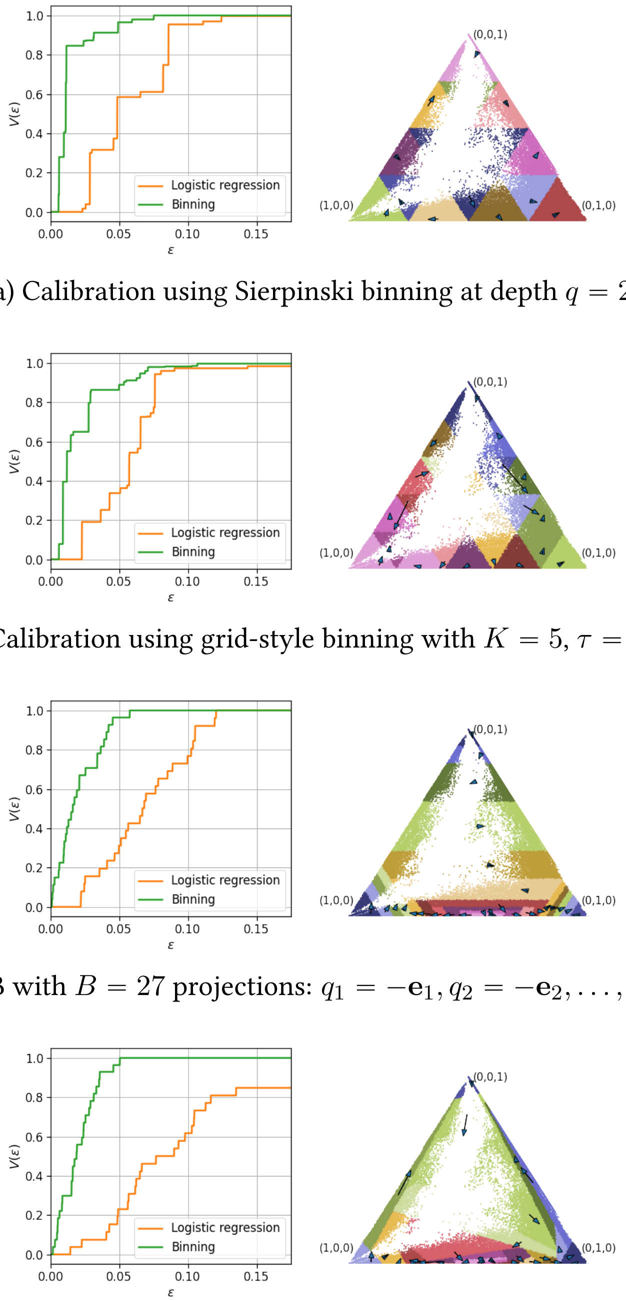
as required for the in-expectation CW-ECE bound of Theorem 5.2. Finally, for the high probability CW-ECE bound, let $\epsilon = \max_{l\in[L]} \epsilon_l^{(2)}$ and $\alpha = \sum_{l=1}^{L} \alpha_l$. By taking a union bound over the the conditional calibration bounds for each h_l, we have, with probability $1-\alpha$, for every $l \in [L]$ and $r \in \text{Range}(h)$,

$$
|P(Y = l \mid c(X) = l, h(X) = r) - r| \leqslant \epsilon_l^{(2)} \leqslant \epsilon.
$$

Thus, with probability $1-\alpha$,

$$
\text{CW-ECE}(c,h) = L^{-1}\sum_{l=1}^{L}\mathbb{E}|\mathbb{P}(Y = l \mid h_l(X)) - h_l(X)| \leqslant \epsilon.
$$

This proves the high-probability bound for the CW-ECE. □



(a) Calibration using Sierpinski binning at depth $q = 2$.

(b) Calibration using grid-style binning with $K = 5$, $\tau = 0.2$.

(c) Projection-based HB with $B = 27$ projections: $q_1 = -\mathbf{e}_1, q_2 = -\mathbf{e}_2, \ldots, q_4, -\mathbf{e}_1, \ldots$, and so on.

(d) Projection-based HB with $B = 27$ random projections (q_i drawn uniformly from the ℓ_2-unit-ball in $\mathbb{R}^3$).

Figure 5.12: Canonical calibration using fixed and histogram binning on a 3-class version of the COVTYPE-7 dataset. The reliability diagrams (left) indicate that all forms of binning improve the calibration of the base logistic regression model. The recalibration diagrams (right) are a scatter plot of the predictions $\mathbf{g}(X)$ on the test data with the points colored in 10 different colors depending on their bin. For every bin, the arrow-tail indicates the mean probability predicted by the base model $\mathbf{g}$ whereas the arrow-head indicates the probability predicted by the updated model $\mathbf{h}$.

Chapter 6

Online Platt scaling with calibeating

This chapter is based on Gupta and Ramdas (2023).

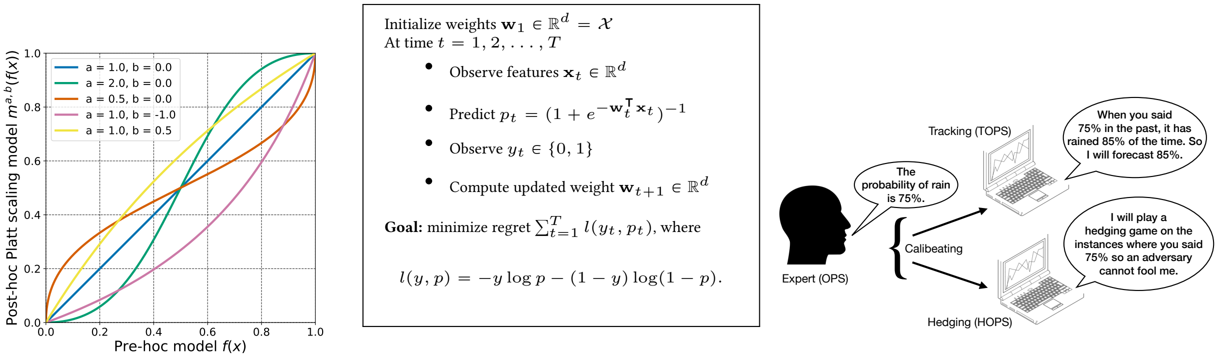
We present an online post-hoc calibration method, called Online Platt Scaling (OPS), which combines the Platt scaling technique with online logistic regression. We demonstrate that OPS smoothly adapts between i.i.d. and non-i.i.d. settings with distribution drift. Further, in scenarios where the best Platt scaling model is itself miscalibrated, we enhance OPS by incorporating a recently developed technique called calibeating to make it more robust. Theoretically, our resulting OPS+calibeating method is guaranteed to be calibrated for adversarial outcome sequences. Empirically, it is effective on a range of synthetic and real-world datasets, with and without distribution drifts, achieving superior performance without hyperparameter tuning. Finally, we extend all OPS ideas to the beta scaling method.

6.1 Introduction

In the past two decades, there has been significant interest in the ML community on post-hoc calibration of ML classifiers (Zadrozny and Elkan, 2002; Niculescu-Mizil and Caruana, 2005; Guo et al., 2017). Consider a pretrained classifier $f : \mathcal{X} \to [0,1]$ that produces scores in $[0,1]$ for covariates in $\mathcal{X}$. Suppose f is used to make probabilistic predictions for a sequence of points $(\mathbf{x}_t, y_t)_{t=1}^T$ where $y_t \in \{0,1\}$. Informally, f is said to be calibrated (Dawid, 1982) if the predictions made by f match the empirically observed frequencies when those predictions are made:

$$\text{for all } p \in [0,1], \text{Average}\{y_t : f(\mathbf{x}_t) \approx p\} \approx p. \tag{6.1}$$

In practice, for well-trained f, larger scores $f(\mathbf{x})$ indicate higher likelihoods of $y = 1$, so that f does well for accuracy or a ranking score like AUROC. Yet we often find that f does not satisfy (some formalized version of) condition (6.1). The goal of post-hoc calibration, or recalibration, is to use additional held-out data to learn a *low-complexity* mapping $m : [0,1] \to [0,1]$ so that $m(f(\cdot))$ retains the good properties of f—accuracy, AUROC, sharpness—as much as possible, but is better calibrated than f.


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Post-hoc Platt scaling model $m^{a,b}(f(x))$
Pre-hoc model $f(x)$
a = 1.0, b = 0.0
a = 2.0, b = 0.0
a = 0.5, b = 0.0
a = 1.0, b = -1.0
a = 1.0, b = 0.5
The probability of rain is 75%.
Expert (OPS)
Calibeating
Tracking (TOPS)
Hedging (HOPS)
When you said 75% in the past, it has rained 85% of the time. So I will forecast 85%.
I will play a hedging game on the instances where you said 75% so an adversary cannot fool me.

```
Initialize weights $\mathbf{w}_1 \in \mathbb{R}^d = \mathcal{X}$
At time $t = 1, 2, \ldots, T$
    • Observe features $\mathbf{x}_t \in \mathbb{R}^d$
    • Predict $p_t = (1 + e^{-\mathbf{w}_t^{\mathsf{T}} \mathbf{x}_t})^{-1}$
    • Observe $y_t \in \{0, 1\}$
    • Compute updated weight $\mathbf{w}_{t+1} \in \mathbb{R}^d$
Goal: minimize regret $\sum_{t=1}^{T} l(y_t, p_t)$, where
    $l(y, p) = -y \log p - (1 - y) \log(1 - p).$
```

(a) Platt scaling (b) Online logistic regression (c) Calibeating + online Platt scaling

Figure 6.1: The combination of Platt scaling and online logistic regression yields Online Platt Scaling (OPS). Calibeating is applied on top of OPS to achieve further empirical improvements and theoretical validity.

The focus of this work is on a recalibration method proposed by Platt (1999), commonly known as Platt scaling (PS). The PS mapping m is a sigmoid transform over f parameterized by two scalars $(a, b) \in \mathbb{R}^2$:

$$m^{a,b}(f(\mathbf{x})) := \text{sigmoid}(a \cdot \text{logit}(f(\mathbf{x})) + b). \tag{6.2}$$

This set of mappings includes the identity mapping $m^{1,0}$ that recovers f. Figure 6.1a has additional illustrative $m^{a,b}$ plots; these are easily interpreted—if f is overconfident, that is if $f(x)$ values are skewed towards 0 or 1, we can pick $a \in (0, 1)$ to improve calibration; if f is underconfident, we can pick $a > 1$; if f is systematically biased towards 0 (or 1), we can pick $b > 0$ (or $b < 0$). The counter-intuitive choice $a < 0$ can also make sense if f's predictions oppose reality (perhaps due to a distribution shift). Given a batch of held-out data points, (a, b) is usually learnt by minimizing log-loss over calibration data or equivalently maximizing log-likelihood under the model $y_i \overset{\text{iid}}{\sim} \text{Bernoulli}(m^{a,b}(f(\mathbf{x}_i)))$.

Although a myriad of recalibration methods now exist, PS remains an empirically strong baseline. In particular, PS is effective when few samples are available for recalibration (Niculescu-Mizil and Caruana, 2005; Gupta and Ramdas, 2021). Scaling before subsequent binning has emerged as a useful methodology (Kumar et al., 2019; Zhang et al., 2020). Multiclass adaptations of PS, called temperature, vector, and matrix scaling have become popular (Guo et al., 2017). Being a parametric method, however, PS comprises a limited family of post-hoc corrections—for instance, since $m^{a,b}$ is always a monotonic transform, PS must fail even for i.i.d. data for some data-generating distributions (see Gupta et al. (2020) for a formal proof). Furthermore, we are interested in going beyond i.i.d. data to data with drifting/shifting distribution. This brings us to our first question,

(Q1) Can Platt scaling (PS) be extended to handle
shifting or drifting data distributions?

A separate view of calibration that pre-dates the ML post-hoc calibration literature is the online adversarial calibration framework (DeGroot and Fienberg, 1981; Foster and Vohra, 1998).

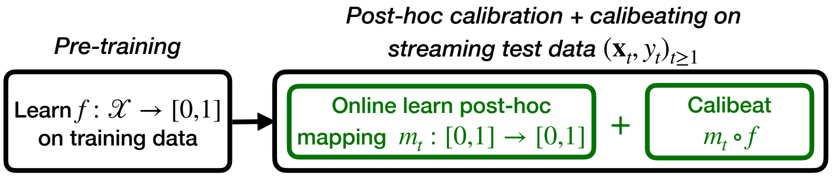

Pre-training
Learn $f : \mathcal{X} \to [0,1]$ on training data
Post-hoc calibration + calibeating on streaming test data $(\mathbf{x}_t, y_t)_{t\geq1}$
Online learn post-hoc mapping $m_t : [0,1] \to [0,1]$
+
Calibeat $m_t \circ f$

Figure 6.2: Online adversarial post-hoc calibration.

Through the latter work, we know that calibration can be achieved for arbitrary y_t sequences without relying on a pretrained model f or doing any other modeling over available features. This is achieved by hedging or randomizing over multiple probabilities, so that "the past track record can essentially only improve, no matter the future outcome" (paraphrased from Foster and Hart (2021)). For interesting classification problems, however, the y_t sequence is far from adversarial and informative covariates $\mathbf{x}_t$ are available. In such settings, covariate-agnostic algorithms achieve calibration by predicting something akin to an average $\sum_{s=1}^{t} y_s/t$ at time $t+1$ (see Appendix 6.D). Such a prediction, while calibrated, is arguably not useful. A natural question is:

(Q2) Can informative covariates (features) be used
to make online adversarial calibration practical?

We answer (Q1) by developing an online version of Platt scaling, and (Q2) by leveraging the recently developed framework of calibeating (Foster and Hart, 2023). The method of calibeating, illustrated in Figure 6.1c, is to perform certain *corrections* on top of pre-existing *expert* forecasts to improve their calibration. A key calibeating idea that we use was already discovered by Kuleshov and Ermon (2017) to resolve (Q2) in a manner similar to ours. Namely, they first proposed the idea of binning and hedging on top of an expert, as we do in HOPS (Section 6.3.3). We return to a more detailed comparison between our work and Kuleshov and Ermon's in Section 6.3.3. To reiterate, while we repeatedly use the term "calibeating" coined by Foster and Hart, the main idea in resolving (Q2) can equally be credited to Kuleshov and Ermon.

Unlike previous papers, the online expert is not a black-box but a centerpiece of our work. In the forthcoming proposal, we describe an end-to-end pipeline, where first, a covariate-based and time-adaptive expert is constructed using post-hoc calibration (OPS), and then it is calibeaten to achieve adversarial calibration (TOPS, HOPS).

6.1.1 Online adversarial post-hoc calibration

The proposal, summarized in Figure 6.2, is as follows. First, train any probabilistic classifier f on some part of the data. Then, perform *online post-hoc calibration* on top of f to get online adaptivity. In effect, this amounts to viewing $f(\mathbf{x}_t)$ as a scalar "summary" of $\mathbf{x}_t$, and the post-hoc mapping $(m_t : [0,1] \to [0,1])_{t\geqslant1}$ becomes the time-series model over the scalar feature $f(\mathbf{x}_t)$. Finally, apply calibeating on the post-hoc predictions $m_t(f(\mathbf{x}_t))$ to obtain adversarial validity.

Figure 6.2 highlights our choice to do both post-hoc calibration and calibeating simultaneously on the streaming test data $(\mathbf{x}_t, y_t)_{t\geqslant 1}$.

Such an online version of post-hoc calibration has not been previously studied to the best of our knowledge. We show how one would make PS online, to obtain Online Platt Scaling (OPS). OPS relies on a simple but crucial observation: PS is a two-dimensional logistic regression problem over "pseudo-features" $\text{logit}(f(\mathbf{x}_t))$. Thus the problem of learning OPS parameters is the problem of online logistic regression (OLR, see Figure 6.1b for a brief description). Several regret minimization algorithms have been developed for OLR (Hazan et al., 2007; Foster et al., 2018; Jézéquel et al., 2020). We consider these and find an algorithm with optimal regret guarantees that runs in linear time. These regret guarantees imply that OPS is guaranteed to perform as well as the best fixed PS model in hindsight for an arbitrarily distributed online stream $(\mathbf{x}_t, y_t)_{t\geqslant 1}$, which includes the entire range of distribution drifts—i.i.d. data, data with covariate/label drift, and adversarial data. We next present illustrative experiments where this theory bears out impressively in practice.

Then, Section 6.2 presents OPS, Section 6.3 discusses calibeating, Section 6.4 presents baseline experiments on synthetic and real-world datasets. Section 6.5 discusses the extension of all OPS ideas to a post-hoc technique called beta scaling.

6.1.2 Illustrative experiments with distribution drift

Covariate drift. We generated data as follows. For $t = 1, 2, \ldots, 6000$,

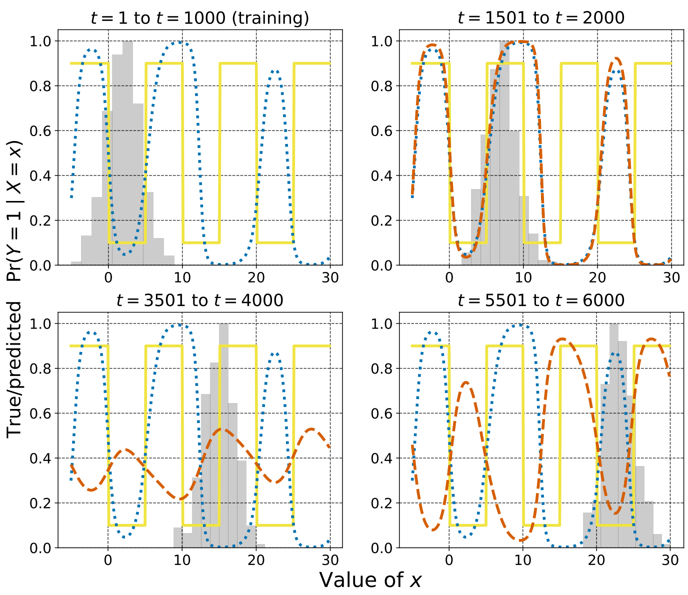
$$\begin{aligned} X_t &\sim \mathcal{N}((t-1)/250, 4); \\ Y_t | X_t &\sim \begin{cases} \text{Ber}(0.1) & \text{if } \text{mod}(\lfloor X_t/5 \rfloor, 2) = 0, \\ \text{Ber}(0.9) & \text{if } \text{mod}(\lfloor X_t/5 \rfloor, 2) = 1. \end{cases} \end{aligned} \tag{6.3}$$

Thus the distribution of Y_t given X_t is a fixed periodic function, but the distribution of X_t drifts over time. The solid yellow line in Figure 6.3 plots $\Pr(Y = 1 \mid X = x)$ against x. We featurized x as a 48-dimensional vector with the components $\sin\left(\frac{x}{\text{freq}} + \text{translation}\right)$, where $\text{freq} \in \{1, 2, 3, 4, 5, 6\}$ and $\text{translation} \in \{0, \pi/4, \pi/2, \ldots 7\pi/4\}$.

A logistic regression base model f is trained over this 48-dimensional representation using the points $(X_t, Y_t)_{t=1}^{1000}$, randomly permuted and treated as a single batch of exchangeable points, which we will call *training points*. The points $(X_t, Y_t)_{t=1001}^{6000}$ form a supervised non-exchangeable test stream: we use this stream to evaluate f, recalibrate f using OPS, and evaluate the OPS-calibrated model.

Figure 6.3 displays f and the recalibrated OPS models at four ranges of t (one per plot). The training data has most x_t-values in the range $[-5, 10]$ as shown by the (height-normalized) histogram in the top-left plot. In this regime, f is visually accurate and calibrated—the dotted light blue line is close to the solid yellow truth. We now make some observations at three test-time regimes of t:

(a) $t = 1501$ to $t = 2000$ (the histogram shows the distribution of $(x_t)_{t=1501}^{2000}$). For these values



t	Model	Acc ↑	CE ↓
1—1000	Base	88.00%	0.1
1501—2000	Base OPS	81.20% 81.68%	0.18 0.19
3501—4000	Base OPS	43.12% 62.16%	0.55 0.36
5501—6000	Base OPS	23.60% 87.76%	0.64 0.13

(a) Accuracy and calibration error (CE) values of the base model and OPS for indicated values of t.

Figure 6.3: The adaptive behavior of Online Platt scaling (OPS) for the covariate drift dataset described in Section 6.1.2. The title of each panel indicates the time-window that panel corresponds to. The histogram of X_t values in the corresponding time window is plotted with maximum height normalized to 1. Also plotted is the true curve for $\Pr(Y = 1 \mid X = x)$ and two predictive curves: a base model trained on $t = 1$ to $t = 1000$, and OPS-calibrated models with parameter values fixed at the start of the time window. The base model is accurate for the training data which is mostly in $[-5, 10]$, but becomes inaccurate and miscalibrated with the covariate-shifted values for larger t (bottom two subplots). OPS adapts well, agreeing with the base model in the top-right subplot, but flipping the base model predictions in the bottom-right subplot.

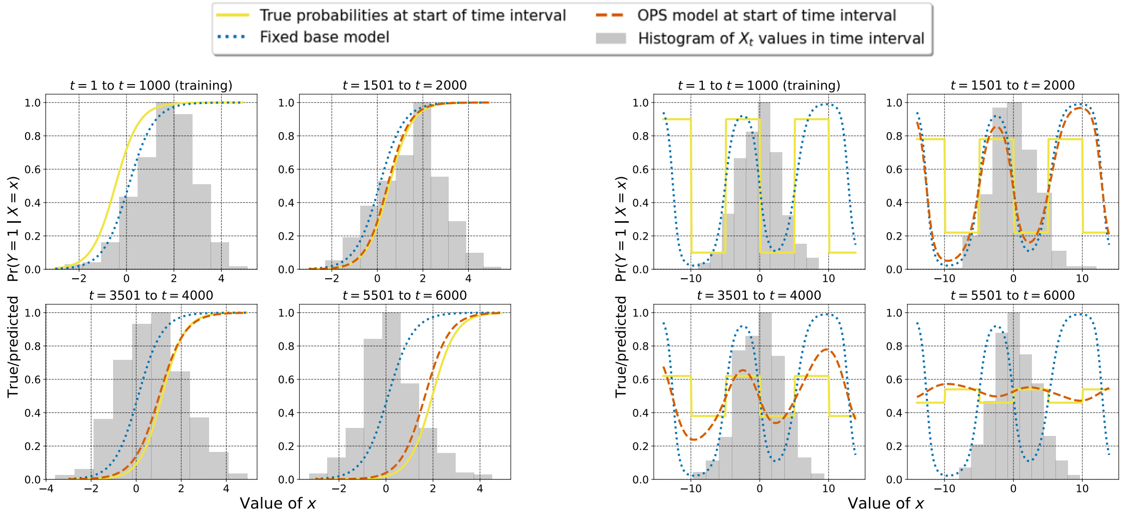
of t, the test data is only slightly shifted from the training data, and f continues to perform well. The OPS model recognizes the good performance of f and does not modify it much.

(b) $t = 3500$ to $t = 4000$. Here, f is "out-of-phase" with the true distribution, and Platt scaling is insufficient to improve f by a lot. OPS recognizes this, and it offers slightly better calibration and accuracy by making less confident predictions between 0.2 and 0.4.

(c) $t = 5500$ to $t = 6000$. In this regime, f makes predictions opposing reality. Here, the OPS model flips the prediction, achieving high accuracy and calibration.

These observations are quantitatively supported by the accuracy and ℓ_1-calibration error (CE) values reported by the table in Figure 6.3a. Accuracy and CE values are estimated using the known true distribution of $Y_t \mid X_t$ and the observed X_t values, making them unbiased and avoiding some well-known issues with CE estimation. More details are provided in Appendix 6.A.2.

Label drift. For $t = 1, 2, \ldots, 6000$, data is generated as:



(a) OPS with label drift.

(b) OPS with regression-function drift.

Figure 6.4: The adaptive behavior of OPS for the simulated label shift and regression-function drift datasets described in Section 6.1.2. For more details on the contents of the figure, please refer to Figure 6.3. The improvement in calibration and accuracy of OPS over the base model is visually apparent, but for completeness, {Acc, CE} values are reported in the Appendix as part of Figures 6.10 and 6.11.

$$\begin{aligned} Y_t &\sim \text{Bernoulli}(0.95(1-\alpha_t) + 0.05\alpha_t), \\ &\quad \text{where } \alpha_t = (t-1)/6000); \\ X_t | Y_t &\sim \mathbb{1}\{Y_t = 0\}\mathcal{N}(0,1) + \mathbb{1}\{Y_t = 1\}\mathcal{N}(2,1). \end{aligned} \tag{6.4}$$

Thus, $X_t \mid Y_t$ is fixed while the label distribution drifts. We follow the same training and test splits described in the covariate drift experiment, but without sinusoidal featurization of X_t; the base logistic regression model is trained directly on the scalar X_t's. The gap between f and the true model increases over time but OPS adapts well (Figure 6.4a).

Regression-function drift. For $t = 1, 2, \ldots, 6000$, the data is generated as follows: $\alpha_t = (t-1)/5000$,

$$X_t \sim \mathcal{N}(0, 10) \text{ and } Y_t | X_t \sim \text{Bernoulli}(p_t), \text{ where} \tag{6.5}$$
$$p_t = \begin{cases} 0.1(1-\alpha_t) + 0.5\alpha_t & \text{if } \text{mod}(\lfloor X_t/5 \rfloor, 2) = 0, \\ 0.9(1-\alpha_t) + 0.5\alpha_t & \text{if } \text{mod}(\lfloor X_t/5 \rfloor, 2) = 1. \end{cases}$$

Thus the distribution of X_t is fixed, but the regression function $\Pr(Y_t = 1 \mid X_t)$ drifts over time. We follow the same training and test splits described in the covariate drift experiment, as well as the 48-dimensional featurization and logistic regression modeling. The performance of the base model worsens over time, while OPS adapts (Figure 6.4b).

6.2 Online Platt scaling (OPS)

In a batch post-hoc setting, the Platt scaling parameters are set to those that minimize log-loss over the calibration data. If we view the first t instances in our stream as the calibration data, the fixed-batch Platt scaling parameters are,

$$(\widehat{a}_t, \widehat{b}_t) = \underset{(a,b)\in\mathbb{R}^2}{\arg\min} \sum_{s=1}^{t} l(m^{a,b}(f(\mathbf{x}_s)), y_s), \tag{6.6}$$

where $l(p, y) = -y \log p - (1 - y) \log(1 - p)$ and $m^{a,b}$ is defined in (6.2). Observe that this is exactly logistic regression over the dataset $(\text{logit}(f(\mathbf{x}_s)), y_s)_{s=1}^{t}$.

The thesis of OPS is that as more data is observed over time, we should use it to update the Platt scaling parameters. Define $p_t^{\text{OPS}} := m^{a_t,b_t}(f(\mathbf{x}_t))$, where (a_t, b_t) depends (in some yet undefined fashion) on $\{(f(\mathbf{x}_1), y_1), \ldots, (f(\mathbf{x}_{t-1}), y_{t-1})\}$.[1] One way to compare methods in this online setting is to consider *regret* R_T with respect to a reference ℓ_2-ball of radius B, $\mathcal{B} := \{(a, b) \in \mathbb{R}^2 : a^2 + b^2 \leqslant B^2\}$:

$$R_T = \sum_{t=1}^{T} l(p_t^{\text{OPS}}, y_t) - \min_{(a,b)\in\mathcal{B}} \sum_{t=1}^{T} l(m^{a,b}(f(\mathbf{x}_t)), y_t). \tag{6.7}$$

R_T is the difference between the total loss incurred when playing (a_t, b_t) at times $t \leqslant T$ and the total loss incurred when playing the single optimal $(a, b) \in \mathcal{B}$ for all $t \leqslant T$. Typically, we are interested in algorithms that have low R_T irrespective of how $(\mathbf{x}_t, y_t)$ is generated.

6.2.1 Logarithmic worst-case regret bound for OPS

OPS regret minimization is exactly online logistic regression (OLR) regret minimization over "pseudo-features" $\text{logit}(f(\mathbf{x}_t))$. Thus our OPS problem is immediately solved using OLR methods. A number of OLR methods have been proposed, and we consider their regret guarantees and running times for the OPS problem. These bounds typically depend on T and two problem-dependent parameters: the dimension (say d) and B, the radius of $\mathcal{B}$.

1. In our case, $d = 2$ since there is one feature $\text{logit}(f(\mathbf{x}))$ and a bias term. Thus d is a constant.
2. B could technically be large, but in practice, if f is not highly miscalibrated, we expect small values of a and b which would in turn lead to small B. This was true in all our experiments.

Regret bounds and running times for candidate OPS methods are presented in Table 6.1, which is an adaptation of Table 1 of Jézéquel et al. (2020) with all poly(d) terms removed. Based on this table, we identify AIOLI and Online Newton Step (ONS) as the best candidates for implementing OPS, since they both have $O(\log T)$ regret and $\widetilde{O}(T)$ running time. In the following theorem,

[1] A variant of this setup also allows (a_t, b_t) to depend on $f(\mathbf{x}_t)$ (Foster et al., 2018).

Algorithm	Regret	Running time
Online Gradient Descent (OGD) (Zinkevich, 2003)	$B\sqrt{T}$	T
Online Newton Step (ONS) (Hazan et al., 2007)	$e^B \log T$	T
AIOLI (Jézéquel et al., 2020)	$B\log(BT)$	$T\log T$
Aggregating Algorithm (AA) (Vovk, 1990; Foster et al., 2018)	$\log(BT)$	$B^{18}T^{24}$

Table 6.1: Asymptotic regret and running times of online logistic regression (OLR) algorithms for OPS as functions of the radius of reference class B and time-horizon T. For general OLR, regret and running times also depend on the dimension of $\mathcal{X}$. However, OPS effectively reduces the dimensionality of $\mathcal{X}$ to 2, so that a second-order method like ONS runs almost as fast as a first-order method like OGD. Also note that $B = \sqrt{a^2 + b^2}$ is small if the base model f is not highly miscalibrated. ONS with fixed hyperparameters was chosen for all OPS experiments; see Section 6.2.2 for implementation details.

we collect explicit regret guarantees for OPS based on ONS and AIOLI. Since the log-loss can be unbounded if the predicted probability equals 0 or 1, we require some restriction on $f(\mathbf{x}_t)$.

Theorem 6.1. *Suppose $\forall t, f(\mathbf{x}_t) \in [0.01, 0.99]$, $B \geqslant 1$, and $T \geqslant 10$. Then, for any sequence $(\mathbf{x}_t, y_t)_{t=1}^T$, OPS based on ONS satisfies*

$$R_T(\mathit{ONS}) \leqslant 2(e^B + 10B)\log T + 1, \tag{6.8}$$

while OPS based on AIOLI satisfies

$$R_T(\mathit{AIOLI}) \leqslant 22B\log(BT). \tag{6.9}$$

The ONS result follows from Hazan (2016, Theorem 4.5) and the AIOLI result follows from Jézéquel et al. (2020, Theorem 1), plugging in the appropriate values for problem-dependent parameters; more details are in Appendix 6.E. Since log-loss is a proper scoring rule (Gneiting and Raftery, 2007), minimizing it has implications for calibration (Bröcker, 2009). However, no "absolute" calibration bounds can be shown as discussed shortly in Section 6.2.4.

6.2.2 Hyperparameter-free ONS implementation

In our experiments, we found ONS to be significantly faster than AIOLI while also giving better calibration. Further, ONS worked without any hyperparameter tuning after an initial investigation was done to select a single set of hyperparameters. Thus we used ONS for experiments based on a *verbatim implementation* of Algorithm 12 in Hazan (2016), with $\gamma = 0.1$, $\rho = 100$, and $\mathcal{K} = \{(a, b) : \|(a, b)\|_2 \leqslant 100\}$. Algorithm 6.1 in the Appendix contains pseudocode for our final OPS implementation.

6.2.3 Follow-The-Leader as a baseline for OPS

The Follow-The-Leader (FTL) algorithm sets $(a_t, b_t) = (\widehat{a}_{t-1}, \widehat{b}_{t-1})$ (defined in (6.6)) for $t \geqslant 1$. This corresponds to solving a logistic regression optimization problem at every time step, making the overall complexity of FTL $\Omega(T^2)$. Further, FTL has $\Omega(T)$ worst-case regret. Since full FTL is intractably slow to implement even for an experimental comparison, we propose to use a computationally cheaper variant, called Windowed Platt Scaling (WPS). In WPS the optimal parameters given all current data, $(\widehat{a}_t, \widehat{b}_t)$, are computed and updated every $O(100)$ steps instead of at every time step. We call this a *window* and the exact size of the window can be data-dependent. The optimal parameters computed at the start of the window are used to make predictions until the end of that window, then they are updated for the next window. This heuristic version of FTL performs well in practice (Section 6.4).

6.2.4 Limitations of regret analysis

Regret bounds are relative with respect to the best in class, so Theorem 6.1 implies that OPS will do no worse than the best Platt scaling model in hindsight. However, even for i.i.d. data, the best Platt scaling model is itself miscalibrated on some distributions (Gupta et al., 2020, Theorem 3). This latter result shows that some form of binning must be deployed to be calibrated for arbitrarily distributed i.i.d. data. Further, if the data is adversarial, any deterministic predictor can be rendered highly miscalibrated (Oakes, 1985; Dawid, 1985); a simple strategy is to set $y_t = \mathbb{1}\{p_t \leqslant 0.5\}$. In a surprising seminal result, Foster and Vohra (1998) showed that adversarial calibration is possible by randomizing/hedging between different bins. The following section shows how one can perform such binning and hedging on top of OPS, based on a technique called calibeating.

6.3 Calibeating the OPS forecaster

Calibeating (Foster and Hart, 2023) is a technique to improve or "beat" an expert forecaster. The idea is to first use the expert's forecasts to allocate data to representative *bins*. Then, the bins are treated *nominally*: they are just names or tags for "groups of data-points that the expert suggests are similar". The final forecasts in the bins are computed using only the outcome (y_t) values of the points in the bin (seen so far), with no more dependence on the expert's original forecast. The intuition is that forecasting inside each bin can be done in a theoretically valid sense, irrespective of the theoretical properties of the expert.

We will use the following "ϵ-bins" to perform calibeating:

$$B_1 = [0, \epsilon), B_2 = [\epsilon, 2\epsilon), \ldots, B_m = [1 - \epsilon, 1]. \tag{6.10}$$

Here $\epsilon > 0$ is the width of the bins, and for simplicity we assume that $m = 1/\epsilon$ is an integer. For instance, one could set $\epsilon = 0.1$ or the number of bins $m = 10$, as we do in the experiments in Section 6.4. Two types of calibeating—tracking and hedging—are described in the following subsections. We suggest recalling our illustration of calibeating in the introduction (Figure 6.1c).

6.3.1 Calibeating via tracking past outcomes in bins

Say at some t, the expert forecasts $p_t \in [0.7, 0.8)$. We look at the instances $s < t$ when $p_s \in [0.7, 0.8)$ and compute

$$\bar{y}^b_{t-1} = \text{Average}\{y_s : s < t, p_s \in [0.7, 0.8)\}.$$

Suppose we find that $\bar{y}^b_{t-1} = 0.85$. That is, when the expert forecasted bin $[0.7, 0.8)$ in the past, the average outcome was 0.85. A natural idea now is to forecast 0.85 instead of 0.75. We call this process "Tracking", and it is the form of calibeating discussed in Section 4 of Foster and Hart (2023). In our case, we treat OPS as the expert and call the tracking version of OPS as TOPS. If $p^{\text{OPS}}_t \in B_b$, then

$$p^{\text{TOPS}}_t := \text{Average}\{y_s : s < t, p^{\text{OPS}}_s \in B_b\}. \tag{6.11}$$

The average is defined as the mid-point of B_b if the set above is empty.

Foster and Hart (2023) showed that the Brier-score of the TOPS forecasts p^{TOPS}_t, defined as $\frac{1}{T}\sum_{t=1}^{T}(y_t - p^{\text{TOPS}}_t)^2$, is better than the corresponding Brier-score of the OPS forecasts p^{OPS}_t, by roughly the squared calibration error of p^{OPS}_t (minus a $\log T$ term). In the forthcoming Theorem 6.2, we derive a result for a different object that is often of interest in post-hoc calibration, called sharpness.

6.3.2 Segue: defining sharpness of forecasters

Recall the ϵ-bins introduced earlier (6.10). Define $N_b = |\{t \leqslant T : p_t \in B_b\}|$ and $\widehat{y}_b = \frac{1}{N_b}\sum_{t \leqslant T, p_t \in B_b} y_t$ if $N_b > 0$, else $\widehat{y}_b = 0$. Sharpness is defined as,

$$\text{SHP}(p_{1:T}) := \frac{1}{T}\sum_{b=1}^{m} N_b \cdot \widehat{y}_b^2.\text{[2]} \tag{6.12}$$

If the forecaster is perfectly knowledgeable and forecasts $p_t = y_t$, its SHP equals $\sum_{t=1}^{T} y_t/T =: \bar{y}_T$. On the other hand, if the forecaster puts all points into a single bin b, its SHP equals $(\sum_{t=1}^{T} y_t/T)^2 = \bar{y}_T^2$. The former forecaster is precise or *sharp*, while the latter is not, and SHP captures this—it can be shown that $\bar{y}_T^2 \leqslant \text{SHP}(p_{1:T}) \leqslant \bar{y}_T$. We point the reader to Bröcker (2009) for further background. One of the goals of effective forecasting is to ensure high sharpness (Gneiting et al., 2007). OPS achieves this goal by relying on the log-loss, a proper scoring rule. The following theorem shows that TOPS suffers a small loss in SHP compared to OPS.

[2] The original definition of sharpness (Murphy, 1973) was (essentially): $-T^{-1}\sum_{b=1}^{m} N_b\widehat{y}_b(1 - \widehat{y}_b)$, which equals $\text{SHP}(p_{1:T}) - \bar{y}_T$. We add the forecast-independent term $\bar{y}_T$ on both sides and define the (now non-negative) quantity as SHP.

Theorem 6.2. *The sharpness of TOPS forecasts satisfies*

$$\mathit{SHP}(p_{1:T}^{\mathit{TOPS}}) \geqslant \mathit{SHP}(p_{1:T}^{\mathit{OPS}}) - \epsilon - \frac{\epsilon^2}{4} - \frac{\log T + 1}{\epsilon T}. \tag{6.13}$$

The proof (in Appendix 6.E) uses Theorem 3 of Foster and Hart (2023) and relationships between sharpness, Brier-score, and a quantity called refinement. If T is fixed and known, setting $\epsilon \approx \sqrt{\log T / T}$ (including constant factors), or equivalently, the number of bins $B \approx \sqrt{T / \log T}$ gives a rate of $\widetilde{O}(\sqrt{1/T})$ for the SHP difference term. While we do not show a calibration guarantee, TOPS had the best calibration performance in most experiments (Section 6.4)

6.3.3 Calibeating via hedging or randomized prediction

All forecasters introduced so far—the base model f, OPS, and TOPS—make forecasts p_t that are deterministic given the past data until time $t-1$. If the y_t sequence is being generated by an adversary that acts after seeing p_t, then the adversary can ensure that each of these forecasters is miscalibrated by setting $y_t = \mathbb{1}\{p_t \leqslant 0.5\}$.

Suppose instead that the forecaster is allowed to hedge—randomize and draw the forecast from a distribution instead of fixing it to a single value—and the adversary only has access to the distribution and not the actual p_t. Then there exist hedging strategies that allow the forecaster to be arbitrarily well-calibrated (Foster and Vohra, 1998). In fact, Foster (1999, henceforth F99) showed that this can be done while hedging between two arbitrarily close points in $[0, 1]$.

In practice, outcomes are not adversarial, and covariates are available. A hedging algorithm that does not use covariates cannot be expected to give informative predictions. We verify this intuition through an experiment in Appendix on historical rain data 6.D—F99's hedging algorithm simply predicts the average y_t value in the long run.

A best-of-both-worlds result can be achieved by using the expert forecaster to bin data using $\mathbf{x}_t$ values, just like we did in Section 6.3.1. Then, inside every bin, a separate hedging algorithm is instantiated. For the OPS predictor, this leads to HOPS (OPS + hedging). Specifically, in our experiments and the upcoming calibration error guarantee, we used F99:

$$p_t^{\text{HOPS}} := \text{F99}(y_s : s < t, p_s \in B_b). \tag{6.14}$$

A standalone description of F99 is included in Appendix 6.C. F99 hedges between consecutive mid-points of the ϵ-bins defined earlier (6.10). The only hyperparameter for F99 is ϵ. For the experiments in Section 6.4, we set $\epsilon = 0.1$; other ϵ values are considered for a limited set of experiments in Appendix 6.A.3. To be clear, p_t is binned on the ϵ-bins, and the hedging inside each bin is again over the ϵ-bins.

The upcoming theorem shows a SHP lower bound on HOPS. In addition, we show an assumption-free upper bound on the (ℓ_1-)calibration error, defined as

$$\text{CE}(p_{1:T}) := \frac{1}{T}\sum_{b=1}^{m} N_b \cdot |\widehat{p}_b - \widehat{y}_b|\,, \tag{6.15}$$

where $N_b, \widehat{y}_b$ were defined in Section 6.3.2, and $\widehat{p}_b = \frac{1}{N_b}\sum_{t\leqslant T, p_t \in B_b} p_t$, if $N_b > 0$, else $\widehat{p}_b =$ mid-point(B_b). Achieving small CE is one formalization of (6.1). The following result is conditional on the $y_{1:T}$, $p_{1:T}^{\text{OPS}}$ sequences. The expectation is over the randomization in F99.

Theorem 6.3. *For adversarially generated data, the expected sharpness of HOPS forecasts using the forecast hedging algorithm of Foster (1999) is lower bounded as*

$$\mathbb{E}\left[\mathit{SHP}(p_{1:T}^{\mathit{HOPS}})\right] \geqslant \mathit{SHP}(p_{1:T}^{\mathit{OPS}}) - \left(\epsilon + \frac{\log T + 1}{\epsilon^2 T}\right), \tag{6.16}$$

and the expected calibration error of HOPS satisfies,

$$\mathbb{E}\left[\mathit{CE}(p_{1:T}^{\mathit{HOPS}})\right] \leqslant \epsilon/2 + 2\sqrt{1/\epsilon^2 T}. \tag{6.17}$$

The proof in Appendix 5.H is based on Theorem 5 of Foster and Hart (2023) and a CE bound for F99 based on Blackwell approachability (Blackwell, 1956). With $\epsilon = \widetilde{\Theta}(T^{-1/3})$, the difference term in the SHP bound is $\widetilde{O}(T^{-1/3})$ and with $\epsilon = \widetilde{\Theta}(T^{-1/4})$, the CE bound is $\widetilde{O}(T^{-1/4})$. Compare (6.17) to the usual (without calibeating) calibration bound of $O(\epsilon + 1/\sqrt{\epsilon T})$ which leads to $O(T^{-1/3})$ (Foster and Vohra, 1998). High-probability versions of (6.17) can be derived using probabilistic Blackwell approachability lemmas, such as those in Perchet (2014)

The "Online Recalibration" method of Kuleshov and Ermon (2017, Algorithm 1) amounts to performing the same binning and hedging that we have described, but on top of a black-box expert. We used a specific expert, OPS, and experimentally demonstrate its benefits on multiple datasets (Section 6.1.2 and 6.4). Theoretically, our calibration bound (6.17) is identical to their Lemma 3 (if Lemma 3 is instantiated with F99). Their Lemma 2 shows a bound on the expected increase of any bounded proper loss on performing the calibeating step. For the case of Brier-loss their bound is $O(\epsilon + 1/\epsilon^2\sqrt{T})$. Our proof of (6.16) can be used to show an improved bound of $O(\epsilon + \log T/\epsilon^2 T)$, as stated formally in Appendix 6.E (Theorem 6.4)

6.4 Experiments

We perform experiments with synthetic and real-data, in i.i.d. and distribution drift setting. Code to reproduce the experiments can be found at https://github.com/aigen/df-posthoc-calibration (see Appendix 6.A.4 for more details). All baseline and proposed methods are described in Collection 1 on the following page. In each experiment, the **base model** f was a random forest (`sklearn`'s implementation). All default parameters were used, except `n_estimators` was set to 1000. No hyperparameter tuning on individual datasets was performed for any of the recalibration methods.

Metrics. We measured the SHP and CE metrics defined in (6.12) and (6.15) respectively. Although estimating population versions of SHP and CE in statistical (i.i.d.) settings is fraught with several issues (Kumar et al. (2019) and Roelofs et al. (2022) and several other works), our definitions

Collection 1. Proposed and baseline methods for online post-hoc calibration. Final forecasts are identified in blue.

```
Input: $f : \mathcal{X} \to [0, 1]$, any pre-learnt model
Input: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_T, y_T) \in \mathcal{X} \times \{0, 1\}$
Input: calibration-set-size $T_{\text{cal}} < T$, window-size $W$
Fixed Platt scaling: $(a^{\text{FPS}}, b^{\text{FPS}}) \leftarrow (\widehat{a}_{T_{\text{cal}}}, \widehat{b}_{T_{\text{cal}}})$ (eq. 6.6)
Windowed Platt scaling: $(a^{\text{WPS}}, b^{\text{WPS}}) \leftarrow (a^{\text{FPS}}, b^{\text{FPS}})$
Online Platt scaling: $(a_1^{\text{OPS}}, b_1^{\text{OPS}}) \leftarrow (1, 0)$
for $t = 2$ to $T$ do
    $(a_t^{\text{OPS}}, b_t^{\text{OPS}}) \leftarrow \text{ONS}((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{t-1}, y_{t-1}))$
    (ONS is Algorithm 6.1 in the Appendix)
end for
for $t = T_{\text{cal}} + 1$ to $T$ do
    $p_t^{\text{BM}} \leftarrow f(\mathbf{x}_t)$
    $p_t^{\text{FPS}} \leftarrow \text{sigmoid}(a^{\text{FPS}} \cdot \text{logit}(f(\mathbf{x}_t)) + b^{\text{FPS}})$
    $p_t^{\text{WPS}} \leftarrow \text{sigmoid}(a^{\text{WPS}} \cdot \text{logit}(f(\mathbf{x}_t)) + b^{\text{WPS}})$
    $p_t^{\text{OPS}} \leftarrow \text{sigmoid}(a_t^{\text{OPS}} \cdot \text{logit}(f(\mathbf{x}_t)) + b_t^{\text{OPS}})$
    $p_t^{\text{TOPS}}$ is set using past $(y_s, p_s^{\text{OPS}})$ values as in (6.11)
    $p_t^{\text{HOPS}}$ is set using past $(y_s, p_s^{\text{OPS}})$ values as in (6.14)
    If $\mod (t - T_{\text{cal}}, W) = 0$, $(a^{\text{WPS}}, b^{\text{WPS}}) \leftarrow (\widehat{a}_t, \widehat{b}_t)$
end for
```

target actual observed quantities which are directly interpretable without reference to population quantities.

Reading the plots. The plots we report show CE values at certain time-stamps starting from $T_{\text{cal}} + 2W$ and ending at T (see third line of Collection 1). T_{cal} and W are fixed separately for each dataset (Table 6.2 in Appendix). We also generated SHP plots, but these are not reported since the drop in SHP was always very small.

6.4.1 Experiments on real datasets

We worked with four public datasets in two settings. Links to the datasets are in Appendix 6.A.1.

Distribution drift. We introduced synthetic drifts in the data based on covariate values, so this is an instance of covariate drift. For example, in the bank marketing dataset (leftmost plot in Figure 6.5), the problem is to predict which clients are likely to subscribe to a term deposit if they are targeted for marketing, using covariates like `age`, `education`, and `bank-balance`. We ordered the available 12000 rows roughly by `age` by adding a random number uniformly from $\{-1, 0, 1\}$ to `age` and sorting all the data. Training is done on the first 1000 points, $T_{\text{cal}} = 1000$, and $W = 500$. Similar drifts are induced for the other datasets, and T_{cal}, W values are set depending on the total number of points; further details are in Appendix 6.A.1.

All simulations were performed 100 times and the average CE and SHP values with ± std-deviation errorbars were evaluated at certain time-steps. Thus, our lines correspond to estimates

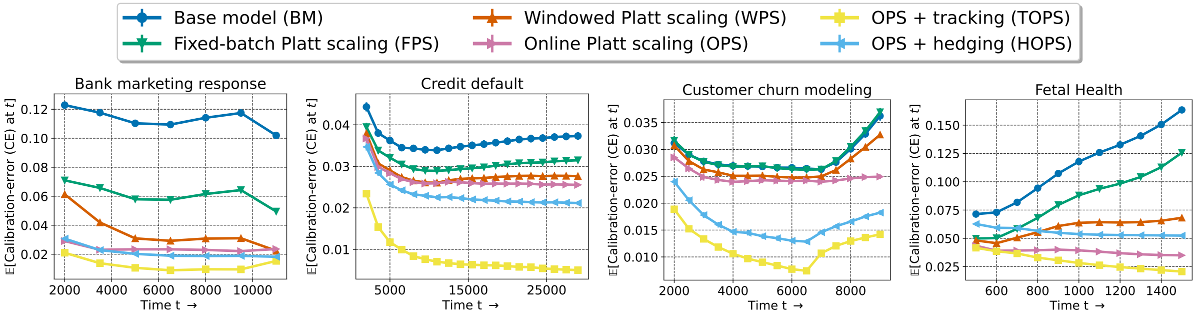


Figure 6.5: **Drifting data.** CE (calibration error) values over time of considered models on four datasets with synthetically induced drifts. The plots have invisible error bars since variation across the 100 runs was small. OPS consistently performs better than BM, FPS, and WPS, while TOPS is the best-performing among all methods across datasets and time. All methods had roughly the same SHP values at a given time-step, so the SHP plots are delayed to Appendix 6.A (Figure 6.8).

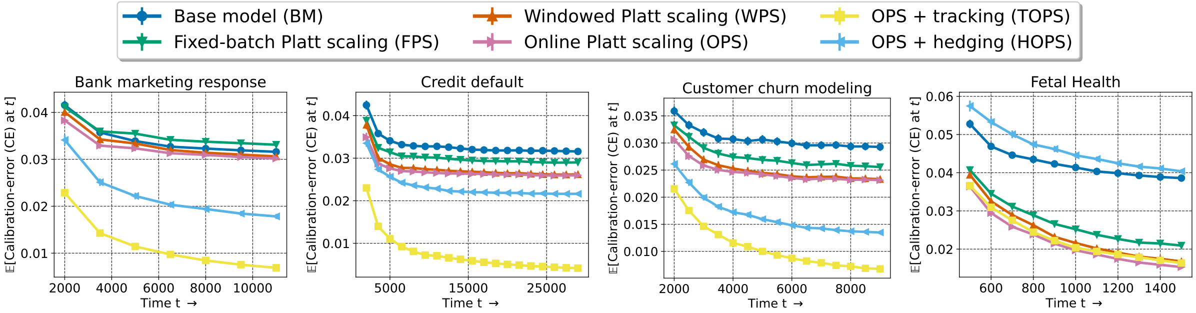


Figure 6.6: **IID data.** CE values over time of considered models with four randomly shuffled (ie, nearly i.i.d.) datasets. The plots have invisible error bars since variation across runs was small. TOPS achieves the smallest values of CE throughout.

of the expected values of CE and SHP, as indicated by the Y-axis labels. We find that across datasets, OPS has the least CE among non-calibeating methods, and both forms of calibeating typically improve OPS further (Figure 6.5). Specifically, TOPS performs the best by a margin compared to other methods. We also computed SHP values, which are reported in Appendix 6.A (Figure 6.8). The drop in SHP is insignificant in each case (around 0.005).

IID data. This is the usual batch setting formed by shuffling all available data. Part of the data is used for training and the rest forms the test-stream. We used the same values of T_{cal} and W as those used in the data drift experiments (see Appendix 6.A.1). In our experiments, we find that the gap in CE between BM, FPS, OPS, and WPS is smaller (Figure 6.6). However, TOPS performs the best in all scenarios, typically by a margin. Here too, the change in SHP was small, so those plots were delayed to Appendix 6.A (Figure 6.9).

6.4.2 Synthetic experiments

In all experiments with real data, WPS performs almost as good as OPS. In this subsection, we consider some synthetic data drift experiments where OPS and TOPS continue performing well, but WPS performs much worse.

Covariate drift. Once for the entire process, we draw random orthonormal vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^{10}$ ($\|\mathbf{v}_1\|_2 = \|\mathbf{v}_2\|_2 = 1$, $\mathbf{v}_1^\intercal \mathbf{v}_2 = 0$), a random weight vector $\mathbf{w} \in \{-1, 1\}^{10+\binom{10}{2}}$ with each component set to 1 or -1 independently with probability 0.5, and set a drift parameter $\delta \geqslant 0$. The data is generated as follows:

$$\mathbf{u}_t = \mathbf{v}_1 \cos(\delta t) + \mathbf{v}_2 \sin(\delta t), X_t \sim \mathcal{N}(\mathbf{0}, \mathcal{I}_{10} + 10\mathbf{u}_t\mathbf{u}_t^\intercal),$$
$$Y_t | X_t \sim \text{Bernoulli}(\text{sigmoid}(\mathbf{w}^\intercal \widetilde{X}_t)), \text{ where}$$
$$\widetilde{X}_t = [\mathbf{x}_1, \ldots, \ \mathbf{x}_{10}, \ \mathbf{x}_1\mathbf{x}_2, \ \mathbf{x}_1\mathbf{x}_3, \ldots, \ \mathbf{x}_9\mathbf{x}_{10}] \in \mathbb{R}^{10+\binom{10}{2}}.$$

Thus the distribution of Y_t given X_t is fixed as a logistic model over the expanded representation $\widetilde{X}_t$ that includes all cross-terms (this is unknown to the forecaster who only sees X_t). The features X_t themselves are normally distributed with mean $\mathbf{0}$ and a time-varying covariance matrix. The principal component (PC) of the covariance matrix is a vector $\mathbf{u}_t$ that is rotating on the two-dimensional plane containing the orthonormal vectors $\mathbf{v}_1$ and $\mathbf{v}_2$. The first 1000 points are used as training data, the remaining $T = 5000$ form a test-stream, and $W = 500$. We report results in two settings: one is i.i.d., that is $\delta = 0$, and the other is where the $\mathbf{u}$ for the first and last point are at a 180° angle (Figure 6.7a).

Label drift. Given some $\delta > 0$, (X_t, Y_t) is generated as:

$$Y_t \sim \text{Bernoulli}(0.5 + \delta t),$$
$$X_t | Y_t \sim \mathbb{1}\{Y_t = 0\} \mathcal{N}(\mathbf{0}, \mathbb{R}^{10}) + \mathbb{1}\{Y_t = 1\} \mathcal{N}(\mathbf{e_1}, \mathbb{R}^{10}).$$

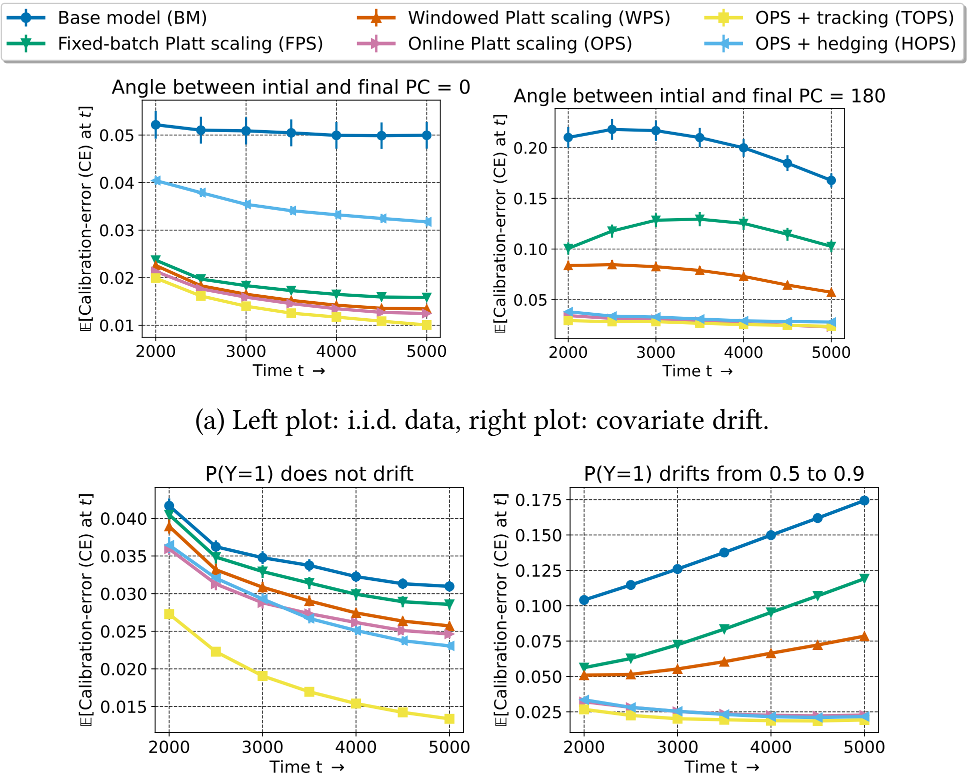
Thus $P(Y_1 = 1) = 0.5 + \delta$ and for the last test point, $P(Y_{6000} = 1) = 0.5 + 6000\delta$. This final value can be set to control the extent of label drift; we show results with no drift (i.e., $\delta = 0$, Figure 6.7b left) and δ set so that final bias $0.5 + 6000\delta = 0.9$ (Figure 6.7b right). The number of training points is 1000, $T = 5000$, and $W = 500$.

6.4.3 Changing ϵ, histogram binning, beta scaling

In Appendix 6.A.3, we report versions of Figures 6.5, 6.6 with $\epsilon = 0.05, 0.2$ (instead of $\epsilon = 0.1$) with similar conclusions (Figures 6.12, 6.13). We also perform comparisons with a windowed version of the popular histogram binning method (Zadrozny and Elkan, 2001) and online versions of the beta scaling method, as discussed in the forthcoming Section 6.5.

6.5 Online beta scaling with calibeating

A recalibration method closely related to Platt scaling is beta scaling (Kull et al., 2017). The beta scaling mapping m has three parameters $(a, b, c) \in \mathbb{R}^3$, and corresponds to a sigmoid transform



(a) Left plot: i.i.d. data, right plot: covariate drift.

(b) Left plot: i.i.d. data, right plot: label drift.

Figure 6.7: Experiments with synthetic data. In all cases, TOPS has the lowest CE across time.

over two pseudo-features derived from $f(\mathbf{x})$: $\log(f(\mathbf{x}))$ and $\log(1 - f(\mathbf{x}))$,

$$m^{a,b,c}(f(\mathbf{x})) := \text{sigmoid}(a \cdot \log(f(\mathbf{x})) + b \cdot \log(1 - f(\mathbf{x})) + c).$$

Observe that enforcing $b = -a$ recovers Platt scaling since $\text{logit}(z) = \log(z) - \log(1 - z)$. The beta scaling parameters can be learnt following identical protocols as Platt scaling: (i) the traditional method of fixed batch post-hoc calibration akin to FPS, (ii) a natural benchmark of windowed updates akin to WPS, and (iii) regret minimization based method akin to OPS. This leads to the methods FBS, WBS, and OBS, replacing the "P" of Platt with the "B" of beta. Tracking + OBS (TOBS) and Hedging + OBS (HOBS) can be similarly derived. Further details on all beta scaling methods are in Appendix 6.B, where we also report plots similar to Figures 6.5, 6.6 for beta scaling (Figure 6.15). In a comparison between histogram binning, beta scaling, Platt scaling, and their tracking versions, TOPS and TOBS are the best-performing methods across experiments (Figure 6.14).

6.6 Conclusion

We provided a way to bridge the gap between the online (typically covariate-agnostic) calibration literature, where data is assumed to be adversarial, and the (typically i.i.d.) post-hoc calibration literature, where the joint covariate-outcome distribution takes centerstage. First, we adapted the post-hoc method of Platt scaling to the online setting, based on a reduction to logistic regression, to give our OPS algorithm. Second, we showed how calibeating can be applied on top of OPS to give further improvements.

The TOPS method we proposed has the lowest calibration error in all experimental scenarios we considered. On the other hand, the HOPS method which is based on online adversarial calibration provably controls miscalibration at any pre-defined level and could be a desirable choice in sensitive applications. The good performance of OPS+calibeating lends further empirical backing to the thesis that scaling+binning methods perform well in practice, as has also been noted in prior works (Zhang et al., 2020; Kumar et al., 2019). Our theoretical results formalize this empirical observation.

We note a few directions for future work. First, online algorithms that control regret on the most recent data have been proposed (Hazan and Seshadhri, 2009; Zhang et al., 2018). These approaches could give further improvements on ONS, particularly for drifting data. Second, while this chapter entirely discusses calibration for binary classification, all binary routines can be lifted to achieve multiclass notions such as top-label or class-wise calibration, as discussed in Chapter 5. Alternatively, multiclass versions of Platt scaling (Guo et al., 2017) such as temperature and vector scaling can also be targeted directly using online multiclass logistic regression (Jézéquel et al., 2021).

Appendices for Chapter 6

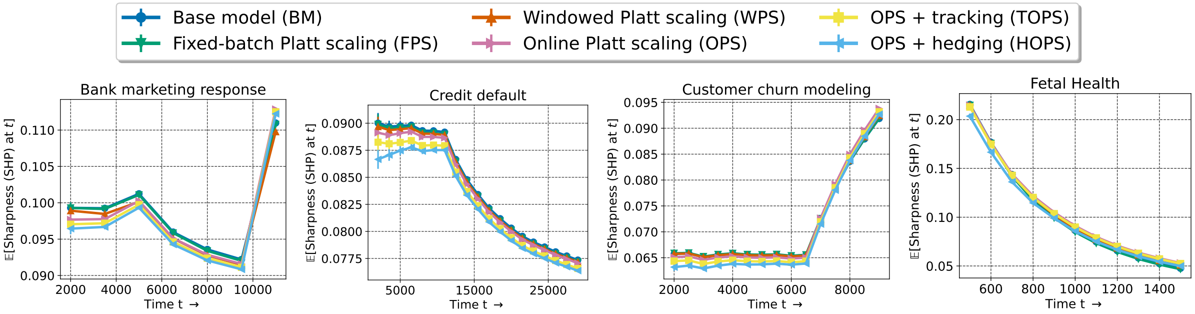


Figure 6.8: **Sharpness results with drifting data.** SHP values over time of considered models on four datasets with synthetically induced drifts (Section 6.4.1). The plots have invisible error bars since variation across the 100 runs was small. The drop in expected sharpness is below 0.005 at all times except on the Fetal Health Dataset.

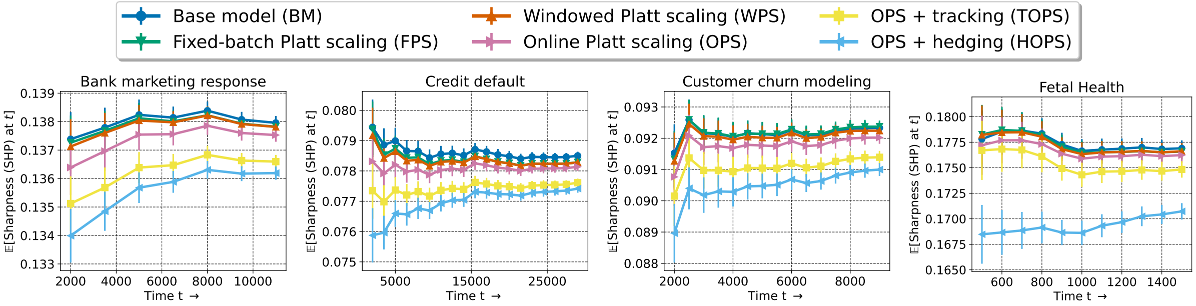


Figure 6.9: **Sharpness results with i.i.d. data.** SHP values over time of considered models on four shuffled (ie, nearly i.i.d.) datasets (Section 6.4.1). The drop in expected sharpness is less than 0.005 in all cases except for the HOPS forecaster on the Fetal Health dataset, where it is 0.01.

6.A Experimental details and additional results

Some implementation details, metadata, information on metrics, and additional results and figures are collected here.

Name	T_{train}	T_{cal}	W	Sort-by	Link to dataset
Bank marketing	1000	1000	500	Age	https://www.kaggle.com/datasets/kukuroo3/bank-marketing-response-predict
Credit default	1000	1000	500	Sex	https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset
Customer churn	1000	1000	500	Location	https://www.kaggle.com/datasets/shrutimechlearn/churn-modelling
Fetal health	626	300	100	Acceleration	https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification

Table 6.2: Metadata for datasets used in Section 6.4.1. The sort-by column indicates which covariate was used to order data points. All datasets are under the Creative Commons CC0 license.

6.A.1 Metadata for datasets used in Section 6.4.1

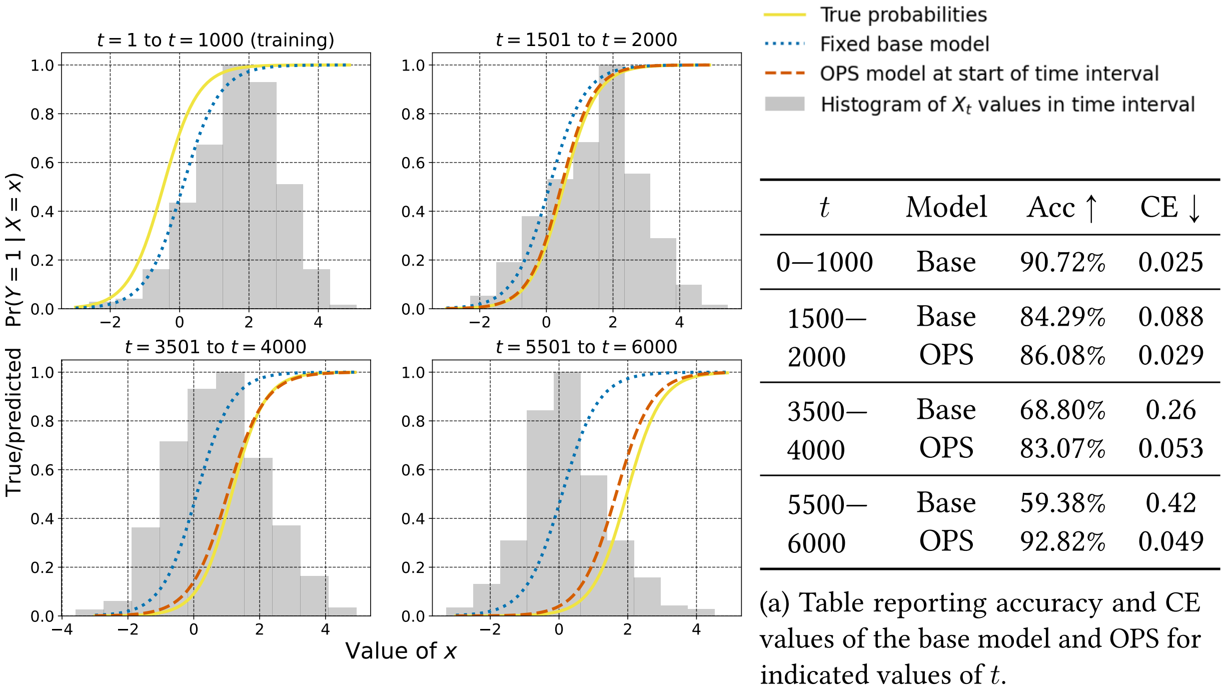
Table 6.2 contains metadata for the datasets we used in Section 6.4.1. T_{train} refers to the number of training examples. The "sort-by" column indicates which covariate was used to order data points. In each case some noise was added to the covariate in order to create variation for the experiments. The exact form of drift can be found in the python file `sec_4_experiments_core.py` in the repository https://github.com/AIgen/df-posthoc-calibration/tree/main/Online%20Platt%20Scaling%20with%20Calibeating.

6.A.2 Additional plots and details for label drift and regression-function drift experiments from Section 6.1

Figures 6.3, 6.10, and 6.11 report accuracy (Acc) and calibration error (CE) values for the base model and the OPS model in the three dataset drift settings we considered. The Acc values are straightforward averages and can be computed without issues. However, estimation of CE on real datasets is tricky and requires sophisticated techniques such as adaptive binning, debiasing, heuristics for selecting numbers of bins, or kernel estimators (Kumar et al., 2019; Roelofs et al., 2022; Widmann et al., 2019). The issue typically boils down to the fact that $\Pr(Y = 1 \mid X = x)$ cannot be estimated for every $x \in \mathcal{X}$ without making smoothness assumptions or performing some kind of binning. However, in the synthetic experiments of Section 6.1, $\Pr(Y = 1 \mid X)$ is known exactly, so such techniques are not required. For some subset of forecasts $p_s, p_2, \dots, p_t$, we compute

$$\text{CE} = \frac{1}{t - s + 1} \sum_{i=s}^{t} \left| p_i - \Pr(Y_i = 1 \mid X_i = \mathbf{x}_i) \right|,$$

on the instantiated values of $X_s, X_{s+1}, \dots, X_t$. Thus, what we report is the true CE given covariate values.



t	Model	Acc ↑	CE ↓
0—1000	Base	90.72%	0.025
1500—2000	Base	84.29%	0.088
	OPS	86.08%	0.029
3500—4000	Base	68.80%	0.26
	OPS	83.07%	0.053
5500—6000	Base	59.38%	0.42
	OPS	92.82%	0.049

(a) Table reporting accuracy and CE values of the base model and OPS for indicated values of t.

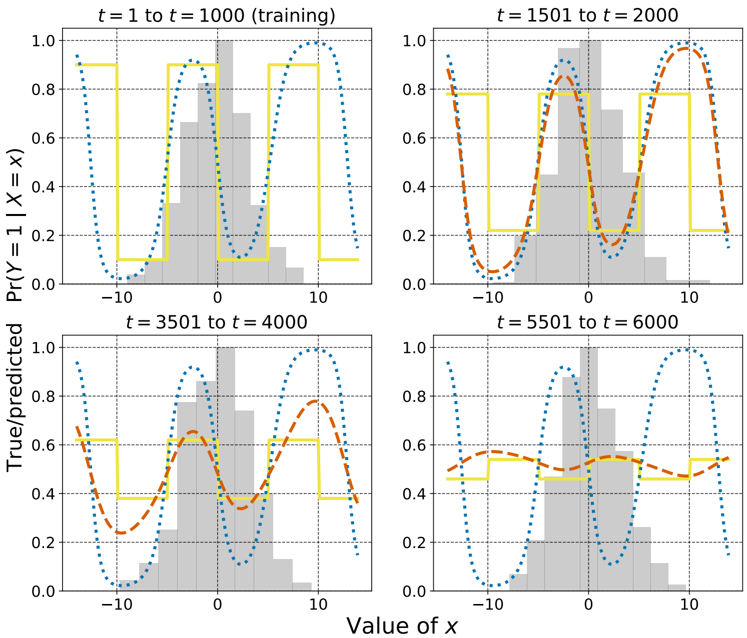
Figure 6.10: The adaptive behavior of OPS for the simulated label drift scenario described in Section 6.1.2.

6.A.3 Additional results with windowed histogram binning and changing bin width

Comparison to histogram binning (HB). HB is a recalibration method that has been shown to have excellent empirical performance as well as theoretical guarantees (Zadrozny and Elkan, 2001; Gupta and Ramdas, 2021). There are no online versions of HB that we are aware of, so we use the same windowed approach as windowed Platt and beta scaling for benchmarking (see Section 6.2.3 and the second bullet in Section 6.B). This leads to windowed histogram binning (WHB), the fixed-batch HB recalibrator that is updated every $O(100)$ time-steps. We compare WHB to OPS and OBS (see Section 6.5). Since tracking improves both OPS and OBS, we also consider tracking WHB. Results are presented in Figure 6.14.

We find that WHB often performs better than OPS and OBS in the i.i.d. case, and results are mixed in the drifting case. However, since WHB is a binning method, it inherently produces something akin to a running average, and so tracking does not improve it further. The best methods (TOPS, TOBS) are the ones that combine one of our proposed parametric online calibrators (OPS, OBS) with tracking.

Changing the bin width ϵ. In the main chapter, we used $\epsilon = 0.1$ and defined corresponding



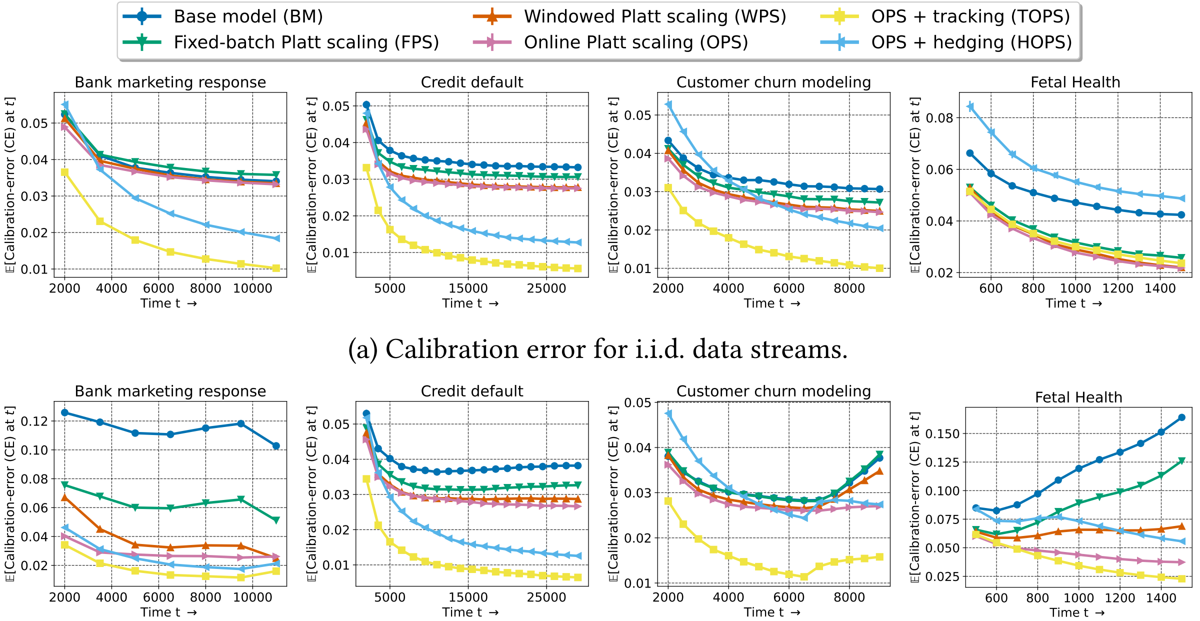
t	Model	Acc ↑	CE ↓
0—1000	Base	84.14%	0.1
1500— 2000	Base OPS	74.64% 75.40%	0.12 0.097
3500— 4000	Base OPS	59.51% 59.60%	0.2 0.067
5500— 6000	Base OPS	44.39% 51.43%	0.34 0.049

(a) Table reporting accuracy and CE values of the base model and OPS for indicated values of t.

Figure 6.11: The adaptive behavior of OPS for the simulated regression-function drift scenario described in Section 6.1.2.

bins as in (6.10). This binning reflects in three ways on the experiments we performed. First, ϵ-binning is used to divide forecasts into representative bins before calibeating (equations (6.11), (6.14)). Second, ϵ-binning is used to define the sharpness and calibration error metrics. Third, the hedging procedure F99 requires specifying a binning scheme, and we used the same ϵ-bins.

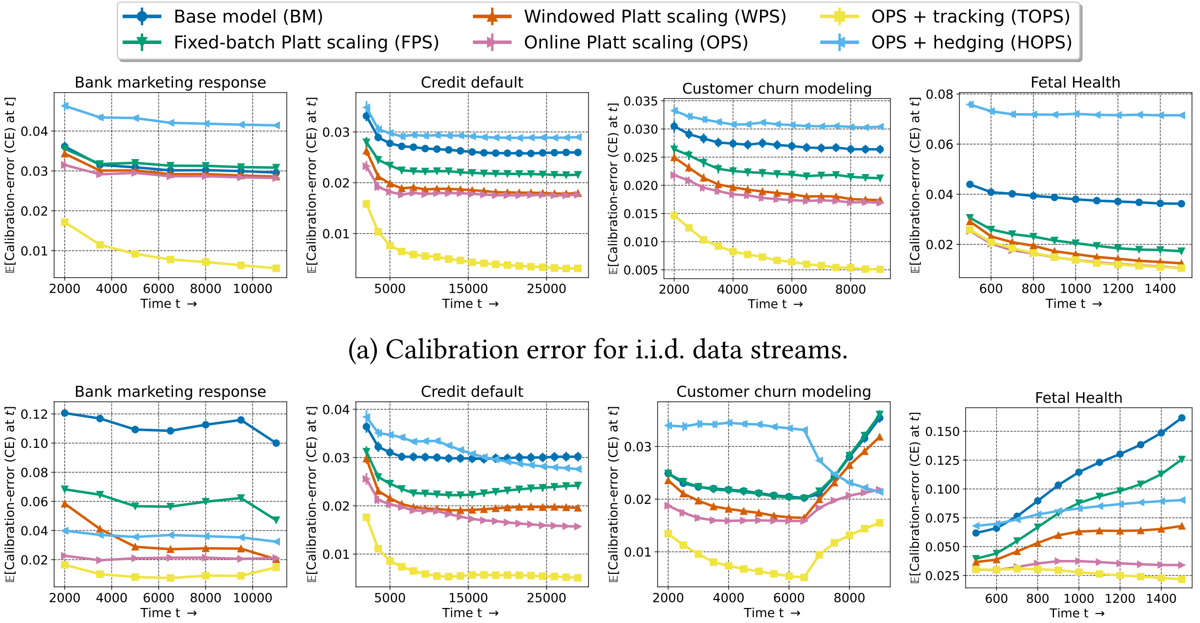
Here, we show that the empirical results are not dependent on the chosen representative value of $\epsilon = 0.1$. We run the same experiment used to produce Figures 6.5 and 6.6 but with $\epsilon = 0.05$ (Figure 6.12) and $\epsilon = 0.2$ (Figure 6.13). The qualitative results remain identical, with TOPS still the best performer and hardly affected by the changing epsilon. In fact, the plots for all methods except HOPS are indistinguishable from their $\epsilon = 0.1$ counterparts at first glance. HOPS is slightly sensitive to ϵ: the performance improves slightly with $\epsilon = 0.05$, and worsens slightly with $\epsilon = 0.2$.



(a) Calibration error for i.i.d. data streams.

(b) Calibration error for drifting data streams.

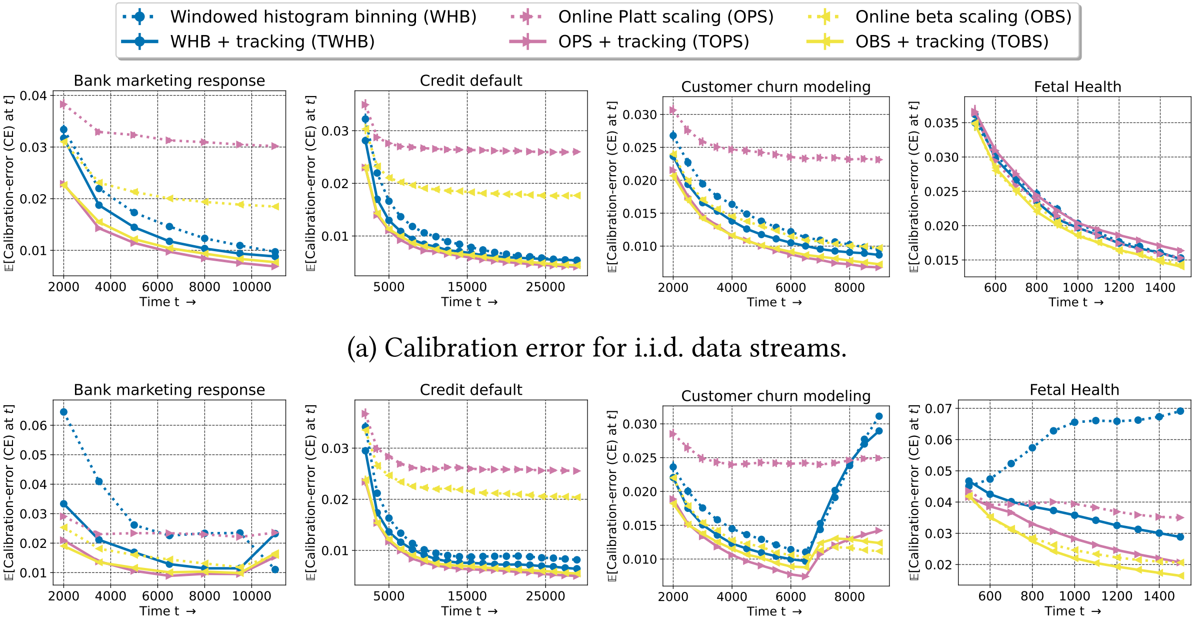
Figure 6.12: Results for the same experimental setup as Figures 6.5 and 6.6, but with $\epsilon = 0.05$.



(a) Calibration error for i.i.d. data streams.

(b) Calibration error for drifting data streams.

Figure 6.13: Results for the same experimental setup as Figures 6.5 and 6.6, but with $\epsilon = 0.2$.



(a) Calibration error for i.i.d. data streams.

(b) Calibration error for drifting data streams.

Figure 6.14: Comparing the performance of windowed histogram binning (WHB), online Platt scaling (OPS), online beta scaling (OBS), and their tracking variants on real datasets with and without distribution drifts. Among non-tracking methods (dotted lines), WHB performs well with i.i.d. data, while OBS performs well for drifting data. Among tracking methods (solid lines), TOBS and TOPS are the best-performing methods in every plot. Tracking typically does not improve WHB much since WHB is already a binning method (so tracking is implicit).

Algorithm 6.1 Online Newton Step for OPS (based on Hazan (2016, Algorithm 12))

```
Input: $\mathcal{K} = \{(x, y) : \|(x, y)\|_2 \leqslant 100\}$, time horizon $T$, and initialization parameter $(a_1^{\text{OPS}}, b_1^{\text{OPS}}) = (1, 0) =: \theta_1 \in \mathcal{K}$
Hyperparameters: $\gamma = 0.1$, $\rho = 100$
Set $A_0 = \rho \mathbf{I}_2$
for $t = 1$ to $T$ do
    Play $\theta_t$, observe log-loss $l(m^{\theta_t}(f(\mathbf{x}_t)), y_t)$ and its gradient $\nabla_t := \nabla_{\theta_t} l(m^{\theta_t}(f(\mathbf{x}_t)), y_t)$
    $A_t = A_{t-1} + \nabla_t \nabla_t^{\intercal}$
    Newton step: $\widetilde{\theta}_{t+1} = \theta_t - \frac{1}{\gamma} A_t^{-1} \nabla_t$
    Projection: $(a_{t+1}^{\text{OPS}}, b_{t+1}^{\text{OPS}}) = \theta_{t+1} = \arg\min_{\theta \in \mathcal{K}} (\widetilde{\theta}_{t+1} - \theta)^{\intercal} A_t (\widetilde{\theta}_{t+1} - \theta)$
end for
```

6.A.4 Reproducibility

All results in this chapter can be reproduced exactly, including the randomization, using the IPython notebooks that can be found at https://github.com/aigen/df-posthoc-calibration in the folder `Online Platt scaling with Calibeating`. The README page in the folder contains a table describing which notebook to run to reproduce individual figures from this chapter.

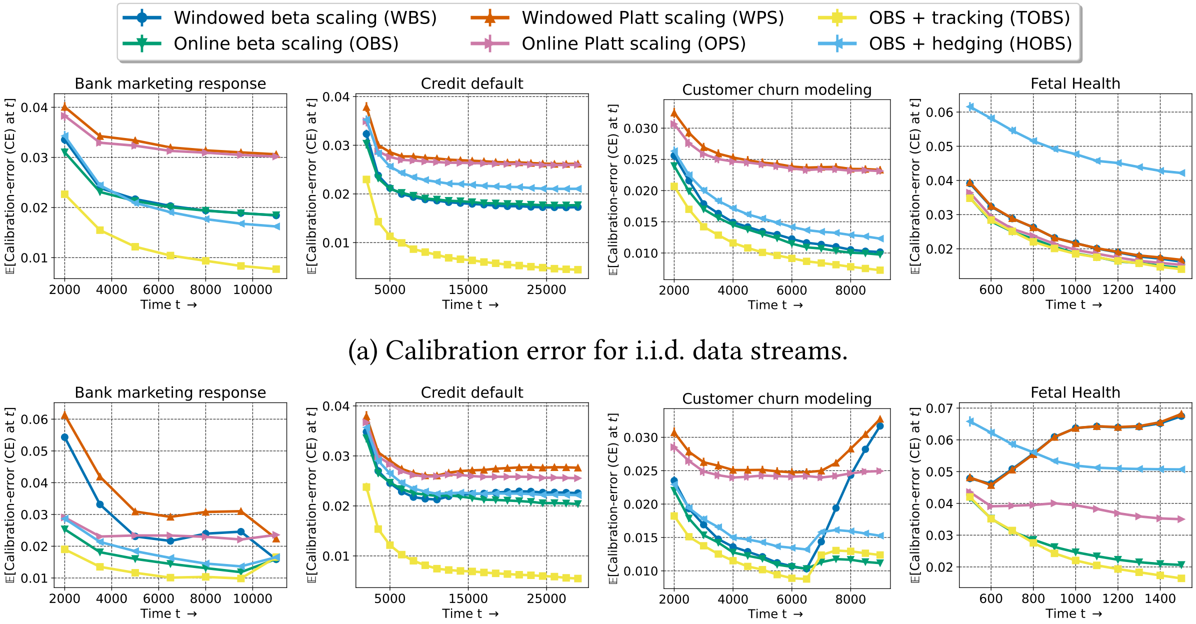
6.B Online beta scaling

This is an extended version of Section 6.5, with some repetition but more details. A recalibration method closely related to Platt scaling is beta scaling (Kull et al., 2017). The beta scaling mapping m has three parameters $(a, b, c) \in \mathbb{R}^3$, and corresponds to a sigmoid transform over two pseudo-features derived from $f(\mathbf{x})$: $\log(f(\mathbf{x}))$ and $\log(1 - f(\mathbf{x}))$:

$$m^{a,b,c}(f(\mathbf{x})) := \text{sigmoid}(a \cdot \log(f(\mathbf{x})) + b \cdot \log(1 - f(\mathbf{x})) + c). \tag{6.18}$$

Observe that enforcing $b = -a$ recovers Platt scaling since $\text{logit}(z) = \log(z) - \log(1 - z)$. The beta scaling parameters can be learnt following identical protocols as Platt scaling.

- The **traditional method** is to optimize parameters by minimizing the log-likelihood (equivalently, log-loss) over a fixed held-out batch of points.
- A **natural benchmark** for online settings is to update the parameters at some frequency (such as every 50 or 100 steps). At each update, the beta scaling parameters are set to the optimal value based on all data seen so far, and these parameters are used for prediction until the next update occurs. We call this benchmark windowed beta scaling (WBS); it is analogous to the windowed Platt scaling (WPS) benchmark considered in the main chapter.
- Our **proposed method** for online settings, called online Beta scaling (OBS), is to use a log-loss regret minimization procedure, similar to OPS. Analogously to (6.7), R_T for OBS



(a) Calibration error for i.i.d. data streams.

(b) Calibration error for drifting data streams.

Figure 6.15: Performance of online beta scaling (OBS) and its calibeating variants on real datasets with and without distribution drift. OBS further improves upon OPS in most cases. In each plot, TOBS is the best-performing method.

predictions $p_t^{\text{OBS}} = m^{a_t,b_t,c_t}(f(\mathbf{x}_t))$ is defined as

$$R_T(\text{OBS}) = \sum_{t=1}^{T} l(p_t^{\text{OBS}}, y_t) - \min_{(a,b,c)\in\mathcal{B}} \sum_{t=1}^{T} l(m^{a,b,c}(f(\mathbf{x}_t)), y_t), \tag{6.19}$$

where $\mathcal{B} := \{(a,b,c) \in \mathbb{R}^3 : a^2 + b^2 + c^2 \leqslant B^2\}$ for some $B \in \mathbb{R}$, and l is the log-loss. We use online Newton step (Algorithm 6.1) to learn (a_t, b_t, c_t), with the following initialization and hyperparameter values:

- $\mathcal{K} = \{(x,y,z) : \|(x,y,z)\|_2 \leqslant 100\}$, $(a_1^{\text{OBS}}, b_1^{\text{OBS}}, c_1^{\text{OBS}}) = (1, 1, 0)$;
- $\gamma = 0.1$, $\rho = 25$, $A_0 = \rho\mathbf{I}_3$.

These minor changes have to be made simply because the dimensionality changes from two to three. The empirical results we present shortly are based on an implementation with exactly these fixed hyperparameter values that do not change across the experiments (that is, we do not do any hyperparameter tuning).

Due to the additional degree of freedom, beta scaling is more expressive than Platt scaling. In the traditional batch setting, it was demonstrated by Kull et al. (2017) that this expressiveness typically leads to better (out-of-sample) calibration performance. We expect this relationship between Platt scaling and beta scaling to hold for their windowed and online versions as well. We confirm this intuition through an extension of the real dataset experiments of Section 6.4.1 to include WBS and OBS (Figure 6.15). In the main chapter we reported that the base model (BM) and fixed-batch Platt scaling model (FPS) perform the worst by a margin, so these lines are not reported again. We find that OBS performs better than both OPS and WBS, so we additionally report the performance of calibeating versions of OBS instead of OPS. That is, we replace OPS + tracking (TOPS) with OBS + tracking (TOBS), and OPS + hedging (HOPS) with OBS + hedging (HOBS).

A regret bound similar to Theorem 6.1 can be derived for OBS by instantiating ONS and AIOLI regret bounds with $d = 3$ (instead of $d = 2$ as done for OPS). The calibeating theorems (6.2 and 6.3) hold regardless of the underlying expert, and so also hold for OBS.

6.C F99 online calibration method

We describe the F99 method proposed by Foster (1999), and used in our implementation of HOPS (Section 6.3.3). The description is borrowed with some changes from Gupta and Ramdas (2022a). Recall that the F99 forecasts are the mid-points of the ϵ-bins (6.10): $B_1 = [0, \epsilon), B_2 = [\epsilon, 2\epsilon), \ldots, B_m = [1-\epsilon, 1]$. For $b \in [m] := \{1, 2, \ldots, m\}$ and $t \geqslant 1$, define:

$$\begin{aligned}
\text{(mid-point of } B_b) \quad & m_b = (b - 0.5)/m = b\epsilon - \epsilon/2, \\
\text{(left end-point of } B_b) \quad & l_b = (b-1)/m = (b-1)\epsilon, \\
\text{(right end-point of } B_b) \quad & r_b = b/m = b\epsilon,
\end{aligned}$$

F99 maintains some quantities as more data set is observed and forecasts are made. These are,

$$\begin{aligned}
\text{(frequency of forecasting } m_b) \quad & N_b^t = \left|\{\mathbb{1}\{p_s = m_b\} : s \leqslant t\}\right|, \\
\text{(observed average when } m_b \text{ was forecasted)} \quad & p_b^t = \begin{cases} \sum_{s=1}^t y_s \mathbb{1}\{p_s = m_b\} / N_b^t & \text{if } N_b^t > 0 \\ m_b & \text{if } N_b^t = 0, \end{cases} \\
\text{(deficit)} \quad & d_b^t = l_b - p_b^t, \\
\text{(excess)} \quad & e_b^t = p_b^t - r_b.
\end{aligned}$$

The terminology "deficit" is used to indicate that p_b^t is smaller l_b similarly. "Excess" is used to indicate that p_b^t is larger than r_b similarly. The F99 algorithm is as follows. Implicit in the description is computation of the quantities defined above.

F99: the online adversarial calibration method of Foster (1999)

- At time $t = 1$, forecast $p_1 = m_1$.
- At time $t + 1$ ($t \geqslant 1$), if

$$\text{condition A: there exists an } b \in [m] \text{ such that } d_b^t \leqslant 0 \text{ and } e_b^t \leqslant 0,$$

is satisfied, forecast $p_{t+1} = m_b$ for any i that verifies condition A. Otherwise,

$$\text{condition B: there exists a } b \in [m-1] \text{ such that } e_b^t > 0 \text{ and } d_{b+1}^t > 0,$$

must be satisfied (see Lemma 5 (Gupta and Ramdas, 2022a)). For any index b that satisfies condition B, forecast

$$p_{t+1} = \begin{cases} m_b & \text{with probability } \frac{d_{b+1}^t}{d_{b+1}^t + e_b^t} \\ m_{b+1} & \text{with probability } \frac{e_b^t}{d_{b+1}^t + e_b^t}. \end{cases}$$

These randomization probabilities are revealed before y_{t+1} is set by the agent that is generating outcomes, but the actual p_t value is drawn after y_{t+1} is revealed.

6.D Forecasting climatology to achieve calibration

Although Foster and Vohra's result (1998) guarantees that calibrated forecasting is possible against adversarial sequences, this does not immediately imply that the forecasts are useful in practice. To see this, consider an alternating outcome sequence, $y_t = \mathbb{1}\{t \text{ is odd }\}$. The forecast $p_t = \mathbb{1}\{t \text{ is odd }\}$ is calibrated and perfectly accurate. The forecast $p_t = 0.5$ (for every t) is also calibrated, but not very useful.

Thus we need to assess how a forecaster guaranteed to be calibrated for adversarial sequences performs on real-world sequences. In order to do so, we implemented the F99 forecaster (described in Appendix 6.C), on Pittsburgh's hourly rain data from January 1, 2008, to December 31, 2012. The data was obtained from ncdc.noaa.gov/cdo-web/. All days on which the hourly precipitation in inches (HPCP) was at least 0.01 were considered as instances of $y_t = 1$. There are many missing rows in the data, but no complex data cleaning was performed since we are mainly interested in a simple illustrative simulation. F99 makes forecasts on an ϵ-grid with $\epsilon = 0.1$: that is, the grid corresponds to the points $(0.05, 0.15, \ldots, 0.95)$. We observe (Figure 6.16) that after around 2000 instances, the forecaster *always* predicts 0.35. This is close to the average number of instances that it did rain which is approximately 0.37 (this long-term average is also called *climatology* in the meteorology literature). Although forecasting climatology can make the forecaster appear calibrated, it is arguably not a useful prediction given that there exist expert rain forecasters who can make sharp predictions for rain that change from day to day.

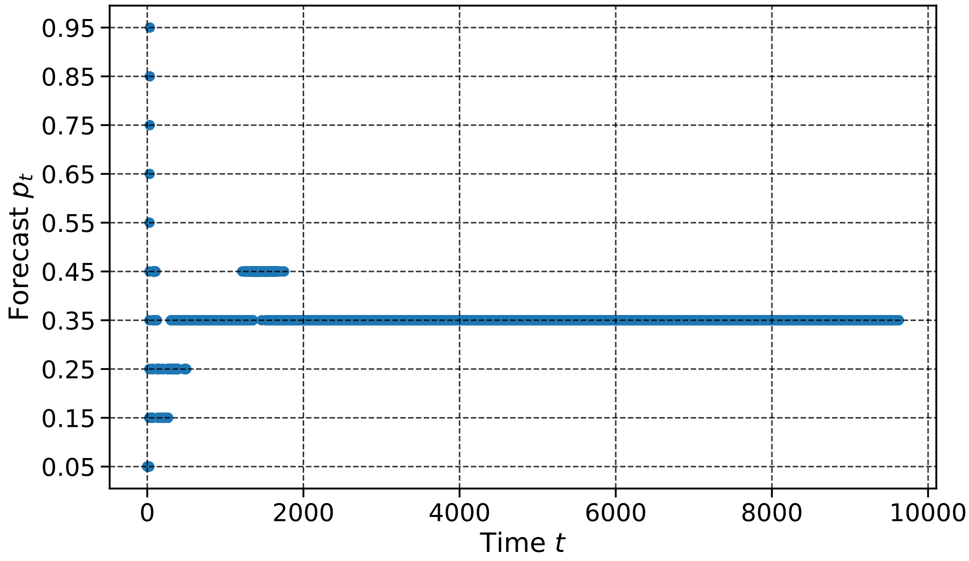


Figure 6.16: Foster (1999)'s ϵ-calibrated forecaster on Pittsburgh's hourly rain data (2008-2012). The forecaster makes predictions on the grid $(0.05, 0.15, \ldots, 0.95)$. In the long run, the forecaster starts predicting 0.35 for every instance, closely matching the average number of instances on which it rained (≈ 0.37).

6.E Proofs

Proof of Theorem 6.1. The regret bounds for ONS and AIOLI depend on a few problem-dependent parameters.

- The dimension $d = 2$.
- The radius of the reference class B.
- Bound on the norm of the gradient, which for logistic regression is also the radius of the space of input vectors. Due to the assumption on $f(\mathbf{x}_t)$, the norm of the input is at most $\sqrt{\text{logit}(0.01)^2 + 1^2} = \sqrt{\text{logit}(0.99)^2 + 1^2} \leqslant 5$.

The AIOLI bound (6.9) follows from Theorem 1, equation (4) of Jézéquel et al. (2020), setting $d = 2$ and $R = 10$.

The ONS bound (6.8) follows from Theorem 4.5 of Hazan (2016), plugging in $G = 5$, $D = 2B$, and $\alpha = e^{-B}$ which is the known exp-concavity constant of the logistic loss over a ball of radius B (Foster et al., 2018).

□

In writing the proofs of the results in Section 6.3, we will use an object closely connected to sharpness called refinement. For a sequence of forecasts $p_{1:T}$ and outcome sequence $y_{1:T}$, the refinement $\mathcal{R}$ is defined as

$$\mathcal{R}(p_{1:T}) := \frac{1}{T}\sum_{b=1}^{m} N_b \cdot \widehat{y}_b(1-\widehat{y}_b), \tag{6.20}$$

where $\widehat{y}_b$ is the average of the outcomes in every ϵ-bin; see the beginning of Section 6.3.2 where sharpness is defined. The function $x(\in[0,1]) \mapsto x(1-x)$ is minimized at the boundary points $\{0,1\}$ and maximized at $1/2$. Thus refinement is lower if $\widehat{y}_b$ is close to 0 or 1, or in other words if the bins discriminate points well. This is captured formally in the following (well-known) relationship between refinement and sharpness.

Lemma 6.1 (Sharpness-refinement lemma)**.** *For any forecast sequence $p_{1:T}$, the refinement $\mathcal{R}$ defined in (6.20) and the sharpness SHP defined in (6.12) are related as:*

$$\mathcal{R}(p_{1:T}) = \bar{y}_T - \mathit{SHP}(p_{1:T}),$$

where $\bar{y}_T = \frac{1}{T}\sum_{t=1}^{T} y_t$.

Proof. Observe that

$$\mathcal{R}(p_{1:T}) = \frac{1}{T}\sum_{b=1}^{B} N_b\widehat{y}_b - \frac{1}{T}\sum_{b=1}^{B} N_b\widehat{y}_b^2 = \frac{1}{T}\sum_{b=1}^{B} N_b\widehat{y}_b - \text{SHP}(p_{1:T}).$$

The final result follows simply by noting that

$$\sum_{b=1}^{B} N_b\widehat{y}_b = \sum_{b=1}^{B}\left(\sum_{t\leqslant T, p_t\in B_b} y_t\right) = \sum_{t=1}^{T} y_t.$$

□

We now state a second lemma, that relates $\mathcal{R}$ to the Brier-score $\mathcal{BS}$ defined as

$$\mathcal{BS}(p_{1:T}) := \frac{\sum_{t=1}^{T}(y_t - p_t)^2}{T}. \tag{6.21}$$

Unlike $\mathcal{R}$ and SHP, $\mathcal{BS}$ is not defined after ϵ-binning. It is well-known (see for example equation (1) of FH23) that if refinement is defined without ϵ-binning (or if the Brier-score is defined with ϵ-binning), then refinement is at most the Brier-score defined above. Since we define $\mathcal{R}$ defined with binning, further work is required to relate the two.

Lemma 6.2 (Brier-score-refinement lemma)**.** *For any forecast sequence $p_{1:T}$ and outcome sequence $y_{1:T}$, the refinement $\mathcal{R}$ and the Brier-score $\mathcal{BS}$ are related as*

$$\mathcal{R}(p_{1:T}) \leqslant \mathcal{BS}(p_{1:T}) + \frac{\epsilon^2}{4} + \epsilon, \tag{6.22}$$

where ϵ is the width of the bins used to define $\mathcal{R}$ (6.10).

Proof. Define the discretization function disc : $[0, 1] \to [0, 1]$ as $\text{disc}(p) = \text{mid-point}(B_b) \iff p \in B_b$. Note that for all $p \in [0, 1]$, $|p - \text{disc}(p)| \leqslant \epsilon/2$. Based on standard decompositions (such as equation (1) of FH23), we know that

$$\mathcal{R}(p_{1:T}) \leqslant \frac{\sum_{t=1}^{T}(y_t - \text{disc}(p_t^{\text{TOPS}}))^2}{T}. \tag{6.23}$$

We now relate the RHS of the above equation to $\mathcal{BS}$

$$\begin{aligned}
\sum_{t=1}^{T}(y_t - \text{disc}(p_t))^2 &= \sum_{t=1}^{T}(y_t - p_t + p_t - \text{disc}(p_t))^2 \\
&= T \cdot \mathcal{BS}(p_{1:T}) + \sum_{t=1}^{T}(p_t - \text{disc}(p_t))^2 + 2\sum_{t=1}^{T}(y_t - p_t)(p_t - \text{disc}(p_t)) \\
&\leqslant T \cdot \mathcal{BS}(p_{1:T}) + T(\epsilon/2)^2 + 2\sum_{t=1}^{T}|y_t - p_t|\,(\epsilon/2). \\
&\leqslant T \cdot \mathcal{BS}(p_{1:T}) + T(\epsilon/2)^2 + T\epsilon.
\end{aligned}$$

The result of the theorem follows by dividing by T on both sides. □

Proof of Theorem 6.2. The calibeating paper (Foster and Hart, 2023) is referred to as FH23 in this proof for succinctness.

We use Theorem 3 of FH23, specifically equation (13), which gives an upper bound on the Brier-score of a tracking forecast ($\mathcal{B}_t^{\mathbf{c}}$ in their notation) relative to the refinement (6.20) of the base forecast. In our case, the tracking forecast is TOPS, the base forecast is OPS, and FH23's result gives,

$$\mathcal{BS}(p_{1:T}^{\text{TOPS}}) = \frac{\sum_{t=1}^{T}(y_t - p_t^{\text{TOPS}})^2}{T} \leqslant \mathcal{R}(p_{1:T}^{\text{TOPS}}) + \frac{\log T + 1}{\epsilon T}. \tag{6.24}$$

Using the Brier-score-refinement lemma 6.2 to lower bound $\mathcal{BS}(p_{1:T}^{\text{TOPS}})$ gives

$$\mathcal{R}(p_{1:T}^{\text{TOPS}}) - \frac{\epsilon^2}{4} - \epsilon \leqslant \mathcal{R}(p_{1:T}^{\text{OPS}}) + \frac{\log T + 1}{\epsilon T}. \tag{6.25}$$

Finally, using the sharpness-refinement lemma 6.1, we can replace each $\mathcal{R}$ with $\bar{y}_T - \text{SHP}$. Rearranging terms gives the final bound.

□

Proof of Theorem 6.3. The calibeating paper (Foster and Hart, 2023) is referred to as FH23 in this proof for succinctness.

Sharpness bound (6.16). Theorem 5 of FH23 showed that the expected Brier-score for a different hedging scheme (instead of F99), is at most the expected refinement score of the base forecast plus $\epsilon^2 + \frac{\log T+1}{\epsilon^2 T}$. In our case, the second term remains unchanged, but because we use F99, the ϵ^2 needs to be replaced, and we show that it can be replaced by ϵ next.

Let us call the combination of OPS and the FH23 hedging method as FH23-HOPS, and the calibeating forecast as $p_{1:T}^{\text{FH23-HOPS}}$. The source of the ϵ^2 term in Theorem 5 of FH23 is the following property of FH23-HOPS: for both values of $y_t \in \{0, 1\}$,

$$\mathbb{E}_{t-1}\left[(y_t - p_t^{\text{FH23-HOPS}})^2 - (y_t - \text{Average}\{y_s : s < t, p_s^{\text{OPS}} = p_t^{\text{OPS}}, p_s^{\text{FH23-HOPS}} = p_t^{\text{FH23-HOPS}}\})^2\right] \leqslant \epsilon^2,$$

where $\mathbb{E}_{t-1}[\cdot]$ is the expectation conditional on $(y_{1:t-1}, p_{1:t-1}^{\text{FH23-hedging}}, p_{1:t-1}^{\text{OPS}})$ (all that's happened in the past, and the current OPS forecast). For HOPS, we will show that

$$Q_t := \mathbb{E}_{t-1}\left[(y_t - p_t^{\text{HOPS}})^2 - (y_t - \text{Average}\{y_s : s < t, p_s^{\text{OPS}} = p_t^{\text{OPS}}, p_s^{\text{HOPS}} = p_t^{\text{HOPS}}\})^2\right] \leqslant \epsilon,$$

for $y_t \in \{0, 1\}$, which would give the required result.

At time t, the F99 forecast falls into one of two scenarios which we analyze separately (see Appendix 6.C for details of F99 which would help follow the case-work).

- **Case 1.** This corresponds to condition A in the description of F99 in Section 6.C. There exists a bin index b such that $q = \text{mid-point}(B_b)$ satisfies

  $$\left|\text{Average}\{y_s : s < t, p_s^{\text{OPS}} = p_t^{\text{OPS}}, p_s^{\text{HOPS}} = q\} - q\right| \leqslant \epsilon/2.$$

  In this case, F99 would set $p_t^{\text{HOPS}} = q$ (deterministically) for some q satisfying the above. Thus,

  $$\begin{aligned} Q_t &= (y_t - q)^2 - (y_t - \text{Average}\{y_s : s < t, p_s^{\text{OPS}} = p_t^{\text{OPS}}, p_s^{\text{HOPS}} = q\})^2 \\ &\leqslant \max((y_t - q)^2 - (y_t - q - \epsilon/2)^2, (y_t - q)^2 - (y_t - q + \epsilon/2)^2) \\ &\leqslant (\epsilon/2)(2\left|y_t - q\right| + \epsilon/2) < \epsilon, \end{aligned}$$

  irrespective of y_t, since $q \in [\epsilon/2, 1 - \epsilon/2]$.

- **Case 2.** This corresponds to condition B in the description of F99 in Section 6.C. If Case 1 does not hold, F99 randomizes between two consecutive bin mid-points $m - \epsilon/2$ and $m - \epsilon/2$, where m is one of the edges of the ϵ-bins (6.10). Define $n_1 := \text{Average}\{y_s : s < t, p_s^{\text{OPS}} = p_t^{\text{OPS}}, p_s^{\text{HOPS}} = m - \epsilon/2\}$ and $n_2 := \text{Average}\{y_s : s < t, p_s^{\text{OPS}} = p_t^{\text{OPS}}, p_s^{\text{HOPS}} = m + \epsilon/2\}$. The choice of m in F99 guarantees that $n_2 < m < n_1$, and the randomization probabilities are given by

  $$\mathbb{P}_{t-1}(p_t^{\text{HOPS}} = m - \epsilon/2) = \frac{m - n_2}{n_1 - n_2}, \text{ and } \mathbb{P}_{t-1}(p_t^{\text{HOPS}} = m + \epsilon/2) = \frac{n_1 - m}{n_1 - n_2},$$

  where $\mathbb{P}_{t-1}$ is the conditional probability in the same sense as $\mathbb{E}_{t-1}$. We now bound Q_t. If $y_t = 1$,

  $$\begin{aligned} Q_t &= \mathbb{E}_{t-1}\left[(y_t - p_t^{\text{HOPS}})^2 - (y_t - \text{Average}\{y_s : s < t, p_s^{\text{OPS}} = p_t^{\text{OPS}}, p_s^{\text{HOPS}} = p_t^{\text{HOPS}}\})^2\right] \\ &= \frac{m - n_2}{n_1 - n_2}\left((1 - (m - \epsilon/2))^2 - (1 - n_1)^2\right) + \frac{n_1 - m}{n_1 - n_2}\left((1 - (m + \epsilon/2))^2 - (1 - n_2)^2\right) \\ &= \underbrace{\frac{m - n_2}{n_1 - n_2}\left((1 - m)^2 - (1 - n_1)^2\right) + \frac{n_1 - m}{n_1 - n_2}\left((1 - m)^2 - (1 - n_2)^2\right)}_{=:A_1} \end{aligned}$$

$$+ \underbrace{2 \cdot (\epsilon/2) \cdot \frac{(m - n_2)(1 - m) - (n_1 - m)(1 - m)}{n_1 - n_2}}_{=:A_2}$$

$$+ (\epsilon/2)^2 \cdot \frac{n_1 - n_2}{n_1 - n_2}.$$

A_1 and A_2 simplify as follows.

$$\begin{aligned} A_1 &= \frac{(m - n_2)(n_1 - m)(2 - (n_1 + m)) + (n_1 - m)(n_2 - m)(2 - (n_2 + m))}{n_1 - n_2} \\ &= \frac{(m - n_2)(n_1 - m)(n_2 - n_1)}{n_1 - n_2} < 0, \end{aligned}$$

since $n_2 < m < n_1$.

$$\begin{aligned} A_2 &= \epsilon \cdot \frac{(m - n_2)(1 - m)}{n_1 - n_2} + \epsilon \cdot \frac{(m - n_1)(1 - m)}{n_1 - n_2} \\ &< \epsilon \cdot \frac{(m - n_2)(1 - m)}{n_1 - n_2} && \text{(since } m < n_1\text{)} \\ &< \epsilon(1 - m). \end{aligned}$$

Overall, we obtain that for $y_t = 1$,

$$Q_t < \epsilon(1 - m) + (\epsilon^2/4) < \epsilon,$$

where the final inequality holds since m is an end-point between two bins, and thus $m \geqslant \epsilon$. We do the calculations for $y_t = 0$ less explicitly since it essentially follows the same steps:

$$\begin{aligned} Q_t &= \mathbb{E}_{t-1}\left[(0 - p_t^{\text{HOPS}})^2 - (0 - \text{Average}\{y_s : s < t, p_s^{\text{OPS}} = p_t^{\text{OPS}}, p_s^{\text{HOPS}} = p_t^{\text{HOPS}}\})^2\right] \\ &= \frac{m - n_2}{n_1 - n_2}\left((m - \epsilon/2)^2 - n_1^2\right) + \frac{n_1 - m}{n_1 - n_2}\left((m + \epsilon/2)^2 - n_2^2\right) \\ &= \frac{(m - n_2)(m - n_1)(m + n_1) + (n_1 - m)(m - n_2)(m + n_2)}{n_1 - n_2} \\ &\qquad + \epsilon \cdot \frac{(n_2 - m)m + (n_1 - m)m}{n_1 - n_2} + \frac{\epsilon^2}{4} \\ &< 0 + \epsilon m + (\epsilon^2/4) < \epsilon. \end{aligned}$$

Finally, by Proposition 1 of FH23 and the above bound on Q_t, we obtain,

$$\mathbb{E}\left[\mathcal{R}(p_{1:T}^{\text{HOPS}})\right] \leqslant \mathbb{E}\left[\mathcal{BS}(p_{1:T}^{\text{HOPS}})\right] \leqslant \epsilon + \mathcal{R}(p_{1:T}^{\text{OPS}}) + \frac{\log T + 1}{\epsilon^2 T}. \tag{6.26}$$

Using the sharpness-refinement lemma 6.1, we replace each $\mathcal{R}$ with $\bar{y}_T - \text{SHP}$. Rearranging terms gives the sharpness result.

Calibration bound (6.17)**.** Recall that the number of bins is $m = 1/\epsilon$. For some bin indices $b, b' \in \{1, 2, \ldots, m\}$, let $S_{b \to b'} = \{t \leqslant T : p_t^{\text{OPS}} \in B_b, p_t^{\text{HOPS}} = \text{mid-point}(B_{b'})\}$ be the set of time instances at which the OPS forecast p_t^{OPS} belonged to bin b, but the HOPS forecast p_t^{HOPS} belonged to bin b' (and equals the mid-point of bin b'). Also, let $S_b = \{t \leqslant T : p_t^{\text{OPS}} \in B_b\}$ be the set of time instances at which the p_t^{OPS} forecast belonged to bin b. Thus $S_b = \bigcup_{b'=1}^{m} S_{b \to b'}$. Also define $N_b^{\text{OPS}} = |S_b|$ and $N_b^{\text{HOPS}} = |\{t \leqslant T : p_t^{\text{HOPS}} = \text{mid-point}(B_b)\}|$.

Now for any specific b, consider the sequence $(y_t)_{t \in S_b}$. On this sequence, the HOPS forecasts correspond to F99 using just the outcomes (with no regard for covariate values once the bin of p_t^{OPS} is fixed). Thus, within this particular bin, we have a usual CE guarantee that F99's algorithm has for any arbitrary sequence:

$$\underbrace{\mathbb{E}\left[\frac{1}{N_b^{\text{OPS}}} \sum_{b'=1}^{m} \left| \sum_{t \in S_{b \to b'}} (y_t - p_t^{\text{HOPS}}) \right|\right]}_{\text{this is the expected CE over the } S_b \text{ instances}} \leqslant \frac{\epsilon}{2} + \frac{2}{\sqrt{\epsilon \cdot N_b^{\text{OPS}}}}. \tag{6.27}$$

This result is unavailable in exactly this form in Foster (1999) which just gives the reduction to Blackwell approachability, after which any finite-sample approachability bound can be used. The above version follows from Theorem 1.1 of Perchet (2014). The precise details of the Blackwell approachability set, reward vectors, and how the distance to the set can be translated to CE can be found in Gupta and Ramdas (2022a, Section 4.1).

Jensen's inequality can be used to lift this CE guarantee to the entire sequence:

$$\begin{aligned}
\mathbb{E}\left[\text{CE}(p_{1:T}^{\text{HOPS}})\right] &= \mathbb{E}\left[\sum_{b=1}^{m} \frac{|N_b^{\text{HOPS}}(\widehat{y}_b^{\text{HOPS}} - \widehat{p}_b^{\text{HOPS}})|}{T}\right] \\
&= \mathbb{E}\left[\sum_{b=1}^{m} \frac{\left|\sum_{t=1}^{T} (y_t - p_t^{\text{HOPS}}) \mathbb{1}\left\{p_t^{\text{HOPS}} \in B_b\right\}\right|}{T}\right] \\
&= \mathbb{E}\left[\sum_{b=1}^{m} \frac{\left|\sum_{b'=1}^{m} \sum_{t \in S_{b' \to b}} (y_t - p_t^{\text{HOPS}})\right|}{T}\right] \\
&\leqslant \mathbb{E}\left[\sum_{b=1}^{m} \sum_{b'=1}^{m} \frac{\left|\sum_{t \in S_{b' \to b}} (y_t - p_t^{\text{HOPS}})\right|}{T}\right] && \text{(Jensen's inequality)} \\
&= \sum_{b'=1}^{m} \mathbb{E}\left[\sum_{b=1}^{m} \frac{\left|\sum_{t \in S_{b' \to b}} (y_t - p_t^{\text{HOPS}})\right|}{T}\right] \\
&\leqslant \sum_{b'=1}^{m} \frac{N_{b'}^{\text{OPS}} \left(\epsilon/2 + 2/\sqrt{\epsilon \cdot N_{b'}^{\text{OPS}}}\right)}{T} && \text{(by (6.27))} \\
&= \frac{\epsilon}{2} + \frac{2}{\sqrt{\epsilon}} \cdot \frac{\sum_{b'=1}^{m} \sqrt{N_{b'}^{\text{OPS}}}}{\sum_{b'=1}^{m} N_{b'}^{\text{OPS}}} && \left(\text{since } T = \sum_{b'=1}^{B} N_{b'}^{\text{OPS}}\right)
\end{aligned}$$

$$\overset{(\star)}{\leqslant} \frac{\epsilon}{2} + \frac{2}{\sqrt{\epsilon}} \cdot \sqrt{\frac{m}{T}} = \frac{\epsilon}{2} + 2\sqrt{\frac{1}{\epsilon^2 T}},$$

as needed to be shown. The inequality $(\star)$ holds because, by Jensen's inequality (or AM-QM inequality),

$$\sqrt{\frac{\sum_{b'=1}^{m} N_{b'}^{\text{OPS}}}{m}} \geqslant \frac{\sum_{b'=1}^{m} \sqrt{N_{b'}^{\text{OPS}}}}{m},$$

so that

$$\frac{\sum_{b'=1}^{m} \sqrt{N_{b'}^{\text{OPS}}}}{\sum_{b'=1}^{m} N_{b'}^{\text{OPS}}} = \frac{\sum_{b'=1}^{m} \sqrt{N_{b'}^{\text{OPS}}}}{\sqrt{\sum_{b'=1}^{m} N_{b'}^{\text{OPS}}}} \cdot \frac{1}{\sqrt{\sum_{b'=1}^{m} N_{b'}^{\text{OPS}}}} \leqslant \frac{\sqrt{m}}{\sqrt{\sum_{b'=1}^{m} N_{b'}^{\text{OPS}}}} = \sqrt{m/T}.$$

□

Theorem 6.4. *For adversarially generated data, the expected Brier-score of HOPS forecasts using the forecast hedging algorithm of Foster (1999) is upper bounded as*

$$\mathbb{E}\left[\mathcal{BS}(p_{1:T}^{HOPS})\right] \leqslant \mathcal{BS}(p_{1:T}^{OPS}) + \left(2\epsilon + \frac{\epsilon^2}{4} + \frac{\log T + 1}{\epsilon^2 T}\right). \tag{6.28}$$

Proof. In the proof of the sharpness result of Theorem 6.3, we showed equation (6.26), which immediately yields (6.28) since $\mathcal{R}(p_{1:T}^{\text{OPS}}) \leqslant \mathcal{BS}(p_{1:T}^{\text{OPS}}) + \epsilon + \epsilon^2/4$ by Lemma 6.2. □

Chapter 7

Parity Calibration

This chapter is based on Chung et al. (2023).

In a sequential regression setting, a decision-maker may be primarily concerned with whether the future observation will increase or decrease compared to the current one, rather than the actual value of the future observation. In this context, we introduce the notion of parity calibration, which captures the goal of calibrated forecasting for the increase-decrease (or "parity") event in a timeseries. Parity probabilities can be extracted from a forecasted distribution for the output, but we show that such a strategy leads to theoretical unpredictability and poor practical performance. We then observe that although the original task was regression, parity calibration can be expressed as binary calibration. Drawing on this connection, we use an online binary calibration method to achieve parity calibration. We demonstrate the effectiveness of our approach on real-world case studies in epidemiology, weather forecasting, and model-based control in nuclear fusion.

7.1 Introduction

Many tasks in the scope of prediction and decision making are sequential in nature. A weather forecaster who uses some procedure to make predictions for tomorrow, may find that tomorrow's events falsify these predictions. A good forecaster must then update their model before using it on the following days. In this work, we study the sequential forecasting setting where the goal is to make predictions about a sequence of real-valued outcomes $y_1, y_2, \ldots \in \mathcal{Y} \subseteq \mathbb{R}$ using informative covariates $\mathbf{x}_1, \mathbf{x}_2, \ldots \in \mathcal{X}$. In the presence of inherent stochasticity or insufficient data, forecasters who provide rich predictions in the form of complete distributions over the output allow us to reason about the inherent uncertainties in the data stream (Gneiting et al., 2007). If a distributional prediction is available, a downstream decision-maker can account for risks that were unknown at the time of forecasting.

Often, a distributional forecast for the real-valued y_t takes the form of a predictive cdf (cumulative distribution function) for y_t, which in this chapter we typically denote as $\hat{F}_t : \mathcal{Y} \to [0, 1]$. We sometimes write $\hat{F}_t$ as $\hat{F}_t(\cdot|\mathbf{x}_t)$ or $\hat{F}_t(\cdot|\mathbf{x}_t, y_{t-1}, \mathbf{x}_{t-1}, \ldots, y_1, \mathbf{x}_1)$; this overloaded notation allows us to be succinct when defining what it means for $\hat{F}_t$ to be calibrated, but explicit when it is necessary to stress that $\hat{F}_t$ depends on all available knowledge. We also refer to $\hat{F}_t$'s as regression forecasts, as it models a continuous distribution over the real-valued output.

In this work, we are interested in the question: can we forecast whether the future outcome y_{t+1} will be greater or less than the current outcome y_t? To motivate this question, consider a hospital in the midst of a fast moving pandemic such as COVID-19. It may be difficult for the hospital to comprehend absolute numbers of patients requiring hospitalization. However, relative numbers are perhaps easier to interpret: hospitals know the situation today, and would like to know if it is going to worsen or improve tomorrow.

A domain expert (e.g. epidemiologist) may have produced a regression forecast $\hat{F}_t$ for y_t. The downstream user (e.g. hospital) can then extract from $\hat{F}_t$ a natural implied probability of the next observation decreasing:

$$\text{for } t \geqslant 2,\ \hat{p}_t = \hat{F}_t(y_{t-1} \mid \mathbf{x}_t). \tag{7.1}$$

The hope of the hospital is that the forecasted probabilities $\hat{p}_t$ are parity calibrated, as defined next.

Definition 7.1 (Parity calibration)**.** The forecasts $\{\hat{p}_t \in [0,1]\}_{t=2,\ldots,T}$ are said to be parity calibrated if

$$\frac{\sum_{t=2}^{T} \mathbb{1}\{y_t \leqslant y_{t-1}\}\, \mathbb{1}\{\hat{p}_t = p\}}{\sum_{t=2}^{T} \mathbb{1}\{\hat{p}_t = p\}} \to p, \forall p \in [0,1]. \tag{7.2}$$

In words, whenever a parity calibrated forecaster predicts with probability p that $y_t \leqslant y_{t-1}$, the event $\mathbb{1}\{y_t \leqslant y_{t-1}\}$ actually occurs with empirical frequency p (in the long run). To avoid confusion with usage of the term "parity" in fairness literature, we remark that our context is purely in comparing two consecutive values.

Our first contribution is showing that even if $\hat{F}_t$ is calibrated (based on some accepted notions of calibration), the seemingly reasonable strategy mentioned above (7.1) can have devastating and unpredictable behavior (Section 7.1.1). Yet, it stands to reason that the expert's rich forecast $\hat{F}_t$ should be used in some way. Our second contribution is a methodology for doing this (Sections 7.2 and 7.3). Our main methodology described in Section 7.2.2 is based on the key observation that although the parity calibration problem is derived from a regression problem, it naturally reduces to a problem of forecasting binary events.

7.1.1 Regression calibration does not give parity calibration

A popular notion of calibration in regression is *probabilistic calibration* (Gneiting et al., 2007). The sequence $\hat{F}_1, \hat{F}_2, \ldots$ is said to be probabilistically calibrated if

$$\frac{1}{T}\sum_{t=1}^{T} F_t(\hat{F}_t^{-1}(p)) \to p,\ \forall p \in [0,1], \tag{7.3}$$

where F_t denotes the ground truth distribution. Probabilistic calibration is also referred to as *quantile calibration*, since it focuses on the quantile function being valid. In other works, it has also been referred to as average calibration (Zhao et al., 2020; Chung et al., 2021b; Sahoo et al., 2021), or simply calibration (Kuleshov et al., 2018; Cui et al., 2020; Charpentier et al., 2022; Marx et al., 2022). We will henceforth refer to this notion as *quantile calibration.*

Another notion of calibration in regression is *distributional calibration* (Song et al., 2019), which assesses the convergence of the full distribution of the observations to the predictive distribution. A distribution calibrated forecaster satisfies $\forall p \in [0,1],\ \forall F \in \mathcal{F}$,

$$\frac{\sum_{t=1}^{T} \mathbb{1}\left\{\hat{F}_t = F\right\} F_t(\hat{F}_t^{-1}(p))}{\sum_{t=1}^{T} \mathbb{1}\left\{\hat{F}_t = F\right\}} \to p, \tag{7.4}$$

where $\mathcal{F}$ is the space of distributions predicted by $\hat{F}_t$. However, distributional calibration is an idealistic notion that cannot be achieved in practice (Song et al., 2019).

Recently, Sahoo et al. (2021) paired calibration with the notion of threshold decisions and proposed *threshold calibration.* Forecasts are said to be threshold calibrated if,

$$\frac{\sum_{t=1}^{T} \mathbb{1}\left\{\hat{F}_t(y_0) \leqslant \alpha\right\} F_t(\hat{F}_t^{-1}(p))}{\sum_{t=1}^{T} \mathbb{1}\left\{\hat{F}_t(y_0) \leqslant \alpha\right\}} \to p,\ \forall y_0 \in \mathcal{Y},\ \forall \alpha \in [0,1], \forall p \in [0,1].$$

Sahoo et al. (2021) show that distribution calibration implies threshold calibration, but the converse may not hold.

A common aspect of the aforementioned notions of calibration is that they all assess how well-aligned the predictive quantiles are to their empirical counterparts. The key difference among the notions is the conditioning over which this assessment is performed.

Since calibration is regarded as a desirable quality of distributional forecasts, one may wonder whether a calibrated $\hat{F}_t$ is sufficient for parity calibration of the implied probabilities as per Eq. (7.1). We show that this is *not* the case with the following examples.

Synthetic example. Let $\mathcal{N}_-$ and $\mathcal{N}_+$ denote the standard normal distributions truncated at 0, with density functions $f_-(x) = \mathbb{1}\{x < 0\}\sqrt{2/\pi}e^{-x^2/2}$ and $f_+(x) = \mathbb{1}\{x \geqslant 0\}\sqrt{2/\pi}e^{-x^2/2}$ respectively. Let F_- and F_+ be the cdfs of $\mathcal{N}_-$ and $\mathcal{N}_+$. Suppose the target sequence $(Y_t)_{t=1}^{\infty}$ is distributed as

$$Y_t \sim \begin{cases} \mathcal{N}_- & \text{if } t \text{ is odd,} \\ \mathcal{N}_+ & \text{if } t \text{ is even.} \end{cases}$$

Consider the following predictive cdf targeting Y_t,

$$\hat{F}_t = \frac{1}{2}F_- + \frac{1}{2}F_+ = \begin{cases} \frac{1}{2}F_-(y), \text{ if } y < 0, \\ 0.5 + \frac{1}{2}F_+(y), \text{ if } y \geqslant 0. \end{cases}.$$

We note that when $y < 0$, $\frac{1}{2}F_-(y) \in [0, 0.5)$, and when $y \geqslant 0.5$, $0.5 + \frac{1}{2}F_+(y) \in [0.5, 1]$. It can be verified that the corresponding quantile function is

$$\hat{F}_t^{-1}(p) = \begin{cases} F_-^{-1}(2p), \text{ if } p < 0.5 \\ F_+^{-1}(2p-1), \text{ if } p \geqslant 0.5. \end{cases}$$

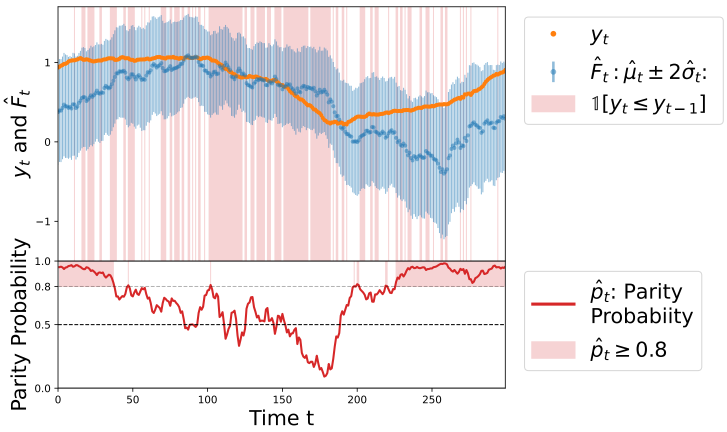


Figure 7.1: Snapshot of the first 300 points from one of our experiment datasets (Pressure from Section 7.3.2) shows a quantile calibrated forecaster that is highly parity miscalibrated. (**top**) The expert forecasts $\hat{F}_t$ are Gaussians, expressed in the plot as prediction intervals $[\hat{\mu}_t - 2\hat{\sigma}_t, \hat{\mu}_t + 2\hat{\sigma}_t]$. This prediction interval almost always contains y_t and its reliability diagram in Figure 7.4 (plot titled "Quantile Calibration") confirms that $\hat{F}_t$ is in fact quantile calibrated when considering the full timeseries. (**bottom**) For $t \in [0, 40]$ and $t \in [230, 300]$, the parity probabilities $\hat{p}_t = \hat{F}_t(y_{t-1})$ assign $\geqslant 0.8$ probability (red shaded areas) to $\mathbb{1}\{y_t \leqslant y_{t-1}\}$. But y_t actually decreases with much lower frequency during these timesteps as can be seen from the top figure. The parity miscalibration when considering the full timeseries is confirmed by Figure 7.4 (plot titled "Prehoc").

We verify that $\hat{F}_t$ is quantile calibrated (following Eq. (7.3)).

<u>When t is odd</u>, $F_t = F_-$.

- $\forall p \in [0, 0.5)$, $F_t(\hat{F}_t^{-1}(p)) = F_-(F_-^{-1}(2p)) = 2p$.
- $\forall p \in [0.5, 1]$, $\hat{F}_t^{-1}(p) = F_+^{-1}(2p - 1) \geqslant 0$, thus $F_t(\hat{F}_t^{-1}(p)) = F_-(F_+^{-1}(2p - 1)) = 1$.

<u>When t is even</u>, $F_t = F_+$.

- $\forall p \in [0, 0.5)$, $\hat{F}_t^{-1}(p) = F_-^{-1}(2p) < 0$, thus $F_t(\hat{F}_t^{-1}(p)) = F_+(F_-^{-1}(2p)) = 0$.
- $\forall p \in [0.5, 1]$, $F_t(\hat{F}_t^{-1}(p)) = F_+(F_+^{-1}(2p - 1)) = 2p - 1$.

Therefore, for $p \in [0, 0.5)$, $\frac{1}{T}\sum_{t=1}^{T} F_t(\hat{F}_t^{-1}(p)) = \frac{1}{T}\sum_{t \text{ is odd}} 2p = p + o(\frac{1}{T}) \to p$, and the same can be verified for $p \in [0.5, 1]$, showing that $\hat{F}_t$ *is quantile calibrated*.

We can easily show that $\hat{F}_t$ is also distribution and threshold calibrated. Since $\hat{F}_t$ is constant for all t, following Eq. (7.4), the space of predicted distributions is a singleton. Thus, measuring distribution calibration is equivalent to measuring quantile calibration, and $\hat{F}_t$ *is distribution calibrated*. Since distribution calibration implies threshold calibration (Sahoo et al., 2021), $\hat{F}_t$ *is threshold calibrated*.

However, as we show next, $\hat{F}_t$ is not parity calibrated.

<u>When t is odd</u>, $Y_t \sim F_-$ and $Y_{t-1} \sim F_+$. Thus $Y_t < Y_{t-1}$ whereas $\hat{p}_t = \hat{F}_t(Y_{t-1}) \geqslant 0.5$.

When t is even, $Y_t \sim F_+$ and $Y_{t-1} \sim F_-$. Thus $Y_t > Y_{t-1}$ whereas $\hat{p}_t = \hat{F}_t(Y_{t-1}) < 0.5$.

Therefore, $\forall \hat{p}_t \geqslant 0.5$, $\mathbb{1}\left\{y_t \leqslant y_{t-1}\right\} = 1$ and $\forall \hat{p}_t < 0.5$, $\mathbb{1}\left\{y_t \leqslant y_{t-1}\right\} = 0$, thus $\hat{F}_t$ *is parity miscalibrated* for all $\hat{p}_t \in (0,1)$, i.e. all $\hat{p}_t \neq 0$ or 1. □

Intuitively, the sequential aspect of predictions and observations is central to the notion of parity calibration, whereas traditional notions of calibration effectively treat the datapoints as an i.i.d. or exchangeable batch of points. Figure 7.1 provides a visualization of how this pitfall can be manifested in a practical example.

The implication is that methods designed to achieve traditional notions of calibration in regression cannot be expected to provide parity calibration. The following section introduces the post-hoc binary calibration framework that can instead be used to achieve parity calibrated forecasts.

7.2 Parity calibration via binary calibration

Define the *parity outcomes* as

$$\text{for } t \geqslant 2,\ \widetilde{y}_t := \mathbb{1}\left\{y_t \leqslant y_{t-1}\right\}, \tag{7.5}$$

and observe that the parity calibration condition (Eq. (7.2)) is equivalently written as,

$$\frac{\sum_{t=2}^{T} \widetilde{y}_t \mathbb{1}\left\{\hat{p}_t = p\right\}}{\sum_{t=2}^{T} \mathbb{1}\left\{\hat{p}_t = p\right\}} \to p, \forall p \in [0,1]. \tag{7.6}$$

Thus parity calibration is in fact targeting the binary sequence $\widetilde{y}_t$, instead of y_t. In this section, we show how this connection allows us to leverage powerful techniques from the rich literature of binary calibration that goes back four decades (DeGroot and Fienberg, 1981; Dawid, 1982; Foster and Vohra, 1998). Of specific interest to us will be a class of methods that have been proposed for *post-hoc calibration* of machine learning (ML) classifiers, which we review next.

7.2.1 Post-hoc binary calibration

This subsection recalls basic post-hoc calibration ideas to enable independent reading. It can be skipped if the reader is familiar with Chapter 1.

Let $f : \mathcal{X} \to [0,1]$ be a binary classifier that takes as input a feature vector in feature space $\mathcal{X}$ and outputs a score in $[0,1]$. Suppose a feature-label pair (X, Y) is drawn from some distribution P over $\mathcal{X} \times \{0,1\}$. Then, f is said to be calibrated (in the binary sense) if

$$P(Y = 1 \mid f(X)) = f(X). \tag{7.7}$$

The terms on either side of the equal sign are random variables and the equality is understood almost-surely. The connection between (7.6) and (7.7) is evident: $\hat{p}_t$ is like $f(X)$, conditioning on the random variable $f(X)$ is akin to using indicators in the numerator/denominator, and $\widetilde{y}_t$ is like Y.

We do not expect ML models to be calibrated "out-of-the-box". So, if f is a logistic regression or neural network trained on some training data, it is unlikely to satisfy an approximate version of (7.7) on unseen data. Post-hoc calibration techniques transform f to a function that is better calibrated by using a so-called *calibration dataset* $\mathcal{D}_{\text{cal}} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_c, y_c)\}$. $\mathcal{D}_{\text{cal}}$ is a set of points on which f was not trained—in practice $\mathcal{D}_{\text{cal}}$ is often just the validation dataset. $\mathcal{D}_{\text{cal}}$ is used to a learn a mapping $m : [0, 1] \to [0, 1]$ so that $m \circ f$ is better calibrated than f. By way of an example, we now introduce the popular Platt scaling technique (Platt, 1999) that will be central to this work (henceforth, Platt scaling is referred to as PS). Given a pair of real numbers $(a, b) \in \mathbb{R}^2$, the PS mapping $m^{a,b} : [0, 1] \to [0, 1]$ is defined as,

$$m^{a,b}(z) = \text{sigmoid}(a \cdot \text{logit}(z) + b).$$

Here $\text{logit}(z) = \log(\frac{z}{1-z})$ and $\text{sigmoid}(z) = 1/(1 + e^{-z})$ are inverses of each other. Thus PS is a logistic model on top of the f-induced one-dimensional feature $\text{logit}(f(x)) \in [0, 1]$, instead of on the raw feature $x \in \mathcal{X}$. In the post-hoc setting, (a, b) are set to the values that minimize log-loss (equivalently cross entropy loss) on $\mathcal{D}_{\text{cal}}$:

$$(\widehat{a}, \widehat{b}) = \underset{(a,b)\in\mathbb{R}^2}{\arg\min} \sum_{(\mathbf{x}_s, y_s)\in\mathcal{D}_{\text{cal}}} l(m^{a,b}(f(\mathbf{x}_s)), y_s), \tag{7.8}$$

where $l(p, y) = -y \log p - (1 - y) \log(1 - p)$.

We briefly note some other popular post-hoc calibration methods. These broadly fall under two categories: parametric scaling methods such as beta scaling (Kull et al., 2017), temperature scaling (Guo et al., 2017), and PS (**platt1999probabilistic**); and nonparametric methods such as binning (Zadrozny and Elkan, 2001; Gupta et al., 2020; Gupta and Ramdas, 2021), isotonic regression (Zadrozny and Elkan, 2002), and Bayesian binning (Naeini et al., 2015).

7.2.2 Parity calibration using online versions of Platt Scaling (PS)

To achieve parity calibration using post-hoc techniques, we start with a base cdf predictor $G : \mathcal{X} \to \Delta(\mathcal{Y})$ derived from an expert—such as an epidemiologist, a weather forecaster, or a stock trader. Here, $\Delta(\mathcal{Y})$ refers to the space of distributions over $\mathcal{Y}$. If the expert is an ML engineer, such a G can be obtained using Gaussian processes (Rasmussen, 2004) or probabilistic neural networks (Nix and Weigend, 1994; Lakshminarayanan et al., 2017), among other methods. The test-stream occurs after G has been trained and fixed. This G gives us a $\hat{F}_t$ as described in the introduction: $\hat{F}_t = G(\mathbf{x}_t)$. Recall that the strategy Eq. (7.1) is to forecast $\hat{p}_t = \hat{F}_t(y_{t-1})$. If $\hat{F}_t$ were the true cdf of y_t given the past, the above $\hat{p}_t$ would be the true probability of $\widetilde{y}_t = 1$, and thus the most useful parity forecast possible.

However, in Section 7.1.1 we showed that we must modify $\hat{p}_t$ in order to achieve parity calibration. We propose using PS to perform this modification (any post-hoc calibration method can be used; we focus on PS in this work). A natural possibility would be to use an initial part of the

Algorithm 7.1 Platt scaling (PS) variants for parity calibration

```
Input: Any base forecaster $G : \mathcal{X} \to \Delta(\mathcal{Y})$, covariate-outcome pairs $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots \in \mathcal{X} \times \mathcal{Y}$, update-frequency $\texttt{uf}$, moving-window-size $\texttt{ws}$.
Output: PS forecasts $(\hat{p}_t^{\text{IW}}, \hat{p}_t^{\text{MW}}, \hat{p}_t^{\text{OPS}})_{t=2}^{\infty}$
Initialize IW, MW, OPS parameters:
  $(a^{\text{IW}}, b^{\text{IW}}) = (a^{\text{MW}}, b^{\text{MW}}) = (a^{\text{OPS}}, b^{\text{OPS}}) \leftarrow (1, 0)$
for $t = 2$ to $T$ do
    $\widetilde{y}_t = \mathbb{1}\{y_t \leqslant y_{t-1}\}$
    $\hat{p}_t = G(\mathbf{x}_t)[y_{t-1}]$
    $\hat{p}_t^{\text{IW}} \leftarrow \text{sigmoid}(a^{\text{IW}} \cdot \text{logit}(\hat{p}_t) + b^{\text{IW}})$
    $\hat{p}_t^{\text{MW}} \leftarrow \text{sigmoid}(a^{\text{MW}} \cdot \text{logit}(\hat{p}_t) + b^{\text{MW}})$
    $\hat{p}_t^{\text{OPS}} \leftarrow \text{sigmoid}(a^{\text{OPS}} \cdot \text{logit}(\hat{p}_t) + b^{\text{OPS}})$
    if $t$ is a multiple of $\texttt{uf}$ then
        $(a^{\text{IW}}, b^{\text{IW}}) \leftarrow$ optimal PS parameters
          based on (7.8) setting $\mathcal{D}_{\text{cal}} = (\mathbf{x}_s, \widetilde{y}_s)_{s=1}^{t}$
        $(a^{\text{MW}}, b^{\text{MW}}) \leftarrow$ optimal PS parameters
          based on (7.8) setting $\mathcal{D}_{\text{cal}} = (\mathbf{x}_s, \widetilde{y}_s)_{s=t-\texttt{ws}+1}^{t}$
    end if
    $(a^{\text{OPS}}, b^{\text{OPS}}) \leftarrow \text{OPS}((\mathbf{x}_1, \widetilde{y}_1), \ldots, (\mathbf{x}_t, \widetilde{y}_t))$
    (OPS is Algorithm 7.2 in Appendix 7.D)
end for
```

test-stream to learn fixed PS parameters once, as described in the previous subsection. However, real-world regression sequences (weather, stocks, etc) have non-stationary shifting behavior across time. Therefore, a fixed model is unlikely to remain calibrated over time.

In Algorithm 7.1 we outline three ways to mitigate this. Increasing Window (IW) updates the PS parameters using all datapoints until some recent time step, such as every 100 timesteps ($t = 100, 200$, etc). A related alternative, Moving Window (MW) is to use only the most recent datapoints when updating the PS parameters (instead of all the points). The third alternative is Online Platt Scaling (OPS) based on our own recent work (Gupta and Ramdas, 2023).

In the following section, we compare these online versions of Platt scaling on three real-world sequential prediction tasks. We find that OPS performs better than the base model, MW, and IW, across multiple settings. Further, while MW and IW involve re-fitting the PS parameters from scratch, OPS makes a constant time update at each step, hence the overall computational complexity of OPS is $O(T)$.

Brief note on theory and limitations of OPS. OPS satisfies a regret bound with respect to the Platt scaling class for log-loss (Gupta and Ramdas, 2023, Theorem 2.1). This means that the OPS forecasts do as well as forecasts of the single best Platt scaling model in hindsight. However, we note that OPS could fail if the best Platt scaling model is itself not good. This limitation can be overcome by combining OPS with a method called calibeating, as discussed in Gupta and Ramdas (2023). We do not pursue calibeating in this work since OPS already performs well on the data we considered.

7.3 Real-world case studies

We study parity calibration in three real-world scenarios: 1) forecasting COVID-19 cases in the United States, 2) forecasting weather, and 3) predicting plasma state evolution in nuclear fusion experiments. This diverse set of domains, datasets, and expert forecasters provides an attractive test-bed to demonstrate the parity calibration concept and the performance of the calibration methods from Section 7.2.2.

In each setting, the prediction target is real-valued, and we assume an expert forecaster provides regression forecasts $\hat{F}_t$ for the target. We also refer to $\hat{F}_t : \mathcal{Y} \to [0,1]$ as the *base regression model.* The expert forecaster implicitly provides parity probabilities $\hat{p}_t$ (following Eq. (7.1)). We refer to $\hat{p}_t$ as the *prehoc* probabilities, in contrast to the *post-hoc* probabilities that the calibration methods produce. We calibrate $\hat{p}_t$ with the calibration methods from Section 7.2.2 to produce the post-hoc probabilities $\hat{p}'_t$. Each calibration method requires a set of hyperparameters, which we tune with a validation set. Details regarding hyperparameter tuning are provided in Appendix 7.C.

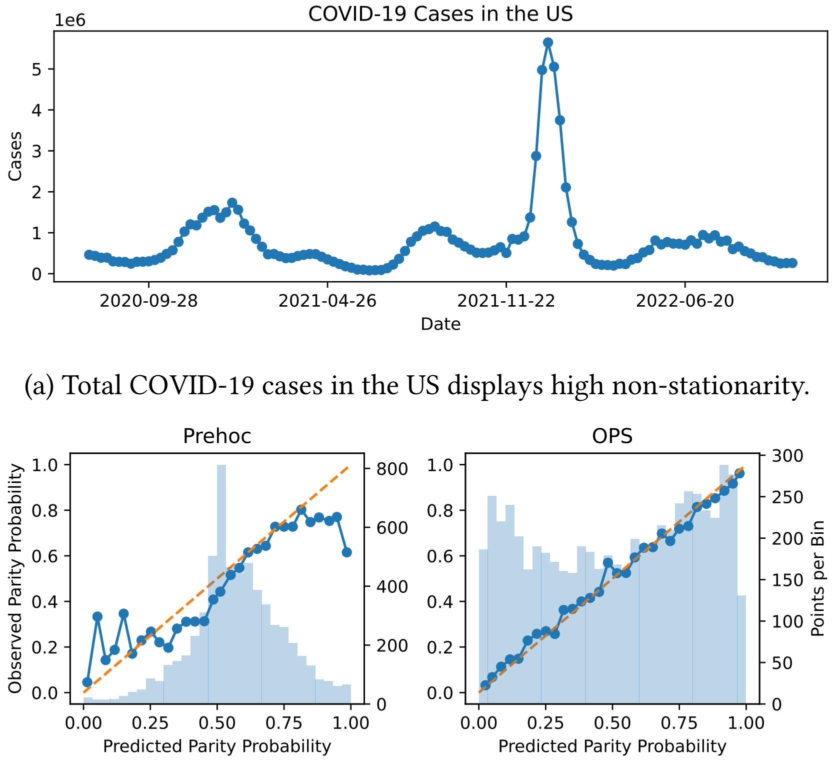
Metrics. Given a test dataset $\mathcal{D}_{\text{test}} = \{\mathbf{x}_t, y_t\}_{t=1}^T$, we initially assess the quantile calibration of $\hat{F}_t$ and the parity calibration of $\hat{p}_t$ and $\hat{p}'_t$ by visualizing the reliability diagrams and measuring calibration errors.

To assess quantile calibration of $\hat{F}_t$, we produce the reliability diagram using the Uncertainty Toolbox (Chung et al., 2021a), which takes a finite set of quantile levels $\mathcal{P} = \{p_i \in [0,1]\}$, computes the empirical coverage of the predictive quantile $\hat{F}_t^{-1}(p_i)$ as $p_{i,\text{obs}} = \frac{1}{T}\sum_{t=1}^T \mathbb{1}\left\{y_t \leqslant \hat{F}_t^{-1}(p_i)\right\}$, and plots each p_i against $p_{i,\text{obs}}$. Calibration error is then summarized into a single scalar with Quantile Calibration Error (QCE), which is computed as $\frac{1}{|\mathcal{P}|}\sum_i \mid p_{i,\text{obs}} - p_i \mid$. In our experiments, we set $\mathcal{P}$ to be 100 equi-spaced quantile levels in $[0,1]$.

To assess parity calibration of a parity probability $\hat{p}_t$, we follow the standard method of producing reliability diagrams in binary calibration (DeGroot and Fienberg, 1981; Niculescu-Mizil and Caruana, 2005). Noting that $\hat{p}_t$ is a predicted probability of the binary parity outcome $\widetilde{y}_t := \mathbb{1}\{y_t \leqslant y_{t-1}\}$, we first bin $\hat{p}_t$ into a finite set of fixed width bins $\mathcal{B} = \{B_m\}$, then for each bin B_m, we compute the average outcome as $\text{obs}(B_m) = \frac{1}{|B_m|}\sum_{t:\hat{p}_t \in B_m} \mathbb{1}\{\widetilde{y}_t = 1\}$ and the average prediction as $\text{pred}(B_m) = \frac{1}{|B_m|}\sum_{t:\hat{p}_t \in B_m} \hat{p}_t$, and finally, we plot $\text{pred}(B_m)$ against $\text{obs}(B_m)$ to produce the reliability diagram. Parity Calibration Error (PCE) summarizes the diagram following the standard definition of (ℓ_1-)expected calibration error (ECE): $\sum_m \frac{|B_m|}{T} \mid \text{obs}(B_m) - \text{pred}(B_m) \mid$. In our experiments, we set $\mathcal{B}$ to be 30 fixed-width bins: $[0, \frac{1}{30}), [\frac{1}{30}, \frac{2}{30}), \ldots [\frac{29}{30}, 1]$.

For the parity probabilities $\hat{p}_t$ and $\hat{p}'_t$, we additionally report sharpness and two metrics for accuracy: binary accuracy and area under the ROC curve. Sharpness (Sharp) is computed as $\sum_m \frac{|B_m|}{T} \cdot \text{obs}(B_m)^2$ and measures the degree to which the forecaster can discriminate events with different outcomes. Binary accuracy (Acc) and area under the ROC curve (AUROC) are computed following their standard definitions in binary classification. Appendix 7.A provides the full set of details on how each metric is computed. Lastly, in reporting the metrics in numeric tables, we denote each metric with their orientation, e.g. ↑ indicates that a higher value is more

Code is available at https://github.com/YoungseogChung/parity-calibration



(a) Total COVID-19 cases in the US displays high non-stationarity.

(b) Reliability diagrams for the prehoc parity probabilities from the expert forecasts (**left**) and OPS calibrated probabilities (**right**). **Blue bars** denote the frequency of predictions in each bin.

Figure 7.2: The prehoc parity probabilities for the COVID-19 single-timeseries setting are miscalibrated and un-sharp. Post-hoc calibration via OPS improves both aspects.

desirable and vice versa.

7.3.1 Case Study 1: COVID-19 cases in the US

In response to the COVID-19 pandemic, research groups across the world have created models to predict the short-term future of the pandemic. The COVID-19 Forecast Hub (Cramer et al., 2021) solicits and collects quantile forecasts of weekly incident COVID-19 cases in each US state (plus Washington D.C.), among other targets. Each week, the Hub generates an ensemble forecast from the dozens of submitted forecasts. This ensemble has proven to be more reliable and accurate than any constituent individual forecast in predicting other targets of interest (e.g. mortality (Cramer et al., 2022)). Thus, we take the ensemble forecast as the expert forecast and use its historical forecasts made between 2020-07-20 and 2022-10-24, which span a total of 119 weeks. Denoting the target y as the number of cases, there are effectively 51 timeseries, $\{y_{s,t}\}$: one for each US state $s \in \{$Alabama, Alaska, Arizona, ..., Wisconsin, Wyoming$\}$, and $t \in \{1, \ldots, 119\}$. For any given s, t, the expert forecast is provided by the Hub as seven forecasted

	Prehoc	$\text{OPS}_{\text{alpha-order}}$	$\text{OPS}_{\text{rand100}}$
PCE ↓	0.0599	0.0216	0.0246 ± 0.0002
Sharp ↑	0.2953	0.3087	0.3090 ± 0.00002
Acc ↑	0.6309	0.6727	0.6737 ± 0.0001
AUROC ↑	0.6922	0.7355	0.7357 ± 0.00002

Table 7.1: In the COVID-19 single-timeseries setting, OPS improves the prehoc parity probabilities w.r.t all metrics. ± indicates mean ± 1 standard error across 100 state orders.

	Prehoc	MW	IW	OPS
PCE ↓	0.0599	0.0748	0.0406	**0.0328**
Sharp ↑	0.2953	0.2882	0.2839	**0.2993**
Acc ↑	0.6309	0.6237	0.6055	**0.6522**
AUROC ↑	0.6922	0.6622	0.6403	**0.7035**

Table 7.2: In the COVID-19 sequential-batch setting, OPS outperforms prehoc and alternative PS methods. Best value for each metric is in bold.

quantiles for the distribution of $y_{s,t}$. Therefore, we must interpolate the quantiles to produce $\hat{F}_t$ (see Appendix 7.B.1 for details).

The observed targets $y_{s,t}$ are the incident number of cases actually reported from each state, for each week. Figure 7.2a visualizes a summary of the target timeseries: the total incident number of cases in the US $(= \sum_s y_{s,t})$. We can observe high non-stationarity, with periods of rapid increases and falls, and other periods of long monotonic trends.

Parity calibration of expert forecasts and OPS

Note that the underlying timeseries $\{y_{s,t}\}$ is indexed by both state and time. We transform this to a fully sequential timeseries by concatenating $\{y_{s,t}\}$ chronologically across t and in alphabetical order across s. In other words, within a given week, we observe the number of cases for the states in alphabetical order. We refer to this experiment setting as the *single-timeseries* setting.

The reliability diagram in Figure 7.2b (left) shows that the prehoc probabilities implied by the expert forecast ($\hat{p}_t$) are parity calibrated in the $[0.25, 0.75]$ region (i.e. higher predicted probabilities result in higher empirical frequencies), but are miscalibrated otherwise. The distribution of $\hat{p}_t$ displayed by the blue bars further indicate that $\hat{p}_t$ is centered around 0.5, an uninformative or less sharp prediction.

Figure 7.2b (right) displays the reliability diagram of $\hat{p}_t^{\text{OPS}}$. We observe significant improvements in both parity calibration and sharpness, i.e. $\hat{p}_t^{\text{OPS}}$ is much more dispersed compared to $\hat{p}_t$. The second column of Table 7.1 (labeled $\text{OPS}_{\text{alpha-order}}$) show these improvements via the PCE and Sharp metrics, and we can also observe improvement in accuracy.

One may question whether this improvement by OPS is specific to the alphabetical order of

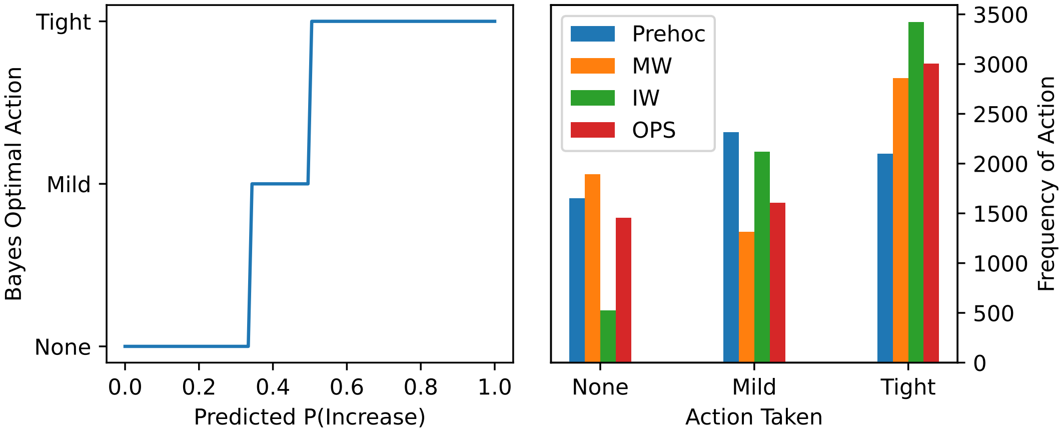


Figure 7.3: (Decision making on the COVID-19 dataset) (**left**) The Bayes optimal action for each predicted probability of increase in number of cases. (**right**) Frequency of each action taken by each method.

states. In the third column of Table 7.1 (labeled $\text{OPS}_{\text{rand100}}$), we show the mean and standard error of each of the metrics across 100 different random orders of the states, and observe that the improvements provided by OPS over prehoc are fairly robust.

Comparing calibration methods

We perform an additional experiment to compare the performance of MW, IW and OPS. In this experiment, we assume a more realistic test setting for the data-stream. At each timestep t, we assume we observe cases from all 51 states, $\{y_{s,t}\}_{s=1}^{51}$, and update the PS parameters with this batch of data. We then fix the PS parameters and calibrate the next full batch of predictions for timestep $t+1$. This settings assumes that PS parameters are updated once per week based on all the data observed during the week. We refer to this experiment setting as the *sequential-batch* setting.

The first 20 weeks of data (i.e. 20 weeks × 51 states = 1020 datapoints) were used to tune the hyperparameters of each method. The subsequent 99 weeks of data was used for testing. Table 7.2 displays the results of the sequential batch setting (note that the prehoc values are the same for this setting as in Table 7.1). OPS is the best performing method on all metrics when compared with MW, IW, and prehoc.

Decision-making with parity probabilities

In this section, we demonstrate the utility of OPS in a decision-making setting where parity outcomes (Eq. (7.5)) dictate the loss incurred. Using the same COVID-19 dataset, we assume a setting where a policymaker (i.e. the decision-maker) at each timestep must decide among

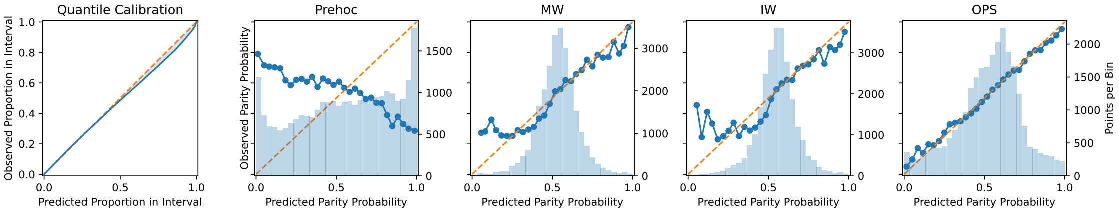


Figure 7.4: OPS significantly improves both parity calibration and sharpness of the base regression model predicting Pressure. The left two plots display the quantile calibration and parity calibration of the base model (Prehoc): it is nearly perfectly quantile calibrated, but terribly parity calibrated. **Blue bars** denote the frequency of predictions in each bin.

three levels of restrictions for disease spread prevention: Tight, Mild, or None. For any chosen level of restriction, the loss is dictated by the parity outcome in the number of cases, and the policymaker's goal is to minimize cumulative loss. A *Bayes optimal* policymaker will always choose an action which minimizes the expected loss, calculated with a predictive distribution over the loss (Lehmann and Casella, 2006). Hence the policymaker will assess the optimality of each action based on predicted parity probabilities.

We design an exemplar loss function $l_{\text{truth, decision}}$ as follows:

# Cases	Tight = 1	Mild = 2	None = 3
Increase = 1	$l_{1,1} = 0.3$	$l_{1,2} = 0.6$	$l_{1,3} = 1$ (max)
Decrease = 2	$l_{2,1} = 0.5$	$l_{2,2} = 0.2$	$l_{2,3} = 0$ (min)

Given this loss function, the Bayes optimal action is visualized in Figure 7.3 (left). On computing the the cumulative loss incurred with the predicted parity probabilities, we find that OPS incurs the lowest cumulative loss.

	Prehoc	MW	IW	OPS
Loss ↓	2119	2177	2196	**2050**

Figure 7.3 (right) shows the frequency of each action chosen by each method. We observe that OPS chooses Mild with relatively low frequency, which is a result of sharper and more accurate parity probabilities. We further note that IW results in a worse loss than prehoc despite being better parity calibrated (Table 7.2). To understand this, notice that IW is also less sharp and less accurate than Prehoc. Thus calibration, while a desirable quality, is not the only aspect to assess for good uncertainty quantification—sharpness and accuracy could also affect decision making.

7.3.2 Case Study 2: Weather forecasting

Our second case study examines weather forecasting using the benchmark Jena climate modeling dataset (2016), which records the weather conditions in Jena, Germany, with 14 different

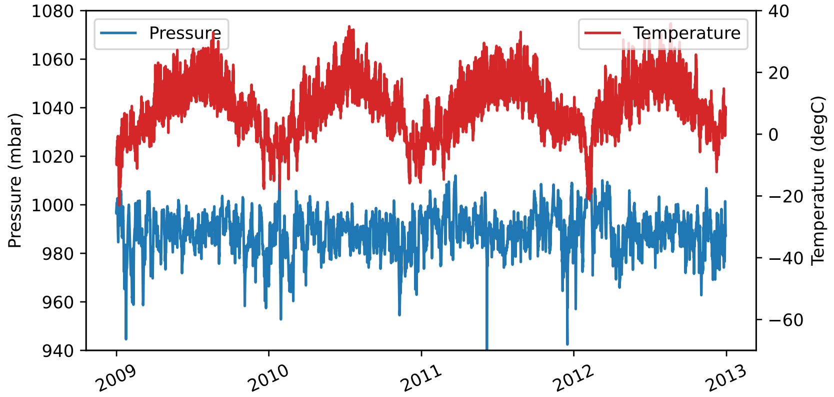


Figure 7.5: Snapshots of 4 years from the Temperature and Pressure timeseries display noise around a cyclical trend.

measurements, in 10 minute intervals, for the years 2009—2016. We did not have access to historical predictions from an expert weather forecaster, so instead we trained our own base regression model.

We follow the Keras tutorial on *Timeseries Forecasting for Weather Prediction*[1] to define our specific problem setup and train our base regression model. In summary, the regression model is implemented with an LSTM network (Hochreiter and Schmidhuber, 1997) which predicts the mean and variance of a Gaussian distribution. We trained 7 different models that each predict one of 7 weather features: Pressure, Temperature, Saturation vapor pressure, Vapor pressure deficit, Specific humidity, Airtight, and Wind speed. Appendix 7.B.2 provides more details on the problem setup.

Lastly, we note that unlike the COVID-19 data, the weather data (Figure 7.5) displays high levels of noise around a cyclical, repeating trend.

	QCE ↓	PCE ↓	Sharp ↑	Acc ↑	AUROC ↑
Prehoc	**0.0181±0.0026**	0.3493±0.0015	0.3019±0.0004	0.4044±0.0006	0.3525±0.0012
MW	N/A	0.0278±0.0005	0.3005±0.0004	0.6124±0.0008	0.6410±0.0012
IW	N/A	0.0322±0.0005	0.3013±0.0004	0.6147±0.0009	0.6450±0.0013
OPS	N/A	**0.0148±0.0002**	**0.3172±0.0004**	**0.6525±0.0007**	**0.7056±0.0010**

Table 7.3: OPS improves the overall quality of parity probabilities from the base regression model predicting Pressure. ± indicates mean ± 1 standard error, across 50 test trials. Best value for each metric is in bold.

Results on Pressure timeseries. We first examine results from one of the 7 models predicting Pressure. Figure 7.4 displays quantile calibration (i.e. probabilistic calibration) of the base

[1] https://keras.io/examples/timeseries/timeseries_weather_forecasting/

	PCE ↓	Sharp ↑	Acc ↑	AUROC ↑
Prehoc	0.0258±0.0005	0.3008±0.0007	0.6069±0.0011	0.6474±0.0016
MW	0.0201±0.0005	0.3002±0.0007	0.6050±0.0012	0.6439±0.0017
IW	0.0166±0.0003	0.3003±0.0008	0.6068±0.0010	0.6456±0.0016
OPS	**0.0150±0.0001**	**0.3232±0.0006**	**0.6665±0.0007**	**0.7275±0.0007**

Table 7.4: While MW, IW, OPS all improve parity calibration of the base classification model for Pressure (Prehoc), OPS is the only method that improves all metrics simultaneously. ± indicates mean ± 1 standard error, across 50 test trials. Best value for each metric is in bold.

model, and parity calibration before and after MW, IW and OPS are applied to the prehoc parity probabilities. We first note that the base model is almost perfectly quantile calibrated, but terribly parity calibrated, which corroborates our argument from Section 7.1.1, that calibration in regression does not imply parity calibration. In the same plot, we can see that MW, IW and OPS are all able to improve parity calibration, but the numerical results in Table 7.3 show that OPS produces superior parity probabilities w.r.t. all of the metrics considered.

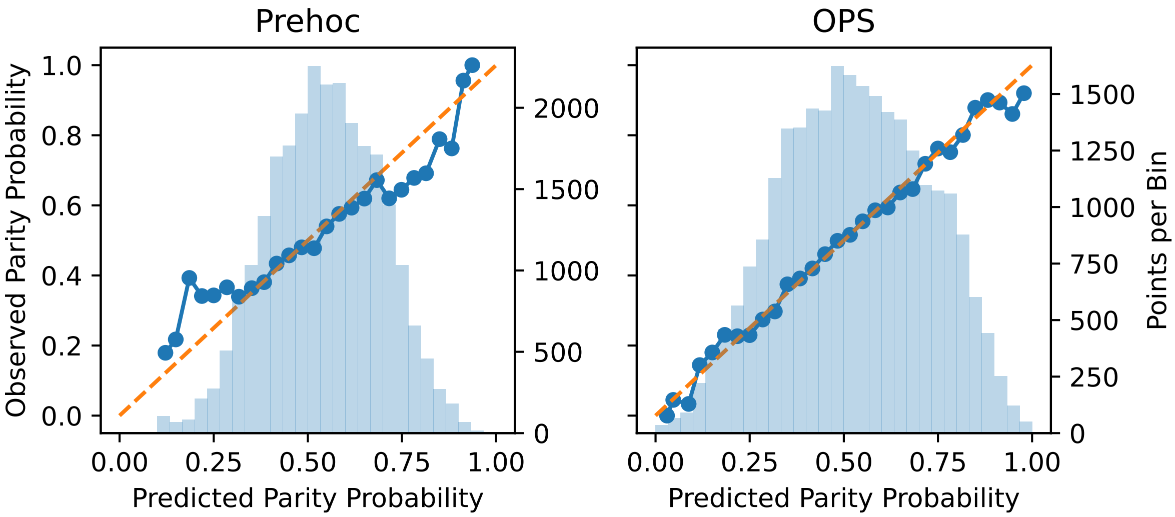


Figure 7.6: The base classification model for Pressure (Prehoc) is better parity calibrated than the base regression model (Figure 7.4 Prehoc), but OPS still improves its parity calibration and sharpness.

Binary classifiers as expert forecasters. While we have so far assumed that the expert forecaster provides regression models $\hat{F}_t$, one may argue that an expert forecaster may be well-aware that the downstream user is primarily concerned with parity probabilities. Accordingly, the expert may choose to directly model parity probabilities in the context of a binary classification problem.

In Figure 7.6 and Table 7.4, we show results from training a base binary classifier with parity

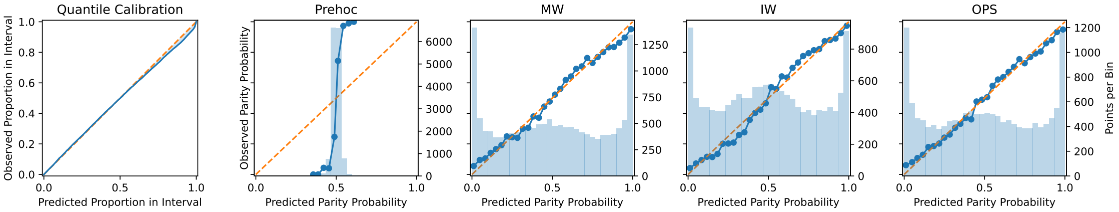


Figure 7.7: All methods (MW, IW, OPS) perform equally well in calibrating the Prehoc parity probabilities of the nuclear fusion dynamics model. The left two plots display the quantile calibration and parity calibration of the base dynamics model.

	QCE ↓	PCE ↓	Sharp ↑	Acc ↑	AUROC ↑
Prehoc	**0.0108±0.0003**	0.2571±0.0003	0.3243±0.0002	**0.7727±0.0003**	**0.8536±0.0002**
MW	N/A	0.0266±0.0002	0.3345±0.0002	0.7665±0.0003	0.8463±0.0002
IW	N/A	0.0291±0.0002	**0.3385±0.0002**	**0.7726±0.0003**	**0.8533±0.0002**
OPS	N/A	**0.0261±0.0002**	0.3334±0.0002	0.7629±0.0002	0.8440±0.0002

Table 7.5: MW, IW, and OPS all improve parity calibration and sharpness of the Prehoc fusion dynamics model predicting β_N, while maintaining roughly the same level of accuracy. ± indicates mean ± 1 standard error, across 50 test trials. Best value for each metric is in bold.

outcome labels and applying post-hoc calibration. As expected, the prehoc parity probabilities of the binary classification model is significantly better calibrated than the regression model. Post-hoc calibration still improves parity calibration further, especially in the case of OPS. In fact, OPS is the only method which improves all of the metrics simultaneously, while MW and IW notably worsen sharpness and AUROC. The full set of reliability diagrams is provided in Figure 7.10 in Appendix 7.B.2.

Results across all 7 timeseries. Table 7.6 in Appendix 7.B.2 shows each metric averaged across all 7 prediction targets: Table 7.6a displaying results with the base regression model, and 7.6b that of the base classification model. The pattern observed for the Pressure timeseries tend to hold on average across all 7 timeseries.

7.3.3 Case Study 3: Model-based Control for nuclear fusion

Nuclear fusion is the physical process during which atomic nuclei combine together to form heavier atomic nuclei, while releasing atomic particles and energy. Although fusion is possibly a safe, clean, and fuel-abundant technology for the future (Morse, 2018), there are various challenges to realizing fusion power, one of which is controlling nuclear fusion reactions (Humphreys et al., 2015).

Recently, model-based control methods, where a dynamics model of the system is learned and used to optimize control policies, has emerged as an effective control method for fusion devices (Abbate et al., 2023). To the experimenter utilizing the dynamics model, it is of significant

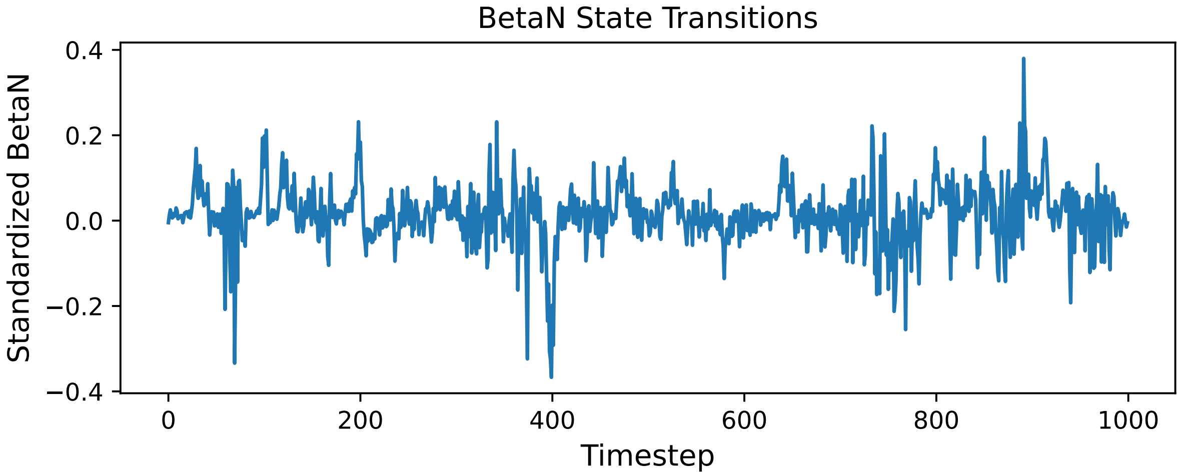


Figure 7.8: State transitions of the β_N signal during nuclear fusion experiments ("shots") concatenated across 50 training shots resemble trend-less noise.

interest to know when certain signals will increase, and whether the dynamics model assigns correct probabilities to the events (Char et al., 2021). In this section, we consider the problem of predicting the parity of β_N, which is a signal indicating reaction efficiency in a fusion device called a tokamak.

To this end, we design our empirical case study as follows. We take a pretrained dynamics model which was trained with a logged database of 10294 fusion experiments (referred to as "shots") conducted on the DIII-D tokamak (Luxon, 2002), a device in San Diego, CA, USA. This pretrained model has been used for model-based policy optimization for deployment in actual fusion experiments on this device (Char et al., 2021; Seo et al., 2021; Abbate et al., 2021). The model architecture is a recurrent probabilistic neural network (RPNN), which is a recurrent neural network with a Gaussian output head. We refer the reader to Appendix 7.B.3 for more details of the dynamics model and dataset. For testing, we allocate a set of 900 held-out test shots. On this test set, we produce the model's distributional predictions for β_N as the expert forecast. We concatenate the forecasts and the actual observed β_N values across the 900 test shots in chronological order into a single timeseries to assess parity calibration.

Figure 7.7 and Table 7.5 indicate that the expert forecast (Prehoc) is quantile calibrated but parity miscalibrated. The accuracy metrics in Table 7.5 indicate that despite prehoc's poor parity calibration, the model is still highly predictive, with an AUROC > 0.85. MW, IW and OPS significantly improve parity calibration and sharpness, while maintaining roughly the same level of accuracy.

We note that the β_N timeseries, as displayed in Figure 7.8, tends to fluctuate rapidly, between timesteps and between shots, almost resembling white noise. The pretrained model still manages to model the signal well, and assigns correct tendencies of increases/decreases in β_N: the relibility diagram of prehoc in Figure 7.7 shows that although the parity probabilities are not aligned with the empirical frequencies, they predict higher probabilities for actually higher frequency events. We believe this provides for a relatively easy post-hoc calibration problem, thus all methods (MW, IW, OPS) perform equally well. Hence, this case study highlights the significance of the base model's initial parity probabilities, especially in alleviating the difficulty of post-hoc calibration.

7.4 Conclusion

We considered the problem of forecasting whether a continuous-valued sequence is going to increase or decrease at the next time step. Such scenarios, where relative changes are more interpretable than actual values, are ubiquitous: COVID-19 cases per day, weather, or stock prices. In this context, we proposed the notion of parity calibration. To be parity calibrated, a forecaster must predict probabilities for the outcome increasing at the next time step, and these probabilities should be calibrated in the binary sense.

A decision-maker may attempt to achieve parity calibration by using regression forecasts produced by an expert forecaster. However, this is unlikely to give parity calibration. Instead, we proposed the usage of post-hoc binary calibration techniques to achieve parity calibration. Specifically, we advocated for a recently proposed online Platt scaling algorithm (OPS) in this setting. In three real-world empirical case studies, OPS consistently improves the overall quality of parity probabilities compared to the expert forecaster.

Appendices for Chapter 7

7.A Details on Evaluation: reliability diagrams and metrics

We provide details on how we assess a sequence of distributional forecasts $\{\hat{F}_t\}_{t=1}^T$ and parity probabilities $\{\hat{p}_t\}_{t=1}^T$, given a test dataset $\mathcal{D}_{\text{test}} = \{\mathbf{x}_t, y_t\}_{t=1}^T$. We assess distributional forecasts via Quantile Calibration, and the parity probabilities via Parity Calibration, Sharpness, and Accuracy metrics.

- **Quantile Calibration: reliability diagram and calibration error**

  To assess the quantile calibration of the distributional forecast $\hat{F}_t$, we produce the reliability diagram using the *Uncertainty Toolbox* (Chung et al., 2021a). This process works as follows. We take 100 equi-spaced quantile levels in $[0, 1]$: $p_i \in$ `np.linspace(0, 1, 100)`, and for each p_i, we compute the empirical coverage of the predictive quantile $\hat{F}_t^{-1}(p_i)$ with $\frac{1}{T}\sum_{t=1}^T \mathbb{1}\left\{y_t \leqslant \hat{F}_t^{-1}(p_i)\right\}$, and we denote this quantity as $p_{i,\text{obs}}$. Note that $p_{i,\text{obs}}$ is an empirical estimate of the term $\frac{1}{T}\sum_{t=1}^T F_t(\hat{F}_t^{-1}(p_i))$, from Eq. (7.3). The reliability diagram is produced by plotting $\{p_i\}$ on the x-axis against $\{p_{i,\text{obs}}\}$ on the y-axis. Quantile Calibration Error (QCE) is then computed as the average of the absolute difference between p_i and $p_{i,\text{obs}}$ over the 100 values of p_i: $\frac{1}{100}\sum_{i=1}^{100} \mid p_{i,\text{obs}} - p_i \mid$.

- **Parity Calibration: reliability diagram and calibration error**

  For parity calibration, we produce the reliability diagram following the standard method in binary classification (DeGroot and Fienberg, 1981; Niculescu-Mizil and Caruana, 2005). Note that the parity probability $\hat{p}_t$ is a prediction for the parity outcome $\widetilde{y}_t := \mathbb{1}\{y_t \leqslant y_{t-1}\}$ (Eq. (7.5)). Specifically, we first take 30 fixed-width bins of the predicted parity probabilities: $\{B_m\}_{m=1}^{30}$, where $B_m = [\frac{m-1}{30}, \frac{m}{30})$ for $m < 30$ and $B_{30} = [\frac{29}{30}, 1]$. The average outcome in bin B_m is computed as $\text{obs}(B_m) = \frac{1}{|B_m|}\sum_{t:\hat{p}_t \in B_m} \mathbb{1}\{\widetilde{y}_t = 1\}$, and the average prediction of bin B_m is computed as $\text{pred}(B_m) = \frac{1}{|B_m|}\sum_{t:\hat{p}_t \in B_m} \hat{p}_t$. The reliability diagram is then produced by plotting $\text{pred}(B_m)$ on the x-axis against $\text{obs}(B_m)$ on the y-axis. The blue bars in the background of each parity calibration reliability diagram represents the size of the bin: $|B_m|$. Parity Calibration Error (PCE) is then computed with this reliability diagram following the standard definition of (ℓ_1-)expected calibration error (ECE): $\sum_{m=1}^{30} \frac{|B_m|}{T} \mid \text{obs}(B_m) - \text{pred}(B_m) \mid$.

- **Sharpness**

  Assuming the same notation as above, sharpness is computed as: $\sum_{m=1}^{M} \frac{|B_m|}{T} \cdot \text{obs}(B_m)^2$, where M is the total number of bins. As indicated above, we use $M = 30$ in all of our experiments. We provide some additional intuition on this metric. A perfectly knowledgeable forecaster which outputs $\hat{p}_t = \widetilde{y}_t$ will place all predictions in either B_1 or B_M and achieve sharpness $= \frac{|B_1|}{T} \cdot \text{obs}(B_1)^2 + \frac{|B_M|}{T} \cdot \text{obs}(B_M)^2 = \frac{|B_1|}{T} \cdot 0^2 + \frac{|B_M|}{T} \cdot 1^2 = \frac{|B_M|}{T} = \frac{\sum_{t=1}^T \widetilde{y}_t}{T}$. On the other hand, if the forecaster places all predictions into a single bin B_k, then its sharpness will be $\text{obs}(B_k)^2 = \left(\frac{\sum_{t=1}^T \widetilde{y}_t}{T}\right)^2$. It can be shown that sharpness is always within the closed interval $\left[\left(\frac{\sum_{t=1}^T \widetilde{y}_t}{T}\right)^2, \frac{\sum_{t=1}^T \widetilde{y}_t}{T}\right]$ (Bröcker, 2009). Intuitively, sharpness measures the degree to which the forecaster attributes different valued predictions to events with different outcomes (i.e. labels). Hence, a sharper, or more precise, forecaster has more discriminative power, and this is reflected in a higher sharpness metric.

- **Accuracy metrics (Acc and AUROC)**

  Accuracy is measured in the binary classification sense, where the true labels are the observed parity outcomes: $\mathbb{1}\{y_t \leqslant y_{t-1}\}$ (Eq. (7.5)).

  - **Binary accuracy (Acc)** is computed by regarding $\hat{p}_t \geqslant 0.5$ as the positive class prediction, and the opposite case as the negative class prediction.
  - **Area under the ROC curve (AUROC)** is computed using the `scikit-learn` Python package, which implements the standard definition of the score. Specifically, we called the function `sklearn.metrics.roc_auc_score` with the predictions $\{\hat{p}_t\}$ and labels $\mathbb{1}\{y_t \leqslant y_{t-1}\}$.

7.B Additional Details on Case Studies

7.B.1 Additional Details on COVID-19 Case Study

Details on Interpolating Expert Forecasts for COVID-19 Case Study

The expert forecast provided by the COVID-19 Forecast Hub is represented as a set of quantiles. To derive the parity probabilities $\hat{p}_{s,t}$, we need to interpolate the expert forecast, as the forecast contains predicted quantiles at only 7 quantile levels : $\{0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975\}$. We interpolate under the assumption that the density between two adjacent quantiles τ_k and τ_{k+1} are defined by the normal distribution specified by those two quantiles. Specifically, for two quantiles τ_k and τ_{k+1} and forecast values $x_k^{(s,t)}$ and $x_{k+1}^{(s,t)}$, we compute

$$\sigma_k^{(s,t)} = \frac{x_{k+1}^{(s,t)} - x_k^{(s,t)}}{\Phi^{-1}(\tau_{k+1}) - \Phi^{-1}(\tau_k)},$$

$$\mu_k^{(s,t)} = x_k^{(s,t)} - \sigma_k^{(s,t)} \Phi^{-1}(\tau_k),$$

where Φ is the standard normal cdf. For each forecast, if $x_k^{(s,t)} \leqslant y_{s,t-1} < x_{k+1}^{(s,t)}$, then the parity probability

$$\hat{p}_{s,t} = \Phi\left(\frac{y_{s,t-1} - \mu_k^{(s,t)}}{\sigma_k^{(s,t)}}\right).$$

If $y_{s,t-1} < x_1^{(s,t)}$, we can extrapolate using $\mu_1^{(s,t)}$ and $\sigma_1^{(s,t)}$, and if $y_{s,t-1} >= x_7^{(s,t)}$, we can extrapolate using $\mu_6^{(s,t)}$ and $\sigma_6^{(s,t)}$. However, this never occurs with the forecasts and observations in this dataset. Figure 7.9 provides a visualization of this interpolation scheme.

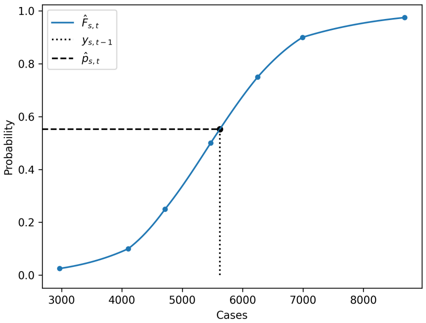


Figure 7.9: We use a piece-wise Gaussian interpolation of the expert forecast quantiles to estimate the predictive cdf, from which we then calculate the parity probabilities.

Details on Experiment Setup for COVID-19 Case Study

Section 7.3.1 compares the expert forecaster, its parity probabilities and post-hoc calibration by OPS. We did not tune OPS hyperparameters in this experiment, so the full 119 weeks' worth of data was used for testing and reporting the results.

For Section 7.3.1, the first 20 weeks' worth of data was used for tuning hyperparameters, and the reported results are based on the remaining 99 weeks' worth of data as the test set.

For the decision-making experiment in Section 7.3.1, we used the parity probabilities produced from Section 7.3.1.

Although the chosen loss function is just one example, we observe that similar results hold with any loss function that satisfies: $l_{2,3} \leqslant l_{2,2} \leqslant l_{1,1} \leqslant l_{2,1} \leqslant l_{1,2} \leqslant l_{1,3}$.

7.B.2 Additional Details on Weather Forecasting Case Study

Details on Experiment Setup for Weather Forecasting Case Study

We used the modeling and training infrastructure provided by the Keras tutorial on *Time-series Forecasting for Weather Prediction*[2] which models this same dataset with an LSTM network (Hochreiter and Schmidhuber, 1997). We made one change to the model provided by the

[2] https://keras.io/examples/timeseries/timeseries_weather_forecasting/

tutorial: since we are interested in probabilistic forecasts instead of point forecasts, we changed the head of the model and the loss function from a point output trained with mean squared error loss to a mean and variance output that parameterizes a Gaussian distribution and trained it with the Gaussian likelihood loss. Such a model is also referred to as a mean-variance network or a probabilistic neural network (Lakshminarayanan et al., 2017; Nix and Weigend, 1994), and it is one of the most popular methods currently used in probabilistic regression.

While the tutorial's setup takes as input the past 120 hours' window of 7 features to predict the value of one feature (Temperature) 12 hours into the future, we expand the setting to predict all 7 features: Pressure, Temperature, Saturation vapor pressure, Vapor pressure deficit, Specific humidity, Airtight, and Wind speed. We thus train 7 separate base regression models, one for each prediction target.

For the in-text experiment **Binary classifers as expert forecasts**, we trained binary classification base models with parity outcomes (Eq. (7.5)) as the labels and took this model as the expert forecaster. We adopted the same model architecture as the base regression model and changed the last layer to output a logit. We then trained the model with the cross entropy loss.

The full Jena dataset spans from the beginning of January 2009 to the end of December 2016, with $420,551$ datapoints in total. In chronological order, we set $272,638$ datapoints to train the base models (both the regression and classification model) and the subsequent $83,390$ datapoints for validation. Following the same model training procedure as the tutorial, training was stopped early if the validation loss did not increase for 20 training epochs.

Afterwards, in running the post-hoc calibration methods (MW, IW, and OPS), we used the last $8,640$ datapoints of the validation set to tune the hyperparameters of each calibration method, and used subsequent windows of $8,640 \times 3 = 25,920$ datapoints for testing.

We run 50 test trials with a moving test timeframe to produce the mean and standard errors reported in Tables 7.3 and 7.4. Denoting the first test window as $[t+1, t+H]$ (i.e. H is set to $25,920$), we move this frame by a multiple of a fixed offset c into the future, and repeat this 50 times, to create a new set of 50 test sets. The resulting new test timeframes are $[t+1+(ck), t+H+(ck)]$, where $k = 0, 1, 2, \ldots 49$, and c was set to 336.

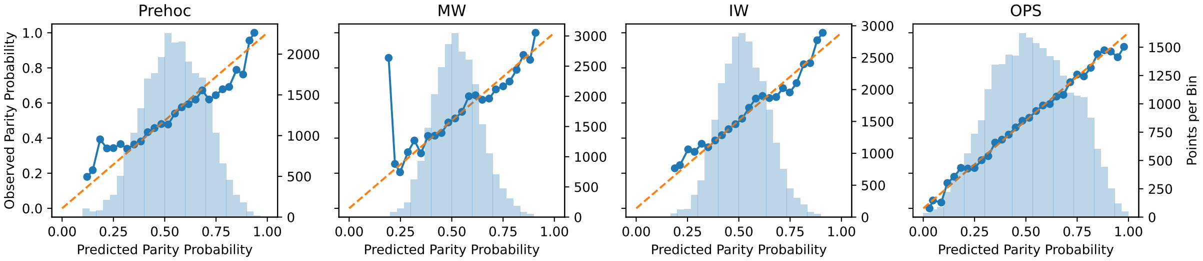


Figure 7.10: Reliability diagrams with a binary classification base model predicting Pressure. This is the full set of reliability diagrams for Figure 7.6 from Section 7.3.2. The left-most plot shows parity calibration of the base classification model (Prehoc), and the next three plots show the effects of MW, IW and OPS in calibrating the Prehoc parity probabilities. OPS produces the most calibrated and sharp parity probabilities.

	QCE ↓	PCE ↓	Sharp ↑	Acc ↑	AUROC ↑
Prehoc	**0.0266 ± 0.0052**	0.2794 ± 0.0161	0.2915 ± 0.0117	0.4902 ± 0.0159	0.4806 ± 0.0249
MW	N/A	0.0233 ± 0.0048	0.2913 ± 0.0117	0.5610 ± 0.0106	0.5419 ± 0.0195
IW	N/A	0.0188 ± 0.0047	0.2913 ± 0.0118	0.5630 ± 0.0099	0.5403 ± 0.0209
OPS	N/A	**0.0159 ± 0.0009**	**0.2961 ± 0.0122**	**0.5790 ± 0.0122**	**0.5830 ± 0.0217**

(a) Numerical results averaged across all 7 prediction settings where the base model is a Gaussian regression model. The base regression model (Prehoc) tends to be well quantile calibrated (QCE) but terribly parity calibrated (PCE). All methods (MW, IW, OPS) improve parity calibration, but OPS is the only method which improves all metrics simultaneously. Best value for each metric is in bold.

	PCE ↓	Sharp ↑	Acc ↑	AUROC ↑
Prehoc	0.0247 ± 0.0016	0.3049 ± 0.0074	0.6078 ± 0.0099	0.6348 ± 0.0136
MW	0.0170 ± 0.0018	0.3049 ± 0.0075	0.6061 ± 0.0102	0.6340 ± 0.0143
IW	0.0156 ± 0.0012	0.3047 ± 0.0074	0.6075 ± 0.0098	0.6340 ± 0.0136
OPS	**0.0135 ± 0.0013**	**0.3134 ± 0.0075**	**0.6278 ± 0.0121**	**0.6643 ± 0.0183**

(b) Numerical results averaged across all 7 prediction settings where the base model is a binary classification model trained with parity outcome labels. The base classification model (Prehoc) tends to be much better parity calibrated than when a regression base model is used (above Table 7.6a). All methods (MW, IW, OPS) improve parity calibration further, but OPS is the only method which improves all metrics simultaneously. Notably, MW and IW tends to decrease the accuracy of the parity probabilities. Best value for each metric is in bold.

Table 7.6: Numerical results from the weather forecasting case study (Section 7.3.2), averaged across all 7 forecasting targets. Table 7.6a displays results with the Gaussian regression base model, and Table 7.6b displays results with the binary classification base model. ± indicates mean ± 1 standard error, across the 7 prediction target settings.

Additional Results on Weather Forecasting Case Study

We shows additional plots and tables from the experimental results in Section 7.3.2 of the main chapter.

Figure 7.10 displays the full set of reliability diagrams for Figure 7.6, which corresponds to the in-text experiment **Binary classifiers as expert forecasts** in Section 7.3.2.

Table 7.6 displays the numerical results from the weather forecasting case study when averaged across all 7 prediction target settings. This corresponds to the in-text experiment **Results across all 7 timeseries** in Section 7.3.2. To produce these results, we fixed the test timeframe to be the first test timeframe $[t+1, t+H]$ for all prediction target settings, then computed the mean and standard errors across the 7 sets of metrics produced (one set for each prediction target).

7.B.3 Additional Details on Control in Nuclear Fusion Case Study

Details on Experiment Setup for Control in Nuclear Fusion Case Study

The expert forecaster for the nuclear fusion experiment in Section 7.3.3 is provided by a pretrained dynamics models that was used to optimize control policies for deployment on the DIII-D tokamak (Luxon, 2002), a nuclear fusion device in San Diego that is operated by General Atomics. The dynamics model was trained with logged data from past experiments (referred to as "shots") on this device. Each shot consists of a trajectory of (state, action, next state) transitions, and one trajectory consists of ~ 20 transitions (i.e. 20 timesteps).

As input, the model takes the current state of the plasma and the actuator settings (i.e. actions). The model outputs a multi-dimensional predictive distribution over the state variables in the next timestep. The state is represented by three signals: β_N (the ratio of plasma pressure over magnetic pressure), *density* (the line-averaged electron density), and *li* (internal inductance). For the actuators, the model takes in the amount of power and torque injected from the neutral beams, the current, the magnetic field, and four shape variables (*elongation*, a_{minor}, *triangularity-top*, and *triangularity-bottom*). This, along with the states, makes for an input dimension of 11 and output dimension of 3 for the states.

The model was implemented with a recurrent probabilistic neural network (RPNN), which features an encoding layer by an RNN with 64 hidden units followed by a fully connected layer with 256 units, and a decoding layer of fully connected layers with [128, 512, 128] units, which finally outputs a 3-dimensional isotropic Gaussian parameterized by the mean and a log-variance prediction.

The training dataset consisted of trajectories from 10294 shots, and the model was trained with the Gaussian likelihood loss, with a learning rate of 0.0003 and weight decay of 0.0001. In using dynamics models to sample trajectories and train policies, the key metric practitioners are concerned with is explained variance, hence explained variance on a held out validation set of 1000 shots was monitored during training. Training was stopped early if there was no improvement in explained variance over the validation set for more than 250 epochs. The test dataset consisted of another held-out set of 900 shots, with which we report all results presented in Section 7.3.3.

In all of our experiments, since β_N is the key signal of interest in our problem setting, we just examine the predictive distribution for β_N in the model outputs and ignore the other dimensions of the outputs.

In running the post-hoc calibration methods (MW, IW, and OPS), we used the same validation set to tune the hyperparameters of each calibration method, and used windows of $15,000$ datapoints from the concatenated test shot data for testing.

We run 50 test trials with a moving test timeframe to produce the mean and standard errors reported in Tables 7.5. Denoting the first test window as $[t+1, t+H]$ (i.e. H is set to $15,000$), we move this frame by a multiple of a fixed offset c into the future, and repeat this 50 times, to create a set of 50 test datasets. The resulting test timeframes are $[t+1+(ck), t+H+(ck)]$, where $k = 0, 1, 2, \ldots 49$, and c was set to 100.

7.C Details on Hyperparameters

Each of the three calibration methods we consider in Section 7.2.2, which we use in our experiments in Section 7.3, requires a set of hyperparameters.

- **MW** requires `uf` and `ws`.
  - `uf` determines how often the PS parameters $(a^{\mathrm{MW}}, b^{\mathrm{MW}})$ are updated.
  - `ws` determines the size of the calibration set that is used to update the PS parameters
- **IW** requires `uf`.
  - `uf` determines how often the PS parameters $(a^{\mathrm{IW}}, b^{\mathrm{IW}})$ are updated.
    Note that IW always uses all of the data seen so far to update the PS parameters.
- **OPS** requires γ and D.
  - γ can be understood as step size for the OPS updates.
  - D can be understood as regularization for the OPS updates.

We provide details on how these hyperparameters were tuned for each of the three case studies.

7.C.1 Hyperparameters for COVID-19 Case Study

We observed that OPS performed well with the default hyperparameters, so we did not tune hyperparameters for OPS for the COVID-19 case study. The default hyperparameter values used for OPS were $\gamma = 0.001$ and $\mathrm{D} = 10$.

For MW and IW, we tuned hyperparameters by optimizing parity calibration error (PCE, Section 7.3) on the first 20 weeks' worth of data as the validation set, over the following grids:

- `uf` $\in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$, separately for MW and IW
- `ws` $\in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$, for MW.

The COVID-19 dataset records data for each week, so the grid size of 1 represents 1 week.

The tuned hyperparameters we used for MW and IW are as follows:

- MW: $\texttt{uf} = 1, \texttt{ws} = 10$
- IW: $\texttt{uf} = 5$

7.C.2 Hyperparameters for Weather Forecasting Case Study

For each calibration method, the hyperparameters were tuned by optimizing parity calibration error (PCE, Section 7.3) on the validation dataset over the following grids:

- `uf` $\in [1, 24, 168, 336, 720, 2160]$, separately for MW and IW
- `ws` $\in [24, 168, 336, 720, 2160, 4320, 8640]$, for MW
- $\gamma \in$ [1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2], for OPS

- $\mathrm{D} \in [1, 10, 30, 50, 70, 100, 150, 200]$, for OPS.

The hyperparameters were tuned separately for each base model setting (regression and classification), for each method (MW, IW, and OPS), and for each base model predicting one of 7 targets (Pressure, Temperature, Saturation vapor pressure, Vapor pressure deficit, Specific humidity, Airtight, and Wind speed).

The tuned hyperparameters we used are as follows:

- **Base Regression Model**
    - Pressure Model
        - MW: $\texttt{uf} = 2160, \texttt{ws} = 8640$
        - IW: $\texttt{uf} = 2160$
        - OPS: $\gamma = 1\text{e-}5, \mathrm{D} = 50$
    - Temperature Model
        - MW: $\texttt{uf} = 336, \texttt{ws} = 8640$
        - IW: $\texttt{uf} = 168$
        - OPS: $\gamma = 1\text{e-}5, \mathrm{D} = 30$
    - Saturation Vapor Pressure Model
        - MW: $\texttt{uf} = 2160, \texttt{ws} = 2160$
        - IW: $\texttt{uf} = 336$
        - OPS: $\gamma = 1\text{e-}4, \mathrm{D} = 10$
    - Vapor Pressure Deficit Model
        - MW: $\texttt{uf} = 1, \texttt{ws} = 4320$
        - IW: $\texttt{uf} = 1$
        - OPS: $\gamma = 1\text{e-}3, \mathrm{D} = 1$
    - Specific Humidity Model
        - MW: $\texttt{uf} = 1, \texttt{ws} = 4320$
        - IW: $\texttt{uf} = 168$
        - OPS: $\gamma = 1\text{e-}5, \mathrm{D} = 30$
    - Airtight Model
        - MW: $\texttt{uf} = 2160, \texttt{ws} = 2160$
        - IW: $\texttt{uf} = 720$
        - OPS: $\gamma = 5\text{e-}5, \mathrm{D} = 10$
    - Wind Speed Model
        - MW: $\texttt{uf} = 1, \texttt{ws} = 168$
        - IW: $\texttt{uf} = 24$
        - OPS: $\gamma = 1\text{e-}4, \mathrm{D} = 10$

- **Base Classification Model**
  - Pressure Model
    - MW: $\texttt{uf} = 2160, \texttt{ws} = 8640$
    - IW: $\texttt{uf} = 720$
    - OPS: $\gamma = 5\text{e-}5, \text{D} = 30$
  - Temperature Model
    - MW: $\texttt{uf} = 1, \texttt{ws} = 4320$
    - IW: $\texttt{uf} = 168$
    - OPS: $\gamma = 1\text{e-}5, \text{D} = 150$
  - Saturation Vapor Pressure Model
    - MW: $\texttt{uf} = 336, \texttt{ws} = 4320$
    - IW: $\texttt{uf} = 720$
    - OPS: $\gamma = 1\text{e-}4, \text{D} = 30$
  - Vapor Pressure Deficit Model
    - MW: $\texttt{uf} = 1, \texttt{ws} = 168$
    - IW: $\texttt{uf} = 1$
    - OPS: $\gamma = 1\text{e-}5, \text{D} = 70$
  - Specific Humidity Model
    - MW: $\texttt{uf} = 1, \texttt{ws} = 2160$
    - IW: $\texttt{uf} = 2160$
    - OPS: $\gamma = 1\text{e-}5, \text{D} = 50$
  - Airtight Model
    - MW: $\texttt{uf} = 24, \texttt{ws} = 4320$
    - IW: $\texttt{uf} = 336$
    - OPS: $\gamma = 1\text{e-}3, \text{D} = 10$
  - Wind Speed Model
    - MW: $\texttt{uf} = 24, \texttt{ws} = 2160$
    - IW: $\texttt{uf} = 1$
    - OPS: $\gamma = 1\text{e-}5, \text{D} = 10$.

7.C.3 Hyperparameters for Control in Nuclear Fusion Case Study

The nuclear fusion dataset records measurements in 25 millisecond intervals. Therefore, in tuning hyperparameters, we design the search grid to represent lengths of time during which evolution of various plasma states are expected to be observable.

For each calibration method, the hyperparameters were tuned by optimizing parity calibration error (PCE, Section 7.3) on a validation dataset consisting of 1000 shot's worth of data, over the following grids:

- $\texttt{uf} \in [1, 2, 4, 8, 24]$, separately for MW and IW
- $\texttt{ws} \in [2, 8, 16, 24, 48, 60, 80, 100, 200]$, for MW
- $\gamma \in$ [1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2], for OPS
- $\mathrm{D} \in [1, 10, 30, 50, 70, 100, 150, 200]$, for OPS

The tuned hyperparameters we used are as follows:

- MW: $\texttt{uf} = 1, \texttt{ws} = 60$
- IW: $\texttt{uf} = 8$
- OPS: $\gamma = 5\text{e-}3, \mathrm{D} = 150$.

7.D Online Platt Scaling Algorithm

Algorithm 7.2 Online Platt Scaling (based on Gupta and Ramdas (2023), notation adapted)

```
**Input:** $\mathcal{K} = \{(x, y) : \|(x, y)\|_2 \leqslant 100\}$, time horizon $H$, and initialization parameter $(a_1^{\text{OPS}}, b_1^{\text{OPS}}) = (1, 0) =: \theta_1 \in \mathcal{K}$
**Hyperparameters and default values:** $\gamma = 0.1$, $D = 1$, $A_0 = (1/\gamma D)^2\, \mathbf{I}_2$
**for** $t = 1$ **to** $H$ **do**
    Play $\theta_t$, observe log-loss $l(m^{\theta_t}(f(\mathbf{x}_t)), y_t)$ and its gradient $\nabla_t := \nabla_{\theta_t} l(m^{\theta_t}(f(\mathbf{x}_t)), y_t)$
    $A_t = A_{t-1} + \nabla_t \nabla_t^\intercal$
    Newton step: $\widetilde{\theta}_{t+1} = \theta_t - \frac{1}{\gamma} A_t^{-1} \nabla_t$
    Projection: $(a_{t+1}^{\text{OPS}}, b_{t+1}^{\text{OPS}}) = \theta_{t+1} = \arg\min_{\theta \in \mathcal{K}} (\widetilde{\theta}_{t+1} - \theta)^\intercal A_t (\widetilde{\theta}_{t+1} - \theta)$
**end for**
```

Part III

Original contributions (related)

This part collates novel research contributions made by the author on the topic of uncertainty quantification, with broad connections to post-hoc calibration. Each chapter reproduces a separate published paper.

Chapter 8

Faster online calibration without randomization: interval forecasts and the power of two choices

This chapter is based on Gupta and Ramdas (2022a).

We study the problem of making calibrated probabilistic forecasts for a binary sequence generated by an adversarial nature. Following the seminal paper of Foster and Vohra (1998), nature is often modeled as an adaptive adversary who sees all activity of the forecaster except the randomization that the forecaster may deploy. A number of papers have proposed randomized forecasting strategies that achieve an ϵ-calibration error rate of $O(1/\sqrt{T})$, which we prove is tight in general. On the other hand, it is well known that it is not possible to be calibrated without randomization, or if nature also sees the forecaster's randomization; in both cases the calibration error could be $\Omega(1)$. Inspired by the equally seminal works on the power of two choices and imprecise probability theory, we study a small variant of the standard online calibration problem. The adversary gives the forecaster the option of making two nearby probabilistic forecasts, or equivalently an interval forecast of small width, and the endpoint closest to the revealed outcome is used to judge calibration. This power of two choices, or imprecise forecast, accords the forecaster with significant power—we show that a faster ϵ-calibration rate of $O(1/T)$ can be achieved even without deploying any randomization.

8.1 Introduction

A number of machine learning and statistics applications rely on probabilistic predictions. In economics, the influential discrete choice framework uses probabilistic modeling at its core (McFadden, 1974). Spiegelhalter (1986) argued that when predictive models are used in medicine for detecting disease, categorizing patient risk, and clinical trials, it is imperative that they provide accurate probabilities, in order to appropriately guide downstream decisions. Weather forecasters (and their audiences) would like to know the probability of precipitation on a given day (Brier, 1950).

We study the problem of producing probabilistic forecasts for binary events, that are calibrated without any assumptions on the data-generating process. Informally, a forecaster is calibrated if, on all the days that the forecaster produces a forecast p_t that is approximately equal to $p \in [0, 1]$, the empirical average of the observations $y_t \in \{0, 1\}$ is also approximately equal to p, and this is true for every $p \in [0, 1]$ that is frequently close to a forecast (Dawid, 1982). We formalize this next.

8.1.1 Calibration games and ϵ-calibration

Calibration-Game-I (classical)
(nature is an adaptive adversary)

At time $t = 1, 2, \ldots,$

- Forecaster plays $u_t \in \Delta([0, 1])$.
- Nature plays $v_t \in \Delta(\{0, 1\})$.
- Forecaster predicts $p_t \sim u_t$.
- Nature reveals $y_t \sim v_t$.

Calibration-Game-II (POTC)
(nature is an adaptive adversary,
forecaster has two nearby choices)

Fix $\epsilon > 0$. At time $t = 1, 2, \ldots,$

- Forecaster plays $p_{t0}, p_{t1} \in [0, 1]$, such that $p_{t0} \leqslant p_{t1}$ and $|p_{t1} - p_{t0}| \leqslant 2\epsilon$.
- Nature reveals $y_t \in \{0, 1\}$.
- If $y_t = 1$, set $p_t = p_{t1}$; else set $p_t = p_{t0}$.

Calibration-Game-I models the problem as a game between a forecaster and nature. The forecaster produces a *randomized* forecast $u_t \in \Delta([0, 1])$, which is a distribution over the space of forecasts $[0, 1]$. $\Delta(S)$ denotes the set of probability distributions over the set S (in every case, S is a standard set like $[0, 1]$ with a canonical σ-algebra). Nature observes u_t and responds with a Bernoulli distribution for the outcome $v_t \in \Delta(\{0, 1\}) = [0, 1]$. We abuse notation slightly and use v_t to denote both the Bernoulli distribution and its parameter in $[0, 1]$. Then forecaster and nature draw their actual actions, the forecast $p_t \sim u_t$ and the outcome $y_t \sim v_t$, simultaneously. At time $T > 1$, the prior activities $(u_t, v_t, p_t, y_t)_{t=1}^{T-1}$ are known to both players. The goal of the forecaster is to appear calibrated, defined shortly. Nature wishes to prevent the forecaster from appearing calibrated. Such a nature is typically referred to as an adaptive adversary.

Even before defining calibration formally, we can see that randomization is essential for the forecaster to demonstrate any semblance of being calibrated. If the forecaster is forced to put all his mass on a single p_t at each time (or equivalently if nature is an adaptive *offline* adversary), nature can play $v_t = y_t = \mathbb{1}\{p_t \leqslant 0.5\}$ to render the forecaster highly miscalibrated (Oakes, 1985; Dawid, 1985).

In anticipation of a forthcoming definition of ϵ-calibration error (equation (8.1)), we note that it will suffice for forecasters to only make discrete forecasts. Let $\epsilon > 0$ be a discretization or tolerance level, which is a small constant such as 0.1 or 0.01 depending on the application. For technical simplicity, we assume that $\epsilon = 1/2m$ for some integer $m \geqslant 2$. Consider the ϵ-cover of $[0, 1]$ given by the m intervals $I_1 = [0, 1/m), I_2 = [1/m, 2/m), \ldots, I_m = [1 - 1/m, 1]$. At time t, the forecaster makes a forecast on the '2ϵ-grid' of the mid-points of these intervals:

$$p_t \in \{M_1 := 1/2m = \epsilon, M_2 := 3/2m = 3\epsilon, \ldots, M_m := 1 - 1/2m = 1 - \epsilon\}.$$

Denote the total number of times the forecast is $p_t = M_i$ until time $T \geqslant 1$ as

$$N_i^T := \left|\{t \leqslant T : p_t = M_i\}\right|,$$

and the observed average of the y_t's when $p_t = M_i$ as

$$p_i^T := \begin{cases} \frac{1}{N_i^T}\sum_{t\leqslant T:p_t=M_i} y_t & \text{if } N_i^T > 0, \\ M_i & \text{otherwise.} \end{cases}$$

Following Foster (1999), the (ℓ_1-)calibration error at time T, CE_T, is defined as the weighted sum of the prediction errors for each possible forecast:

$$\mathrm{CE}_T := \sum_{i=1}^{m} \frac{N_i^T}{T} \cdot \left|M_i - p_i^T\right|, \text{ or equivalently } \sum_{i=1}^{m} \left| \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_i\right\}(M_i - y_t) \right|.$$

Finally, we define the ϵ-calibration error (ϵ-CE_T) as the calibration error with a slack of ϵ:

$$\epsilon\text{-}\mathrm{CE}_T := \max(\mathrm{CE}_T - \epsilon, 0). \tag{8.1}$$

In Calibration-Game-I, the forecaster and nature are allowed to randomize, thus ϵ-CE_T is a random quantity. A commonly studied object is its expected value. A forecaster is said to be ϵ-calibrated if, for *any* strategy of nature, the forecaster satisfies

$$\lim_{T\to\infty} \mathbb{E}\left[\epsilon\text{-}\mathrm{CE}_T\right] = 0, \text{ or equivalently } \mathbb{E}\left[\epsilon\text{-}\mathrm{CE}_T\right] = \underbrace{o_T(1)}_{f(T)}. \tag{8.2}$$

We are interested in the worst case value of $\mathbb{E}\left[\epsilon\text{-}\mathrm{CE}_T\right]$ against an adversarial nature, denoted as $f(T)$, henceforth called the ϵ-calibration rate or simply calibration rate. We show results about the asymptotic dependence of $f(T)$ as $T \to \infty$, holding ϵ as a fixed problem parameter on which f may depend arbitrarily.

8.1.2 Related work and our contributions

A number of papers have proposed ϵ-calibrated forecasting algorithms which guarantee $f(T) = O(1/\sqrt{T})$—the first was the seminal work of Foster and Vohra (1998), followed by a number of alternative proofs and generalizations of their result (Foster (1999), Fudenberg and Levine (1999), Vovk et al. (2005b), Mannor and Stoltz (2010, Section 4.1), Abernethy et al. (2011, Theorem 22), Perchet (2015, Section 4.2)).

In Theorem 8.3, we show that the $O(1/\sqrt{T})$ rate achieved by these algorithms is tight. There is a strategy for nature that ensures $f(T) = \Omega(1/\sqrt{T})$. Our proof uses a non-constructive lower bound for Blackwell approachability games (Mannor and Perchet, 2013).

Qiao and Valiant (2021) recently showed that the worst-case calibration error without the ϵ-slack, $\mathbb{E}\left[\mathrm{CE}_T\right]$, is $\Omega(T^{-0.472})$. In contrast, we treat ϵ as a small constant fixed ahead of time, and consider lower bounds on $\mathbb{E}\left[\epsilon\text{-CE}_T\right]$. Neither goal subsumes the other, so our lower bound complements theirs. In particular, observe that $\mathbb{E}\left[\text{CE}_T\right] = \Omega(T^{-1/2})$ can be forced by nature by playing a non-adaptive Bernoulli strategy, drawing independently each $y_t \sim \text{Bernoulli}(p)$ for some p. This strategy seems insufficient for deriving a useful lower bound on $\mathbb{E}\left[\epsilon\text{-CE}_T\right]$. These comparisons are further discussed in Section 8.4.6.

Foster (1999) showed that calibration is a Blackwell approachability instance (see Section 8.4.1), and while the rate $f(T) = \Omega(1/\sqrt{T})$ has not been formally established earlier (to the best of our knowledge), it is the rate one expects from a general Blackwell approachability instance (Cesa-Bianchi and Lugosi, 2006, Remark 7.7). Instead, the community has looked to establish positive results for alternative notions: calibration with more stringent tests than ϵ-calibration (Perchet, 2015; Rakhlin et al., 2011), calibration where the output space takes more than two values (Mannor and Stoltz, 2010), calibration with checking rules (Lehrer, 2001; Sandroni et al., 2003; Vovk et al., 2005b), weak calibration (Kakade and Foster, 2004), and smooth calibration (Foster and Hart, 2018). In particular, while no deterministic forecaster playing Calibration-Game-I can be ϵ-calibrated, there exist deterministic forecasters who are weakly/smoothly calibrated (Foster and Hart, 2018).

In our work, we take a slightly different approach from these papers. We retain the classical definition of ϵ-calibration but change the calibration game. In Calibration-Game-II, also called the Power-Of-Two-Choices (POTC) game, the forecaster reveals two forecasts $p_{t0}, p_{t1} \in [0, 1]$, such that $p_{t0} \leqslant p_{t1}$ and $|p_{t1} - p_{t0}| \leqslant 2\epsilon$. Since the earlier binning scheme used a 2ϵ-grid, that is $1/m = 2\epsilon$, this effectively allows the forecaster to choose a full bin as their forecast (rather than its midpoint), or equivalently to choose two consecutive bin midpoints. Thus there is no randomization, and nature knows the two forecasts. If nature chooses to play $y_t = 0$, $p_t = p_{t0}$ is used to judge the calibration of the forecaster, and if nature chooses to play $y_t = 1$, $p_t = p_{t1}$ is used. (One could say that the forecast closer to reality is used for measuring calibration, or that the forecaster decides which one of the two forecasts to use; these are all equivalent.) Obviously, without the restriction of p_{t0}, p_{t1} being 2ϵ-close, the problem is trivial: the forecaster would predict $p_{t0} = 0$ and $p_{t1} = 1$ in each round, and achieve zero error in every round. Requiring $|p_{t1} - p_{t0}| \leqslant 2\epsilon$ makes the problem interesting. The POTC setup may appear surprising to some and we devote Section 8.2 to motivating it.

The summary of our main result (Theorem 8.1) is as follows. In the POTC game, the forecaster can ensure—deterministically—that

$$\epsilon\text{-CE}_T = O(1/T). \tag{8.3}$$

Compared to (8.2), there is no expectation operator anymore since the forecaster is deterministic and nature being fully adaptively adversarial does not benefit from randomizing.

Our forecaster is a variant of Foster (1999). While Foster's forecaster randomizes over two nearby forecasts (making it *almost deterministic* in the sense of Foster and Hart (2021)), our forecaster predicts both these values and is judged with respect to the better one (and is actually, not almost, deterministic).

Remark 8.1 (Generalization from binary to bounded outputs). The POTC game can be modified for bounded, instead of binary, outputs. That is, nature can play $v_t \in [0, 1]$ and calibration would be judged with respect to the average of the v_t's on the instances when $p_t = M_i$. Note that this is not the same as nature playing $y_t \sim \text{Bernoulli}(v_t)$, since the calibration loss (left-hand-side of (8.3)) is not linear in y_t. With bounded outputs, the same $O(1/T)$ calibration rate can be achieved by a minor modification to our proposed forecasting strategy; see Appendix 8.C for more details. A similar remark holds for Calibration-Game-I and the corresponding lower bound of $\Omega(1/\sqrt{T})$. This latter fact is evident without further details since the lower bound can only increase if nature is given more flexibility.

Organization. Section 8.2 provides further context and motivation for the POTC game. Section 8.3 presents our algorithm for the POTC game and proves the fast calibration rate of $O(1/T)$ for it (Theorem 8.1). Section 8.4 reviews the well-known equivalence between calibration and Blackwell's approachability theorem (Blackwell, 1956), using which we prove the slow calibration rate of $\Omega(1/\sqrt{T})$ for Calibration-Game-I (Theorem 8.3). Most proofs are presented alongside the results. Section 8.5 concludes with a discussion.

8.2 Motivation for the POTC calibration game

Calibration-Game-II or the POTC game is motivated by two rich fields of literature: imprecise probability and the power of two choices.

8.2.1 A practical perspective via imprecise probability

The reader may wonder what the practical usefulness of the POTC game is. Why would we judge the forecaster in such a manner? The answer is that our earlier problem was phrased in a fashion that makes the connection to the power of two choices transparent. But one can also re-cast the problem in the language of *imprecise probability*. In this area, one is typically not restricted to work with single, unique probability measures, but instead the axioms of probability are relaxed, and added flexibility is provided in order to work with *upper* and *lower* probability measures (Walley and Fine, 1982).

In the context of our problem, instead of saying that the probability of rain is 0.3, a forecaster is allowed to say $0.3 \pm \epsilon$. One may just say that the forecaster is slightly uncertain and does not wish to commit to a point forecast, and indeed we may not force a forecaster to announce a point forecast against their will. From a Bayesian or game-theoretic perspective, we may say that the forecaster allows bets against their forecast, represented as a contract which pays off y_t, but the forecaster's prices for buying and selling such a contract are slightly different. From a practical perspective, this type of *interval* forecast arguably has almost the same utility and interpretability to a layman as the corresponding point forecast. The use of upper and lower forecasts (translated to prices or betting odds) is standard in game-theoretic probability (Shafer and Vovk, 2019). Separately, the recent work of Cooman and De Bock (2022) establishes that *randomness is inherently imprecise* in a formal sense, and provides a different justification for the use of interval forecasts for binary sequences.

Remarkably, this small and seemingly insignificant change in reporting leads to a huge change in our ability to achieve calibration. (This gain can be rather puzzling: we were binning/gridding anyway, so why not report a full bin rather than its midpoint? How could that possibly improve our calibration error?!) Of course, we must figure out how to judge the quality of such an interval forecast: we must swap out $\mathbb{1}\{p_t = M_i\}(M_i - y_t)$ in CE_T for a generalized notion of error that dictates how far y_t was from the forecasted interval $A_t := [p_{t0}, p_{t1}]$. To do this, we use the distance from a point to a convex set: we replace M_i with the projection of y_t onto A_t, denoted $\text{proj}(y_t, A_t)$. This is exactly what our POTC version does, just expressed differently.

When we generalize the definition of calibration, it is notationally simpler to restrict the forecasted interval endpoints to be the same m gridpoints, meaning that $A_t = [M_i, M_{i+1}]$ for some $i \leqslant m - 1$ (rather than $A_t = I_i$, the intervals whose midpoints are M_i). In this case, we call a method that produces interval forecasts $(I_t)_{t \geqslant 1}$ as being ϵ-calibrated if:

$$\max\left(\sum_{i=1}^{m}\left|\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\left\{\text{proj}(y_t, A_t) = M_i\right\}\text{dist}(y_t, A_t)\right| - \epsilon, 0\right) = o(1), \tag{8.4}$$

where $\text{dist}(y, A) := |y - \text{proj}(y, A)|$ is the distance of y to interval A. (When the interval is a single point, we recover the original definition of calibration, but in this case we know that randomization is necessary for ϵ-calibration.)

Imprecise probability has also made an intriguing appearance in the simpler setting commonly considered in machine learning, of achieving calibration in offline binary classification in the presence of i.i.d. data. This is a problem where theoretical progress has been made on designing *distribution-free* algorithms that have calibration guarantees by just assuming that the covariate-label pairs of data are i.i.d., while also performing well on real data (Gupta et al., 2020; Gupta and Ramdas, 2021; Gupta and Ramdas, 2022b). Venn predictors are a class of distribution-free algorithms that produce imprecise probability forecasts (Vovk et al., 2003; Vovk and Petej, 2014). On observing the covariates of a new point, Venn predictors output a particular interval of probabilities $[p_0, p_1]$ for the unknown binary label. A strong, but slightly odd, calibration property holds: the authors prove that p_Y (a random and unknown prediction, since Y is unknown and random) achieves *exact* calibration in finite samples. One can, in some sense, view our work as extending the use of such imprecise interval forecasts to the online calibration setting with adversarial data.

8.2.2 The varied applications of the power of two choices

The power of two choices (POTC) refers to a remarkable result by Azar et al. (1994) for the problem of load balancing. Suppose n balls are placed independently and uniformly at random into n bins. It can be shown that with high probability, the maximum number of balls in a bin (the maximum *load*) will be $\widetilde{\Theta}(\log n)$. Consider a different setup where the balls are placed sequentially, and for each ball, two bin indices are drawn uniformly at random and offered to a *load-balancer* who gets to decide which of the two bins to place the ball in. The load-balancer attempts to reduce the maximum load by following a natural strategy: at each step, place the ball in the bin with lesser load. It turns out that with this strategy the maximum load drops exponentially to $\widetilde{\Theta}(\log\log n)$.

The POTC result has led a number of applications. In a network where one of many servers can fulfil a request, it is exponentially better to choose two servers (instead of one) at random and allocate the server with fewer existing requests (Azar et al., 1994). Using two hashes instead of one significantly reduces the load of a single hash bucket (Broder and Mitzenmacher, 2001). In circuit routing, selecting one of two possible circuits provably leads to decongestion (Cole et al., 1998). When allocating a task to one of many resources where an intensive query needs to be made about the resource capacity, querying two resources is often better than querying all resources, or querying a single resource (Azar et al., 1994). Recently, Dwivedi et al. (2019) used the POTC to develop an online thinning algorithm that produces *low-discrepancy sequences* on hypercubes, with applications to quasi Monte Carlo integration. For further applications and a survey of mathematical techniques, we refer the reader to the thesis of Mitzenmacher (1996), or the survey by Mitzenmacher et al. (2001).

In this work, we find yet another intriguing phenomenon involving the POTC, this time in the context of calibration. We modify the classical setup of calibration (Calibration-Game-I) to the POTC setup (Calibration-Game-II), by offering the forecaster two nearby choices. We show that this change accords the forecaster with significant power, enabling faster calibration, even without randomization.

8.3 Main results: algorithm and analysis

Consider the POTC game (Calibration-Game-II). Recall that the forecaster's probabilities correspond to the mid-points of the intervals $I_1 = [0, 1/m), \ldots, I_m = [1 - 1/m, 1]$, given by $M_1 = 1/2m, \ldots, M_m = 1 - 1/2m$. The forecaster can play either $(p_{t0}, p_{t1}) = (M_i, M_i)$ or $(p_{t0}, p_{t1}) = (M_i, M_{i+1})$ for some i. We can also say that the forecaster predicts one of the two intervals $\{M_i\}$ or $[M_i, M_{i+1}]$ respectively.

We introduce some notation to describe the algorithm. For $i \in [m] := \{1, 2, \ldots, m\}$ and $t \geqslant 1$, define:

$$\begin{aligned}
&\text{(left endpoint of interval } i\text{)} \quad l_i = (i-1)/m, \\
&\text{(right endpoint of interval } i\text{)} \quad r_i = i/m, \\
&\text{(frequency of interval } i\text{)} \quad N_i^t = \left|\{\mathbb{1}\{p_s = M_i\} : s \leqslant t\}\right|, \\
&\text{(observed average when } M_i \text{ was forecasted)} \quad p_i^t = \begin{cases} \sum_{s=1}^t y_s \mathbb{1}\{p_s = M_i\} / N_i^t & \text{if } N_i^t > 0 \\ M_i & \text{if } N_i^t = 0, \end{cases} \\
&\text{(deficit)} \quad d_i^t = l_i - p_i^t, \\
&\text{(excess)} \quad e_i^t = p_i^t - r_i.
\end{aligned}$$

The terminology 'deficit' alludes to the fact that if p_i^t is smaller than desired (to the left of l_i), then $d_i^t > 0$ (p_i^t is 'in deficit'). 'Excess' has the opposite interpretation.

POTC-Cal: Algorithm for forecaster in Calibration-Game-II
(for notation, see Section 8.3 below)

- At time $t = 1$, play $(p_{10}, p_{11}) = (M_1, M_1)$. Thus $p_1 = M_1$.
- At time $t+1$ ($t \geqslant 1$), if

  condition A: there exists an $i \in [m]$ such that $d_i^t \leqslant 0$ and $e_i^t \leqslant 0$,

  is satisfied, play (M_i, M_i) for any i that verifies condition A (that is, $p_{t+1} = M_i$). Otherwise,

  condition B: there exists an $i \in [m-1]$ such that $e_i^t > 0$ and $d_{i+1}^t > 0$,

  must be satisfied (see Lemma 8.5). Play (M_i, M_{i+1}) for any index i that verifies condition B (that is, $p_{t+1} = M_i$ if $y_{t+1} = 0$ and $p_{t+1} = M_{i+1}$ if $y_{t+1} = 1$).

8.3.1 Forecasting algorithm

The algorithm, presented on top of this page, is a variant of the one proposed by Foster (1999). Foster's forecaster isolates two relevant M_i's and randomizes over them; we use the same M_i's to form the reported interval. At time $t+1$, if there is a forecast M_i that is already 'good' in the sense that $p_i^t \in [l_i, r_i]$, the forecaster predicts M_i. Otherwise, the forecaster finds two consecutive values (M_i, M_{i+1}) such that p_i^t is in excess and p_{i+1}^t is in deficit (such an i exists by Lemma 8.5, Appendix 8.A). The forecaster plays (M_i, M_{i+1}). If nature reveals $y_{t+1} = 0$, then $p_{t+1} = M_i$, and the excess of p_i^t decreases. If nature reveals $y_{t+1} = 1$, then $p_{t+1} = M_{i+1}$, and the deficit of p_{i+1}^t decreases.

8.3.2 Analysis of POTC-Cal

We now present our main result along with a short proof.

Theorem 8.1. *POTC-Cal satisfies, at any time $T \geqslant 1$, for any strategy of nature,*

$$\epsilon\text{-}CE_T \leqslant m/T. \tag{8.5}$$

Proof. Consider any $t \geqslant 1$. We write each of the m terms in the calibration error at time t, CE_t, as follows:

$$\left| \frac{1}{t} \sum_{s=1}^{t} \mathbb{1}\left\{p_s = M_i\right\}(M_i - y_s) \right| = \frac{N_i^t \left|M_i - p_i^t\right|}{t} = \frac{N_i^t(\epsilon + \max(d_i^t, e_i^t))}{t}.$$

Define $E_t^{(i)} := N_i^t \max(d_i^t, e_i^t)$, and observe that

$$\epsilon\text{-CE}_T = \max\left(\sum_{i=1}^{m} \frac{N_i^T \epsilon + E_T^{(i)}}{T} - \epsilon, 0\right) = \max\left(\sum_{i=1}^{m} \frac{E_T^{(i)}}{T}, 0\right).$$

We will show that for every $i \in [m]$, $E_T^{(i)} \leqslant 1$, proving the theorem.

Consider some specific $i \in [m]$. If action i is never played, then $N_i^T = 0$, and $E_T^{(i)} = 0$. Suppose an action i has $N_i^T > 0$. For each $1 \leqslant t < T$, if $a_{t+1} \neq i$, then $E_{t+1}^{(i)} = E_t^{(i)}$. If $a_{t+1} = i$, then by Lemmas 8.1 and 8.2 (stated and proved below), $E_{t+1}^{(i)} \leqslant \max(E_t^{(i)}, 1)$. In other words, at all t, the value of $E_{t+1}^{(i)}$ either stays bounded by 1, or decreases compared to the previous value $E_t^{(i)}$. A trivial inductive argument thus implies $E_T^{(i)} \leqslant 1$. For completeness, we verify the base case: since $p_1 = M_1$, $E_1^{(i \neq 1)} = 0$ and $E_1^{(1)} \leqslant 1$ (as $d_1^1 \leqslant 0$ and $e_1^1 \leqslant 1$). □

Lemma 8.1. *Suppose condition A was satisfied at time $t+1$ and the forecast was $p_{t+1} = M_i$. Then, $N_i^{t+1} \max(d_i^{t+1}, e_i^{t+1}) \leqslant 1$.*

Proof. Since $p_{t+1} = M_i$, $N_i^{t+1} = N_i^t + 1$, and $N_i^{t+1} p_i^{t+1} = N_i^t p_i^t + y_{t+1}$. Then,

$$\begin{aligned}\left|d_i^{t+1} - d_i^t\right| = \left|e_i^{t+1} - e_i^t\right| = \left|p_i^{t+1} - p_i^t\right| &= \left|\frac{N_i^t p_i^t + y_{t+1}}{N_i^t + 1} - \frac{N_i^t p_i^t + p_i^t}{N_i^t + 1}\right| \\ &= \left|\frac{y_{t+1} - p_i^t}{N_i^t + 1}\right| \leqslant \frac{1}{N_i^t + 1} = \frac{1}{N_i^{t+1}}.\end{aligned} \tag{8.6}$$

Since by condition A, $\max(d_i^t, e_i^t) \leqslant 0$, we obtain $\max(d_i^{t+1}, e_i^{t+1}) \leqslant 1/N_i^{t+1}$. □

Lemma 8.2. *Suppose condition A was not satisfied at time $t+1$ and the forecast was $p_{t+1} = M_i$, following condition B. Then $N_i^{t+1} \max(d_i^{t+1}, e_i^{t+1}) \leqslant \max(N_i^t \max(d_i^t, e_i^t), 1)$.*

Proof. Suppose $y_{t+1} = 0$. Since we are playing as per condition B, $e_i^t > 0$. Since $d_i^t + e_i^t = l_i - r_i = -1/m$, we have that $d_i^t < 0$. Thus,

$$y_{t+1} = 0 \implies e_i^t > 0 \text{ and } d_i^t < 0.$$

Similarly, it can be verified that $y_{t+1} = 1 \implies e_i^t < 0$ and $d_i^t > 0$. Below we assume without loss of generality that $y_{t+1} = 0$. (A similar argument goes through for the case $y_{t+1} = 1$.)

We derive how $N_i^t \max(d_i^t, e_i^t)$ changes when going from t to $t+1$. There are two cases: $e_i^{t+1} \geqslant d_i^{t+1}$ or $e_i^{t+1} < d_i^{t+1}$. If $e_i^{t+1} \geqslant d_i^{t+1}$, then

$$\begin{aligned}N_i^{t+1} \max(d_i^{t+1}, e_i^{t+1}) = N_i^{t+1} e_i^{t+1} &= N_i^{t+1} p_i^{t+1} - N_i^{t+1} r_i \\ &= N_i^t p_i^t - N_i^{t+1} r_i \quad (\text{since } y_{t+1} = 0) \\ &= N_i^t e_i^t - r_i \\ &= N_i^t \max(d_i^t, e_i^t) - r_i \leqslant N_i^t \max(d_i^t, e_i^t).\end{aligned}$$

On the other hand if $e_i^{t+1} < d_i^{t+1}$, then,

$$\begin{aligned}N_i^{t+1} \max(d_i^{t+1}, e_i^{t+1}) &= N_i^{t+1} d_i^{t+1} \\ &\leqslant N_i^{t+1}\left(d_i^t + \left|d_i^{t+1} - d_i^t\right|\right) \\ &\overset{(*)}{<} N_i^{t+1}(0 + 1/N_i^{t+1}) = 1.\end{aligned}$$

Inequality $(*)$ holds since $d_i^t < 0$ and $\left|d_i^{t+1} - d_i^t\right| \leqslant 1/N_i^{t+1}$ (see set of equations (8.6)).

□

8.4 $\Omega(1/\sqrt{T})$ lower bound for the classical calibration game

Calibration-Game-I can be viewed as a repeated game with vector-valued payoffs/rewards. Such games were studied by Blackwell (1956), and are now commonly referred to as Blackwell approachability games. We review the reduction from calibration to Blackwell approachability and use it to prove the lower bound. Throughout this section, we denote the action space of the forecaster as $\mathcal{X} = \{M_1, M_2, \ldots, M_m\}$ and that of nature as $\mathcal{Y} = \{0, 1\}$. The random plays of the forecaster lie in $\Delta(\mathcal{X})$ which is a probability simplex in m dimensions. We embed $\Delta(\mathcal{X})$ in $\mathbb{R}^m$ to simplify discussion.

8.4.1 Calibration as an instance of Blackwell approachability

The fact that calibration can be modelled as a Blackwell approachability instance is well-known (since Foster (1999) and Hart and Mas-Colell (2000)). Suppose the actions of the forecaster and nature give a reward $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^m$ defined as follows: the i-th component of the reward vector $a = r(p \in \mathcal{X}, y \in \mathcal{Y}) \in \mathbb{R}^m$ is given by

$$a_i = \mathbb{1}\{p = M_i\} \cdot (M_i - y). \tag{8.7}$$

Let $\bar{a}^T := \sum_{i=1}^T r(p_t, y_t)/T$ be the average reward vector given component-wise by $\bar{a}_i^T = \sum_{t=1}^T \mathbb{1}\{p_t = M_i\}(M_i - y_t)/T$. Let B_ϵ be the ℓ_1-ball with radius ϵ, and dist the ℓ_1-distance function. Note that

$$\text{dist}(\bar{a}^T, B_\epsilon) = \max\left(\sum_{i=1}^m \left|\frac{1}{T}\sum_{t=1}^T \mathbb{1}\{p_t = M_i\}(M_i - y_t)\right| - \epsilon, 0\right) = \epsilon\text{-CE}_T.$$

Thus, the ϵ-calibration condition (8.2) is equivalent to $\lim_{T\to\infty} \mathbb{E}\left[\text{dist}(\bar{a}^T, B_\epsilon)\right] = 0$. If this condition is satisfied, we say that $\bar{a}^T$ *approaches* B_ϵ in the limit. Blackwell (1956) established necessary and sufficient conditions for approachability.

Theorem 8.2 (Corollary to Theorem 3 by Blackwell (1956))**.** *Assume the same setup as Calibration-Game-I, but the players receive a vector-valued reward* r*, as defined component-wise in* (8.7)*. The forecaster can ensure that* $\lim_{T\to\infty} \mathbb{E}\left[dist(\bar{a}^T, B_\epsilon)\right] = 0$ *if and only if for every* $v \in \Delta(\mathcal{Y}) = [0, 1]$*, there exists* $u \in \Delta(\mathcal{X})$ *such that the expected reward belongs to* B_ϵ*:*

$$\forall v \in \Delta(\mathcal{Y}),\ \exists u \in \Delta(\mathcal{X}) : \mathbb{E}_{p\sim u, y\sim v}\left[r(p, y)\right] = \sum_{i=1}^m u_i((1-v)\cdot r(M_i, 0) + v\cdot r(M_i, 1)) \in B_\epsilon. \tag{8.8}$$

The hypothetical situation considered in the theorem is akin to a one-shot game but with the order of the players reversed: nature plays v first and the forecaster responds with u. If the forecaster can *respond* to every play by nature and ensure that the expected reward lies in B_ϵ, then the forecaster can ensure that $\bar{a}^T$ approaches B_ϵ in the sequential game (where nature goes second each time). Abernethy et al. (2011) call this *response-satisfiability*; in their words, Theorem 8.2 is interepreted as response-satisfiability $\iff$ approachability.

Proposition 8.1. *The forecaster can exhibit response satisfiability* (8.8). *Thus the forecaster playing Calibration-Game-I can ensure* $\lim_{T\to\infty} \mathbb{E}\left[\mathit{dist}(\bar{a}^T, B_\epsilon)\right] = 0$ *and be* ϵ*-calibrated* (8.2).

The proof of this well-known result is in Appendix 8.A.3. A second question is of the rate at which the expected reward vector approaches the desired set. As reviewed in the introduction, a number of papers have shown that the rate of approachability for the ϵ-calibration game is $O(1/\sqrt{T})$. We show that this rate cannot be improved.

8.4.2 $\Omega(1/\sqrt{T})$ lower bound for the ϵ-calibration error rate

Theorem 8.3. *A forecaster playing Calibration-Game-I against an adversarial nature cannot achieve* ϵ*-calibration at a rate faster than* $O(1/\sqrt{T})$*. That is, for every strategy of the forecaster, there is a strategy of nature that ensures*

$$\mathbb{E}\left[\epsilon\text{-}CE_T\right] = \Omega(1/\sqrt{T}). \tag{8.9}$$

Mannor and Perchet (2013) analyzed the convergence rate in approachability games and characterized conditions, which if satisfied by a target set $\mathcal{C}$, entail that nature can ensure $\mathbb{E}\left[\text{dist}(\bar{a}^T, \mathcal{C})\right] = \Omega(1/\sqrt{T})$. In particular, if these conditions are satisfied B_ϵ, Theorem 8.3 follows immediately since $\text{dist}(\bar{a}^T, \mathcal{C}) = \epsilon\text{-CE}_T$.

Theorem 8.4 (Theorem 6.ii by Mannor and Perchet (2013)). *Let* $\mathcal{C}$ *be a closed convex set that is (i) minimal approachable, and (ii) mixed approachable. Then* $\mathcal{C}$ *cannot be approached at a rate faster than* $O(1/\sqrt{T})$*, or in other words,* $\mathbb{E}\left[\mathit{dist}(\bar{a}^T, \mathcal{C})\right] = \Omega(1/\sqrt{T})$*, where* $\bar{a}^T$ *is the average reward vector.*

In what follows, we introduce the conditions (i) minimal approachability and (ii) mixed approachability in the context of our calibration game, and show that they are satisfied by B_ϵ (Lemma 8.3 and Lemma 8.4 respectively).

8.4.3 Minimal approachability

For a point $u \in \mathbb{R}^m$ and a convex set $K \subseteq \mathbb{R}^m$, define the distance of u from K as $d_K(u) = \inf_{u' \in K} \|u - u'\|_2$. For any $\lambda > 0$, a convex set $K' \subseteq K$ is said to be a λ-shrinkage of K if $\{u : d_{K'}(u) \leqslant \lambda\} \subseteq K$. In the following definition of minimal approachability, we implicitly assume that the set of action sets of the players and the corresponding rewards have been fixed, and the goal is to characterize which convex sets are approachable and which are not.

Definition 8.1. A set K is *minimal* approachable if K is approachable, but no λ-shrinkage of K is approachable.

We now show the first condition required by Theorem 8.4.

Lemma 8.3. *The set* B_ϵ *is minimal approachable.*

Proof. Proposition 8.1 shows that B_ϵ is approachable so it remains to show that the minimality condition holds. Let K be a λ-shrinkage of B_ϵ for some $\lambda > 0$. We first argue that $K \subseteq B_{\epsilon-\lambda}$. Suppose this were not the case, that is, there exists $u \in K$ such that $u \notin B_{\epsilon-\lambda}$. By definition of the ℓ_1-ball $B_{\epsilon-\lambda}$, this means that $\|u\|_1 > \epsilon - \lambda$. We will show that such a u cannot belong to any λ-shrinkage of B_ϵ, in particular it cannot belong to K, leading to a contradiction.

Consider the point $u' = u + (\lambda/\|u\|_2)u$. Note that $d_K(u') \leqslant \|u - u'\|_2 = \lambda$. Since K is a λ-shrinkage of B_ϵ, this implies that $u' \in B_\epsilon$, or $\|u'\|_1 \leqslant \epsilon$. On the other hand, we have,

$$\begin{aligned}(\epsilon \geqslant)\ \|u'\|_1 &= \|u\|_1(1 + \lambda/\|u\|_2) \\ &\geqslant \|u\|_1(1 + \lambda/\|u\|_1) && \text{(for any vector } v \in \mathbb{R}^m,\ \|v\|_1 \geqslant \|v\|_2) \\ &= \|u\|_1 + \lambda > (\epsilon - \lambda) + \lambda = \epsilon,\end{aligned}$$

which is a contradiction. Thus $K \subseteq B_{\epsilon-\lambda}$, as claimed.

It follows that K is approachable only if $B_{\epsilon-\lambda}$ is approachable. We now show that for every $\lambda > 0$, $B_{\epsilon-\lambda}$ is not approachable. As in the proof of Proposition 8.1, the i-th component of the reward vector is given by $u_i(M_i - v)$. Suppose $v = 1/m$. Then for every M_i, $|M_i - v| \geqslant 1/2m = \epsilon$, by definition of m. Thus $|u_i(M_i - v)| \geqslant u_i\epsilon$, and $\sum_{i=1}^m |u_i(M_i - v)| \geqslant \sum u_i\epsilon = \epsilon$. Equivalently, $\mathbb{E}\left[r(p, y)\right] \notin B_{\epsilon-\lambda}$. By Theorem 8.2, $B_{\epsilon-\lambda}$ is not approachable. □

In order to describe the second condition required by Theorem 8.4 and show that it holds for B_ϵ, we need additional technical setup. The following subsection serves this purpose.

8.4.4 Reducing approachability to scalar-valued games

The vector-valued approachability game induces a number of scalar-valued min-max games, one for each direction in $\mathbb{R}^m$. The value of these scalar games is closely connected to the question of approachability.

Consider the approachability of B_ϵ with respect to individual directions, represented by arbitrary vectors $q \in \mathbb{R}^m$ (for intuition, one may equivalently think of q being direction vectors, those with ℓ_2-norm equal to one, but this restriction is technically unnecessary; we stick to $q \in \mathbb{R}^m$). Let $c \in B_\epsilon$ be such that q belongs to the *normal cone* of B_ϵ at c, that is, $\langle c, q\rangle = \sup_{c' \in B_\epsilon} \langle c', q\rangle = \epsilon \|q\|_\infty$. We call such a pair (c, q) as *admissible*. Consider the following one-shot min-max game defined for every admissible (c, q):

$$\begin{aligned}\mathrm{Val}(c, q) &= \min_{u \in \Delta(\mathcal{X})} \max_{v \in [0,1]} \langle \mathbb{E}_{p\sim u, y\sim v}\left[r(p, y)\right] - c, q\rangle \\ &= \min_{u \in \Delta(\mathcal{X})} \max_{v \in [0,1]} \left(\sum_{i=1}^m u_i q_i((1 - v)M_i + v(M_i - 1)) - \langle c, q\rangle\right) \\ &= \min_{u \in \Delta(\mathcal{X})} \max_{v \in [0,1]} \left(\sum_{i=1}^m u_i q_i(M_i - v) - \epsilon \|q\|_\infty\right).\end{aligned}$$

To appreciate the relationship between the Val(c, q) games and the B_ϵ-approachability game, consider the following. Suppose the forecaster can guarantee *one-shot approachability*, that is, there exists a fixed $u^\star \in \Delta(\mathcal{X})$ such that for every $v \in [0,1]$, $\mathbb{E}_{p\sim u^\star, y\sim v}[r(p,y)] \in B_\epsilon$. By definition of the normal cone, for every admissible (c,q), and any $c' \in B_\epsilon$, it holds that $\langle c'-c, q\rangle \leqslant 0$. In particular, $\mathbb{E}_{p\sim u^\star, y\sim v}[r(p,y)] \in B_\epsilon$ is such a c'. It follows that for every admissible (c,q), Val(c, q) $\leqslant 0$.

This observation does not hold in the reverse direction: even if $\mathrm{Val}(c,q) \leqslant 0$ for every admissible (c,q), one-shot approachability need not hold. The intuition is that the optimal u for different (c,q) can be different, and it is unclear how to merge them to achieve one-shot forecasting for the approachability game. In a remarkable result, Blackwell (1956) showed that the result does hold in the reverse direction for the *repeated* approachability game (as opposed to the one-shot approachability game).

Theorem 8.5 (Theorem 1 by Blackwell (1956))**.** *A convex set K is approachable if and only if for every admissible (c,q), Val$(c,q) \leqslant 0$.*

This condition has also been termed as halfspace-satisfiability by Abernethy et al. (2011). The result was stated in the language of convex cones by Mannor and Perchet (2013). Notice that for our problem, the min-max game does not depend on c, once we replace $\langle c,q\rangle$ with $\epsilon\left\|q\right\|_\infty$. We thus simplify notation and index our games only by $q \in \mathbb{R}^m$:

$$\mathrm{Val}(q) := \min_{u\in\Delta(\mathcal{X})}\max_{v\in[0,1]}\left(\sum_{i=1}^{m} u_i q_i (M_i - v) - \epsilon\left\|q\right\|_\infty\right). \tag{8.10}$$

We know that B_ϵ is approachable (Proposition 8.1) and hence by Blackwell's result, halfspace-satisfiability must hold. That is, for every $q \in \mathbb{R}^m$, $\mathrm{Val}(q) \leqslant 0$. For completeness, we verify this in Proposition 8.2 (Appendix 8.A).

Having defined the $\mathrm{Val}(q)$ games, we are now ready to define mixed approachability.

8.4.5 Pure* game and mixed approachability

In order to achieve a small value in the $\mathrm{Val}(q)$ game, the forecaster may play a randomized strategy, that is, $u^\star \neq e_i$, where e_i is one of the canonical basis vectors of $\mathbb{R}^m$. On the other hand, since nature goes second, she has no incentive to randomize: there will exist an optimal strategy $v^\star \in \{0,1\}$. In the following 'pure' game, the forecaster is also not allowed to randomize over his actions.

$$\mathrm{Val}^{\mathrm{p}}(q) := \min_{p\in\mathcal{X}}\max_{y\in\{0,1\}} \langle r(p,y), q\rangle - \epsilon\left\|q\right\|_\infty. \tag{8.11}$$

The superscript 'p' in $\mathrm{Val}^{\mathrm{p}}(\cdot)$ refers to the min-max game being over pure actions $p \in \mathcal{X}$ and $y \in \{0,1\}$. Let us refer to this as the pure game, and the game (8.10) as the mixed game.

Suppose the approaching player (forecaster in our case) can ensure halfspace-satisfiability using only pure actions: $\forall q,\ \mathrm{Val}^{\mathrm{p}}(q) \leqslant 0$. Mannor and Perchet (2013) showed that if this is true then then the approaching player can ensure approachability at a fast rate of $O(1/n)$. However, it is possible to achieve the fast rate even if the above condition is not true. Characterizing a situation where the fast rate is unachievable requires considering another game, whose value lies between the pure and mixed games. Define

$$\mathcal{X}^\star = \{p \in \mathcal{X} : p \in \text{support}(u^\star), \text{ where } u^\star \text{ is some optimal mixed strategy for the forecaster}\};$$
$$\mathcal{Y}^\star = \{y \in \{0,1\} : y \in \text{support}(v^\star), \text{ where } v^\star \text{ is some optimal mixed strategy for nature}\}.$$

Then define the pure* game and its value as follows,

$$\text{Val}^\star(q) := \min_{p \in \mathcal{X}^\star} \max_{y \in \mathcal{Y}^\star} \langle r(p,y), q \rangle - \epsilon \, \|q\|_\infty \, . \tag{8.12}$$

It can be shown that for any q, $\text{Val}(q) \leqslant \text{Val}^\star(q) \leqslant \text{Val}^\text{p}(q)$ (Mannor and Perchet, 2013). We now define mixed approachability.

Definition 8.2. An approachable set is said to be mixed approachable if there exists a $q \in \mathbb{R}^m$ such that while $\text{Val}(q) = 0$, $\text{Val}^\star(q) > 0$.

The following lemma witnesses a q that shows that the mixed approachability condition required by Theorem 8.4 is satisfied. The witnessed q in the proof is identified based on case (d) in the proof of Proposition 8.2.

Lemma 8.4. *There exists a $q \in \mathbb{R}^m$ such that Val$^\star(q) >$ Val$(q) = 0$. Thus B_ϵ is mixed approachable.*

Proof. Set $q_{1:m-1} = -1$ (i.e. $q_i = -1$ for all $i \in [m-1]$) and $q_m = 1$. Let us compute Val(q). The game for nature reduces to maximizing $(\sum_{i=1}^{m-1} u_i - u_m)v$ which can be done by playing $v = \mathbb{1}\left\{u_m \leqslant \sum_{i=1}^{m-1} u_i\right\}$. We perform case work to identify the optimal play for the forecaster.

- If $\sum_{i=1}^{m-1} u_i \leqslant u_m$, $v(u_m - \sum_{i=1}^{m-1} u_i) = 0$. The objective for the forecaster reduces to:

  $$\min_{u \in \Delta(\mathcal{X}), \sum_{i=1}^{m-1} u_i \leqslant u_m} \quad u_m M_m - \sum_{i=1}^{m-1} u_i M_i - \epsilon.$$

  Note that $0 < M_1 < M_2 < \ldots < M_m$. Thus the forecaster would set the minimum possible value of u_m, which under the constraints is equal to 0.5. For the second term, the largest coefficient of a u_i in the summation is M_{m-1}. Thus in order to minimize, the forecaster would set the maximum possible value of u_{m-1}, which under the constraints is $1 - u_m = 0.5$. We conclude that the minimum occurs at $u_m = u_{m-1} = 0.5$, $u_{i \notin \{m-1,m\}} = 0$. The objective value is equal to $0.5(M_m - M_{m-1}) - \epsilon = 0$.

- If $\sum_{i=1}^{m-1} u_i \geqslant u_m$, $v = 1$ is an optimal play for nature. The objective for the forecaster reduces to:

  $$\min_{u \in \Delta(\mathcal{X}), \sum_{i=1}^{m-1} u_i \geqslant u_m} \quad u_m(M_m - 1) - \sum_{i=1}^{m-1} u_i(M_i - 1) - \epsilon.$$

  As in the other case, this game is solved by observing that since $M_1 < M_2 < \ldots < M_m < 1$, the forecaster would want to put the maximum possible value on u_m which is 0.5 under the constraints. Among $u_{1:m-1}$, the multiplier of $(M_{m-1} - 1)$ hurts the least. Thus the forecaster sets $u_{m-1} = 0.5$ and $u_{i \notin \{m-1,m\}}$. The objective value at the minimum is equal to 0.

In each case, the forecaster's optimal play is $u_m = u_{m-1} = 0.5$, $u_{i \notin \{m-1,m\}} = 0$. This essentially makes nature's action irrelevant; nature's optimal response is any $v \in [0,1]$. Thus we have shown that $\text{Val}(q) = 0$, $\mathcal{X}^\star = \{M_{m-1}, M_m\}$, and $\mathcal{Y}^\star = \{0,1\}$. Let us now compute

$$\text{Val}^\star(q) = \min_{p \in \mathcal{X}^\star} \max_{y \in \mathcal{Y}^\star} \left(\mathbb{1}\left\{p = M_m\right\}(M_m - y) - \mathbb{1}\left\{p = M_{m-1}\right\}(M_{m-1} - y) - \epsilon \right).$$

If the forecast is $p = M_m$, nature can respond $y = 0$ to achieve an overall objective of $1 - 2\epsilon > 0$. (We have assumed $m \geqslant 2$ so that $\epsilon < 0.5$.) If the forecast is $p = M_{m=1}$, nature can respond $y = 1$ to achieve an overall objective of $2\epsilon > 0$. Thus, $\text{Val}^\star(q) > 0$.

□

8.4.6 Relationship to previous lower bounds for calibration

Qiao and Valiant (2021) recently constructed a strategy of nature that ensures that the calibration error of any forecaster playing Calibration-Game-I satisfies $\mathbb{E}\left[\text{CE}_T\right] = \Omega(T^{-0.472})$. This bound is interesting on its own and neither weaker nor stronger than the bound we show in Theorem 8.3. By studying $\mathbb{E}\left[\epsilon\text{-CE}_T\right]$, we allow the forecaster a slack of ϵ in his calibration error, which is standard in several earlier cited works (see Section 8.1.2), and may be sufficient in practice given that the forecasts are themselves on a 2ϵ-grid.

Qiao and Valiant (2021) also noted that $\mathbb{E}\left[\text{CE}_T\right] = \Omega(T^{-0.5})$ can be forced using a *Bernoulli strategy*: at each time step, nature plays $y_t \sim \text{Bernoulli}(p)$ for some fixed $p \in [0,1]$ unknown to the forecaster. However, in Appendix 8.B, we provide initial (but not conclusive) evidence that the Bernoulli strategy seems insufficient to guarantee $\mathbb{E}\left[\epsilon\text{-CE}_T\right] = \Omega(T^{-0.5})$. We construct an ϵ-calibrated forecaster that satisfies $\mathbb{E}\left[\text{CE}_T - \epsilon\right] \leqslant O(\text{poly}(\log T)/T)$ for the Bernoulli strategy ($\text{poly}(\log T)$ corresponds to polynomial terms in $\log(T)$). We conjecture that the stronger statement $\mathbb{E}\left[\epsilon\text{-CE}_T\right] = \mathbb{E}\left[\max(\text{CE}_T - \epsilon, 0)\right] \leqslant O(\text{poly}(\log T)/T)$ also holds, which would mean that the Bernoulli strategy is insufficient to derive a $\Omega(1/\sqrt{T})$ bound on the ϵ-calibration rate (as shown by Theorem 8.3).

8.5 Summary

This work connects three rich areas of the literature in a natural way: online calibration, the power of two choices, and imprecise probability. In summary, we show that by allowing the forecaster to output a deterministic short interval of probabilities (of length at most 2ϵ), we can achieve a faster rate of $O(1/T)$ for ϵ-calibration against a fully adaptive adversarial reality who presents the binary outcome after observing the interval forecast. This should be compared to the $\Theta(1/\sqrt{T})$ rate achievable with randomized point forecasts (the upper bound is a seminal result by Foster and Vohra (1998), the lower bound is ours), or the $\Theta(1)$ for deterministic point forecasts.

Arguably, such narrow intervals are as practically interpretable as point forecasts, and since some sort of binning anyway underlies most calibration algorithms, it also feels natural to allow the forecaster to express their uncertainty in this fashion, especially since it avoids randomization and improves calibration. Thus, we view our work as a theoretical contribution with clear practical implications.

Several open questions remain, since we open a rather new line of investigation. We mention two: (a) lower bounds for our setting are unknown, and (b) we don't know if models providing more than two choices could possibly improve the rate further. We suspect that $1/T$ is the optimal rate since it corresponds to constant cumulative calibration error (without normalization by T), which is incurred at $t = 1$ itself and seems unavoidable. Finally, it would be interesting to (c) figure out multidimensional analogs of our work. We leave these for future work.

We also note some POTC-style results in online learning. Neu and Zhivotovskiy (2020) show that for expert-based classification, providing the learner the choice to abstain from making a prediction improves the regret from $\Omega(1/\sqrt{T})$ to $O(1/T)$, similar to what we obtain in Theorem 8.1. In zero-order convex optimization or bandit convex optimization, allowing the learner two function evaluations enables the unknown gradient to be estimated using finite difference, leading to significantly improved rates (Agarwal et al. (2010) and follow-up work). For example, in the strongly convex case the rate improves from $\Omega(\sqrt{T})$ to $O(\log T)$. Such improvements also hold for non-smooth functions (Shamir, 2017). It would be interesting to consider POTC setups for multi-armed bandits (two arm-pulls instead of one) or expert-based online learning (choosing two experts instead of randomizing or choosing one expert).

Appendices for Chapter 8

8.A Supplementary results and deferred proofs

8.A.1 Lemma 8.5 with proof

Lemma 8.5. *In POTC-Cal, for any $t \geqslant 1$, if condition A is not satisfied, then condition B must be satisfied.*

Proof. Note that for all t, $d_1^t \leqslant 0$ and $e_m^t \leqslant 0$, since $l_1 = 0$ and $r_m = 1$ (there cannot be a deficit for interval 1 or an excess for interval m). If condition A does not hold for $i = m$, $d_m^t > 0$. Since $d_1^t \leqslant 0$ and $d_m^t > 0$, there exists an $i \in [m-1]$ such that, $d_i^t \leqslant 0$ and $d_{i+1}^t > 0$. If condition A does not hold, then $d_i^t \leqslant 0$ implies $e_i^t > 0$. Thus we have that $d_{i+1}^t > 0$ and $e_i^t > 0$; in other words, condition B holds at the exhibited i. □

8.A.2 Proposition 8.2 with proof

Proposition 8.2. *The forecaster in Calibration-Game-I can ensure halfspace-satisfiability. That is, for every $q \in \mathbb{R}^m$, Val$(q) \leqslant 0$.*

Proof. Our construction is directly inspired by the proof of calibration by Foster (1999). We perform a case analysis for different values of q:

(a) Suppose any $q_i = 0$. Then, playing $u_i = 1$ and $u_{j \neq i} = 0$ gives the objective value of $-\epsilon \left\| q \right\|_\infty \leqslant 0$ irrespective of the value of v.

(b) Suppose $q_1 > 0$. Then, playing $u_1 = 1$ and $u_{j>1} = 0$ gives the objective value as $q_1(M_1 - v) - \epsilon \left\| q \right\|_\infty \leqslant q_1\epsilon - \epsilon \left\| q \right\|_\infty \leqslant 0$ (note that $M_1 = \epsilon$ and $v \geqslant 0$).

(c) Suppose $q_m < 0$. Then, playing $u_m = 1$ and $u_{j<m} = 0$ gives the objective value as $q_m(M_m - v) - \epsilon \left\| q \right\|_\infty \leqslant |q_m|\,\epsilon - \epsilon \left\| q \right\|_\infty \leqslant 0$ (note that $M_m = 1 - \epsilon$ and $v \leqslant 1$).

(d) Suppose that neither of cases (a), (b), or (c) hold. Namely, $q_1 < 0$, $q_m > 0$ and $q_i \neq 0$ for any i. Let $j \in [m-1]$ be the smallest index such that $q_j < 0$ and $q_{j+1} > 0$. Then consider u given by

$$u_j = \frac{|q_{j+1}|}{|q_j| + |q_{j+1}|}, \; u_{j+1} = \frac{|q_j|}{|q_j| + |q_{j+1}|}, \; u_{i \notin \{j, j+1\}} = 0.$$

Observe two facts. First, $\sum_{i_1}^{m} u_i q_i v = (u_j q_j + u_{j+1} q_{j+1}) v = 0$ since the value inside the brackets is itself equal to 0 (any play v of nature is thus rendered ineffective). Second,

$$\begin{aligned}
\sum_{i_1}^{m} u_i q_i M_i &= u_j q_j M_j + u_{j+1} q_{j+1} M_{j+1} \\
&= (u_j q_j + u_{j+1} q_{j+1}) M_j + u_{j+1} q_{j+1} (M_{j+1} - M_j) \\
&= 0 + 2 u_{j+1} q_{j+1} \epsilon \\
&= \frac{2 \left| q_j \right| \left| q_{j+1} \right| \epsilon}{\left| q_j \right| + \left| q_{j+1} \right|} \\
&\leqslant \frac{\left\| q \right\|_\infty \left| q_j \right| \epsilon}{\left| q_j \right| + \left| q_{j+1} \right|} + \frac{\left\| q \right\|_\infty \left| q_{j+1} \right| \epsilon}{\left| q_j \right| + \left| q_{j+1} \right|} \\
&= \left\| q \right\|_\infty \epsilon.
\end{aligned}$$

Thus the overall objective value is at most 0.

The cases considered for q are exhaustive, and in each case we verified that the forecaster can guarantee that the objective value is at most 0. □

8.A.3 Proof of Proposition 8.1

The i-th component of the expected reward vector (8.8) is $u_i[(1 - v)M_i + v(M_i - 1)] = u_i(M_i - v)$. Suppose $v \in I_j$. Consider playing $u \in \Delta(\mathcal{X})$ given by $u_j = 1$, $u_{i \neq j} = 0$. Then $|\mathbb{E}_{p \sim u, y \sim v}\left[r(p, y)\right]| = |\sum_{i=1}^{m} u_i(M_i - v)| = |M_j - v|$. Since $v \in I_j$, M_j is the mid-point of I_j, and the radius of each interval is ϵ, $|M_j - v| \leqslant \epsilon$. Thus $\mathbb{E}_{u,v}\left[r(p, y)\right] \in B_\epsilon$.

8.B Bernoulli strategy seems insufficient to prove an $\Omega(1/\sqrt{T})$ lower bound on the ϵ-calibration rate

This section is in the setup of Calibration-Game-I. Nature plays the 'Bernoulli strategy': for a fixed $p \in [0, 1]$ (unknown to the forecaster), nature plays $y_t \sim$ Bernoulli(p) each time. We describe a strategy for the forecaster that satisfies $\mathbb{E}\left[\epsilon\text{-CE}_T\right] = O(1/\sqrt{T})$ against a general strategy of nature; however, if nature is playing the Bernoulli strategy (instead of an arbitrary strategy), then our proposed strategy satisfies $\mathbb{E}\left[\text{CE}_T - \epsilon\right] \leqslant O(\text{poly}(\log T)/T)$. Our result stops short of proving the following stronger statement against the Bernoulli strategy: $\mathbb{E}\left[\epsilon\text{-CE}_T\right] = \mathbb{E}\left[\max(\text{CE}_T - \epsilon, 0)\right] \leqslant O(\text{poly}(\log T)/T)$. We conjecture that this stronger statement holds as well, meaning that the Bernoulli strategy is insufficient to derive a $\Omega(1/\sqrt{T})$ bound on the ϵ-calibration rate as shown in Theorem 8.3.

```
PI-F99: pre-initialized version of the $\epsilon$-calibrated strategy by Foster (1999)
Fix $T_0 \in \mathbb{N}$. Set $T_k := 2^k T_0$, $K_k$ as in (8.13), $T^{(0)} := 0$, and $T^{(k)} := \sum_{j=1}^{k-1} T_j = (2^k - 1)T_0$.

For each time $t = 1, 2, \ldots$
    • Identify smallest $k \in \mathbb{N}$ such that $t \leqslant T^{(k)}$.
    • Play PI-F99$(T_{k-1})$ based on observations from $T^{(k-1)} + 1$ until $t$:
        ▪ if $t \leqslant T^{(k-1)} + mK_k$ (initialization phase):
                identify $j$ such that $t - T^{(k-1)} \in ((j-1)K_k + jK_k]$ and predict $M_j$.
        ▪ if $t > T^{(k-1)} + mK_k$:
                follow Foster's strategy based on observations from $T^{(k-1)} + 1$ until $t$.
```

8.B.1 The strategy

The strategy we propose is a pre-initialized version of the ϵ-calibration strategy of Foster (1999)—we call it PI-F99 (for pre-initialized-Foster-99). PI-F99 relies on a few constants: a large enough *doubling* horizon $T_0 \in \mathbb{N}$, *doubled* versions of T_0, namely $T_k =: 2^k T_0$ for $k \in \mathbb{N}$, and an initialization parameter

$$K_k := \left\lceil (0.85 \log T_k/\epsilon)^2 (\log\log(T_k/2) + 0.72 \log(5.2 m T_k^2)) \right\rceil \tag{8.13}$$

defined for each k. K_k is the sufficient number of samples required to estimate the bias of m Bernoulli random variables simultaneously and uniformly across time to a certain degree of reliability; further details become clear when analyzing. The constants in the definition of K_k are not crucial (looser constants still lead to the same asymptotic dependence on T), but we identify them nevertheless in order to be precise.

PI-F99 is a concatenation of certain sub-strategies PI-F99(T_k) for $k \in \mathbb{N}$, each of which are strategies for Calibration-Game-I assuming the game only goes on until time $t = T_k$. The forecaster playing PI-F99(T_k) first forecasts $p_t = M_i$, K_k times each, for each $i \in [m]$ (thus until time $t \leqslant mK_k$). This is the *initialization* phase. Then, for $t \in \{mK_k + 1, \ldots, T_k\}$, the forecaster follows Foster's strategy initialized with *current empirical frequencies* for each bin, based on what has been observed so far in the initialization phase.

The actual strategy of the forecaster corresponds to a concatenation of PI-F99(T_0), PI-F99(T_1), PI-F99(T_2), and so on. This is a version of the doubling trick (Cesa-Bianchi and Lugosi, 2006). The forecaster first plays PI-F99(T_0) from $t = 1$ to $t = T_0$, then plays PI-F99(T_1) from $t = T_0 + 1$ to $t = T_0 + T_1$, then PI-F99(T_2) and so on. To be clear, when switching from PI-F99(T_{k-1}) to PI-F99(T_k), the forecaster completely ignores the forecasts and observations so far, and restarts. The overall strategy is described in the box on top of the previous page.

8.B.2 Analysis

Ignoring terms in m and ϵ which are constants in T_k, $mK_k = O(\log^2 T_k) \ll \sqrt{T_k}$ (for sufficiently large T_0 and all $k \geqslant 0$). Assuming a worst case error of 1 for each time until $T \leqslant mK_k$, we can show that the ϵ-calibration error of $\texttt{PI-F99}(T_k)$ at any time $T \leqslant T_k$ satisfies the following:

$$\begin{aligned}
&\mathbb{E}\left[\max\left(\sum_{i=1}^{m}\left|\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\left\{p_t = M_i\right\}(M_i - y_t)\right| - \epsilon, 0\right)\right] \\
\leqslant\ &\mathbb{E}\left[\max\left(\sum_{i=1}^{m}\left|\frac{1}{T}\sum_{t=1}^{mK_k}\mathbb{1}\left\{p_t = M_i\right\}(M_i - y_t)\right| + \sum_{i=1}^{m}\left|\frac{1}{T}\sum_{t=mK_k+1}^{T}\mathbb{1}\left\{p_t = M_i\right\}(M_i - y_t)\right| - \epsilon, 0\right)\right] \\
\leqslant\ &\frac{mK_k}{T} + \mathbb{E}\left[\max\left(\sum_{i=1}^{m}\left|\frac{1}{T}\sum_{t=mK_k+1}^{T}\mathbb{1}\left\{p_t = M_i\right\}(M_i - y_t)\right| - \epsilon, 0\right)\right] \\
\leqslant\ &\frac{O(\sqrt{T_k})}{T} + \mathbb{E}\left[\max\left(\sum_{i=1}^{m}\left|\frac{1}{T}\sum_{t=mK_k+1}^{T}\mathbb{1}\left\{p_t = M_i\right\}(M_i - y_t)\right| - \epsilon, 0\right)\right].
\end{aligned}$$

To bound the second term, note that Foster's strategy has a calibration rate of $O(1/\sqrt{T})$ starting with any initialization. Thus,

$$\mathbb{E}\left[\max\left(\sum_{i=1}^{m}\left|\frac{1}{T}\sum_{t=mK_k+1}^{T}\mathbb{1}\left\{p_t = M_i\right\}(M_i - y_t)\right| - \epsilon, 0\right)\right] \leqslant \frac{O(\sqrt{T - mK_k})}{T} \leqslant \frac{C\sqrt{T_k}}{T}, \tag{8.14}$$

for some constant C independent of T_0 and k. Thus for $\texttt{PI-F99}(T_k)$ at any time $T \leqslant T_k$, we have shown that $\mathbb{E}\left[\epsilon\text{-CE}_T\right] \leqslant O(1/\sqrt{T})$. From this, it follows that the overall strategy $\texttt{PI-F99}$ satisfies $f(T) = O(1/\sqrt{T})$ asymptotically. This is shown in Proposition 8.4, later in this subsection. Next, we perform the analysis for the Bernoulli strategy.

Proposition 8.3. *PI-F99 satisfies* $\mathbb{E}\left[CE_T\right] \leqslant \epsilon + O(\mathit{poly}(\log T)/T)$ *if nature follows the Bernoulli strategy.*

Proof. Following the notation of Section 8.3, let N_i^T denote the number of times the mid-point M_i is forecasted until time T. We have,

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{m}\left|\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\left\{p_t = M_i\right\}(M_i - y_t)\right|\right] = \\
&\underbrace{\frac{\sum_{i\in[m]:|p-M_i|>\epsilon}\mathbb{E}\left[N_i^T\left|M_i - \sum_{t=1}^{T}\mathbb{1}\left\{p_t = M_i\right\}y_t/N_i^T\right|\right]}{T}}_{E_1} \\
+\ &\underbrace{\frac{\sum_{i\in[m]:|p-M_i|\leqslant\epsilon}\mathbb{E}\left[N_i^T\left|M_i - \sum_{t=1}^{T}\mathbb{1}\left\{p_t = M_i\right\}y_t/N_i^T\right|\right]}{T}}_{E_2}.
\end{aligned}$$

We will show that

$$E_1 = O(\text{poly}(\log T)/T),$$

and

$$E_2 = \epsilon + O(1/T),$$

which will complete the argument. To this end, define A^T as the number of times that the forecast is ϵ-close to p, until time T:

$$A^T := \sum_{i \in [m] : |p - M_i| \leqslant \epsilon} N_i^T. \tag{8.15}$$

Lemma 8.6 shows that `PI-F99` satisfies $\mathbb{E}\left[A^T\right] = T - O(\text{poly}(\log T))$. This immediately leads to the bound for E_1; note that $\left|M_i - \sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_i\right\} y_t / N_i^T\right| \leqslant 1$, and thus

$$E_1 \leqslant \frac{\sum_{i \in [m] : |p - M_i| > \epsilon} \mathbb{E}\left[N_i^T\right]}{T} = 1 - \frac{\mathbb{E}\left[A^T\right]}{T} = O(\text{poly}(\log T)/T).$$

Bounding E_2 takes more work. The proof relies on the following 'good' event occurring with high probability:

$$G \equiv G_T : \text{for every } i \in [m], \left|p - \sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_i\right\} y_t / N_i^T\right| \leqslant \epsilon/2.$$

Due to the pre-initialization steps in `PI-F99`, it can be guaranteed that $\Pr(G) = \Pr(G_T) = 1 - O(1/T)$. For the details, we refer the reader to the proof of Lemma 8.6 (see case (a) in the proof), where we show a stronger version of this fact (namely, with $\epsilon/\log T_k$ instead of $\epsilon/2$), for `PI-F99`(T_k) using a time-uniform concentration inequality (due to the time uniformity, the implication holds for `PI-F99` as well). We now do case work to bound E_2.

(a) Suppose there exists an index $j \in [m]$ such that $|p - M_j| \leqslant \epsilon/2$. This index must be unique since the M_j's are 2ϵ apart. Further, no $i \neq j$ can satisfy $|p - M_i| \leqslant \epsilon$. We now obtain the following (below, we use $\text{abs}(\cdot)$ instead of $|\cdot|$ to avoid confusion with the conditioning operator):

$$\begin{aligned}
T \cdot E_2 &= \mathbb{E}\left[N_j^T \text{abs}(M_j - \sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_j\right\} y_t / N_j^T)\right] \\
&\leqslant \mathbb{E}\left[N_j^T \text{abs}(M_j - \sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_j\right\} y_t / N_j^T) \mid G\right] + (1 - \Pr(G)) \cdot T \\
&= \mathbb{E}\left[N_j^T \text{abs}(M_j - \sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_j\right\} y_t / N_j^T) \mid G\right] + O(1) \\
&\leqslant T \cdot \mathbb{E}\left[\text{abs}(M_j - \sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_j\right\} y_t / N_j^T) \mid G\right] + O(1)
\end{aligned}$$

$$\leqslant T \cdot \mathbb{E}\left[|M_j - p| + \text{abs}(p - \sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_i\right\} y_t/N_j^T) \mid G \right] + O(1)$$
$$\leqslant T(\epsilon/2 + \epsilon/2) + O(1) = T\epsilon + O(1),$$

where the inequality in the last line follows by the case assumption and the definition of G. Thus for this case, we have shown that $E_2 = \epsilon + O(1/T)$.

(b) Suppose p is such that for every i, $|p - M_i| > \epsilon/2$. To bound E_2, we are interested in the M_i's for which $|p - M_i| \leqslant \epsilon$. There can be at most two such M_i's: an M_j that satisfies $M_j \in (p + \epsilon/2, p + \epsilon]$, and an M_l that satisfies $M_j \in [p - \epsilon, p - \epsilon/2)$.

Suppose there is an M_j satisfying $M_j \in (p+\epsilon/2, p+\epsilon]$. Set $R = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_j\right\}(M_j - y_t)$ and note that $R \in [-1, 1]$. By Lemma 8.7, $\mathbb{E}\left[|R|\right] \leqslant \mathbb{E}\left[R\right] + 2 \cdot \Pr(R < 0)$. Note that $\Pr(R \geqslant 0) \geqslant \Pr(G)$, since if G holds,

$$\begin{aligned}
R \cdot T &= \sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_j\right\}(M_j - y_t) \\
&= \sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_j\right\}((M_j - p) + (p - y_t)) \\
&\geqslant \sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_j\right\}(\epsilon/2 + (p - y_t)) \\
&= N_i^T \epsilon/2 + \sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_j\right\}(p - y_t) \\
&\geqslant N_i^T \epsilon/2 - \left|\sum_{t=1}^{T} \mathbb{1}\left\{p_t = M_j\right\}(p - y_t)\right| \\
&\geqslant N_i^T \epsilon/2 - N_i^T \epsilon/2 = 0,
\end{aligned}$$

where the inequality in the last line is implied by G. Thus, $\Pr(R < 0) \leqslant 1 - \Pr(G) = O(1/T)$. Next, we bound $\mathbb{E}\left[R\right]$.

$$\begin{aligned}
\mathbb{E}\left[R\right] &= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{1}\left\{p_t = M_j\right\}(M_j - y_t)\right] \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{1}\left\{p_t = M_j\right\}(M_j - \mathbb{E}\left[y_t \mid (y_1, \ldots, y_{t-1}), (p_1, \ldots, p_t)\right])\right] \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{1}\left\{p_t = M_j\right\}(M_j - p)\right] \\
&= \frac{\mathbb{E}N_j^T(M_j - p)}{T} \leqslant \frac{\mathbb{E}N_j^T}{T} \cdot \epsilon.
\end{aligned}$$

Putting it together, we obtain $\mathbb{E}\left[|R|\right] \leqslant \frac{\mathbb{E}N_j^T}{T} \cdot \epsilon + O(1/T)$.

Similarly, suppose there is an M_l satisfying $M_l \in [p-\epsilon, p-\epsilon/2)$. For this l, define $S = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\{p_t = M_l\}(M_l - y_t)$. An identical argument as the one used for R goes through; we use the inequality $\mathbb{E}[|S|] \leqslant \mathbb{E}[-S] + 2 \cdot \Pr(S > 0)$ (from Lemma 8.7) and the relationship of $\Pr(S > 0)$ to G to obtain $\mathbb{E}[|S|] \leqslant \frac{\mathbb{E}N_l^T}{T} \cdot \epsilon + O(1/T)$.

Finally, we conclude if both M_k and M_l with the given relationship to p exist, then $E_2 \leqslant \mathbb{E}[|R| + |S|]$; if only M_j exists, then $E_2 \leqslant \mathbb{E}[|R|]$; if only M_l exists, then $E_2 \leqslant \mathbb{E}[|S|]$. In each case,

$$E_2 \leqslant \frac{\mathbb{E}\left[A^T\right] \cdot \epsilon}{T} + O(1/T) \leqslant \epsilon + O(1/T).$$

Since the two cases considered are exhaustive, this completes the proof.

□

Lemma 8.6 (for proving Proposition 8.3)**.** *PI-F99 satisfies* $\mathbb{E}\left[A^T\right] = T - O(\textit{poly}(\log T))$*, where* A^T *is defined in the proof of Proposition 8.3.*

Proof. We first show PI-F99(T_k) satisfies $\mathbb{E}\left[A^T\right] = T - O(\text{poly}(\log T))$ for $T \leqslant T_k$ once k is large enough. We do so via two cases.

(a) For the first case, suppose $p = r_j = l_{j+1}$ for some $j \in [m-1]$, that is, the bias of the Bernoulli is exactly at the common endpoint of two intervals. In other words, $A^T = N_j^T + N_{j+1}^T$. We show that with high probability, the forecaster will *learn* this index j in the initialization phase of each PI-F99(T_k), and continue playing either M_j or M_{j+1} until he switches to PI-F99(T_{k+1}).

Consider the strategy PI-F99(T_k) for some $k \geqslant 0$. From time $t = mK_k$ onwards, each M_i has been forecasted at least K_k times, so that the value of p_i^t is *close* to p. To formalize close, we will use a time-uniform sub-Gaussian concentration inequality shown by Howard et al. (2021, equation (3.4)). We use their inequality, replacing each instance of t with $K_k/4$, since a Bernoulli is $(1/4)$-sub-Gaussian and each M_i has been forecasted at least K_k times. Additionally, we replace α with $1/mT_k^2$. It can be verified that the final deviation term inside the brackets is at most $\epsilon/\log T_k$; in other words, with probability at least $1 - 1/T_k^2$, the following 'good' event occurs:

$$\text{for all times } mK_k \leqslant t \leqslant T_k, \max_{i\in[m]} \left|p_i^t - p\right| \leqslant \epsilon/\log T_k \leqslant \epsilon.$$

The radius of each interval is ϵ. Thus if the above event occurs, it follows that for intervals $i < j$, the right-endpoint $r_i < p - \epsilon \leqslant p_i^t$, so we have an excess ($e_i^T > 0$) until T_k; and for intervals $i > j+1$, the left-endpoint $l_i > p + \epsilon \geqslant p_i^t$, so we have a deficit ($d_i^T > 0$) until T_k. For interval j, either both $d_j^t, e_j^t \leqslant 0$ or $e_j^t > 0$; for interval $j+1$, either both $d_j^t, e_j^t \leqslant 0$ or $d_j^t > 0$. Overall, with probability at least $1 - 1/T_k^2$, for times $mK_k < t \leqslant T_k$, Foster's algorithm randomizes between M_j and M_{j+1} (possibly playing one of them deterministically).

(b) The other case is when p belongs to the interior of some interval I_j, $j \in [m]$, or $p \in \{0,1\}$. Then, there exists some $\delta > 0$ such that $|p - M_i| \geqslant \epsilon + \delta$ for all $i \neq j$. For a sufficiently large value of $\widetilde{k} \in \mathbb{N}$, $\delta > \epsilon/\log(T_{\widetilde{k}})$. Consider the strategy $\texttt{PI-F99}(T_k)$ for $k \geqslant \widetilde{k}$. As noted in the previous case, our choice of K_k ensures that with probability at least $1 - 1/T_k^2$, for all times $mK_k \leqslant t \leqslant T_k$, $\max_{i\in[m]} |p_i^t - p| \leqslant \epsilon/\log T_k < \delta$. Using triangle inequality, we conclude that $\left|p_j^t - M_j\right| \leqslant \epsilon$ and $|p_i^t - M_{i\neq j}| > \epsilon$. It follows that for every $i \neq j$, there is either a deficit or an excess, and for j there is neither. Thus with probability at least $1 - 1/T_k^2$, Foster's algorithm plays M_j after time mK_k.

Cases (a) and (b) lead to a lower bound on $\mathbb{E}\left[A^T\right]$ for $\texttt{PI-F99}(T_k)$, the expected number of times an M_i is forecasted that is ϵ-close to p. Namely, we obtain that for $k \geqslant \widetilde{k}$, for the strategy $\texttt{PI-F99}(T_k)$ that plays assuming a horizon of T_k from $t = 1$ itself, we have for $T \leqslant T_k$:

$$\begin{aligned}\mathbb{E}\left[A^T\right] &\geqslant (1 - 1/T_k^2)(T - mK_k) \\ &\geqslant T(1 - 1/T_k^2) - O(\text{poly}(\log T_k)) \\ &\geqslant T - O(\text{poly}(\log T)). \end{aligned} \tag{8.16}$$

The final inequality above holds since $T \leqslant T_k$.

We derive the implication for the overall strategy that is actually played, $\texttt{PI-F99}$. Recall the notation $T^{(0)} = 0$ and $T^{(k)} = T_0 + T_1 + \ldots + T_{k-1}$. In $\texttt{PI-F99}$, the $\texttt{PI-F99}(T_k)$ strategy is played from time $T^{(k-1)} + 1$ to time $T^{(k-1)} + T_k = T^{(k)}$. Let T be such that $T \in [T^{(k)} + 1, T^{(k+1)}]$ for any $k \geqslant \widetilde{k}$. Then by (8.16),

$$\mathbb{E}\left[A^T - A^{T^{(k)}}\right] \geqslant T - T^{(k)} - O(\text{poly}(\log T)).$$

Again by (8.16), the above holds with $T \leftarrow T^{(k)}$, $T^{(k)} \leftarrow T^{(k-1)}$, if $k \geqslant \widetilde{k} + 1$ ($\leftarrow$ corresponds to replacing the term on the left with the term on the right):

$$\mathbb{E}\left[A^{T^{(k)}} - A^{T^{(k-1)}}\right] \geqslant T^{(k)} - T^{(k-1)} - O(\text{poly}(\log T)).$$

Instantiating this recursively for all $k \geqslant \widetilde{k} + 1$, and adding the inequalities together gives us:

$$\mathbb{E}\left[A^T\right] \geqslant T - T^{(\widetilde{k})} - \log(T) \cdot O(\text{poly}(\log T)) = T - O(\text{poly}(\log T)),$$

since $\widetilde{k}$ is some fixed constant (given p). This completes the argument.

□

Lemma 8.7. *For any bounded random variable $R \in [-a, a]$,*

$$\mathbb{E}\left[|R|\right] \leqslant \min(\mathbb{E}\left[R\right] + 2a \cdot Pr(R < 0), \mathbb{E}\left[-R\right] + 2a \cdot Pr(R > 0)). \tag{8.17}$$

Proof. Note that,

$$\mathbb{E}\left[|R|\right] = \mathbb{E}\left[R \cdot \mathbb{1}\left\{R \geqslant 0\right\} - R \cdot \mathbb{1}\left\{R < 0\right\}\right]$$

$$
\begin{aligned}
&= \mathbb{E}\left[R \cdot \mathbb{1}\{R \geqslant 0\} + R \cdot \mathbb{1}\{R < 0\} - 2R \cdot \mathbb{1}\{R < 0\}\right] \\
&= \mathbb{E}\left[R - 2R \cdot \mathbb{1}\{R < 0\}\right] \\
&\leqslant \mathbb{E}\left[R + 2a \cdot \mathbb{1}\{R < 0\}\right] \\
&= \mathbb{E}\left[R\right] + 2a \cdot \Pr(R < 0).
\end{aligned}
$$

In the above proof, we can replace R with $-R$ everywhere, since $-R \in [-a, a]$ as well. Thus we also obtain,

$$\mathbb{E}\left[|R|\right] = \mathbb{E}\left[|-R|\right] \leqslant \mathbb{E}\left[-R\right] + 2a \cdot \Pr(R > 0).$$

□

Proposition 8.4. *PI-F99 achieves a calibration rate of $O(1/\sqrt{T})$ against any strategy of nature.*

Proof. Define $T^{(0)} = 0$ and $T^{(k)} = T_0 + T_1 + \ldots + T_{k-1} = (2^k - 1)T_0$, for $k \geqslant 1$. Further, define the cumulative (non-normalized) calibration error corresponding only to the times $t = t_1 + 1$ to t_2 as follows:

$$\mathrm{CE}(t_1, t_2) := \mathbb{E}\left[\max\left(\sum_{i=1}^{m}\left|\sum_{t=t_1+1}^{t_2} \mathbb{1}\{p_t = M_i\}(M_i - y_t)\right|, \epsilon(t_2 - t_1)\right)\right].$$

From (8.14), PI-F99 satisfies, for every $k \in \mathbb{N}$ and $T^{(k-1)} < t \leqslant T^{(k)}$,

$$\mathrm{CE}(T^{(k-1)}, t) \leqslant \epsilon(t - T^{(k-1)}) + C\sqrt{T_{k-1}} \tag{8.18}$$

for some universal constant C that does not depend on k.

Now for a given $T > T^{(2)}$, let $k \geqslant 3$ be such that $T^{(k-1)} < T \leqslant T^{(k)}$. By triangle inequality and (8.18),

$$
\begin{aligned}
&\mathbb{E}\left[\max\left(\sum_{i=1}^{m}\left|\sum_{t=1}^{T} \mathbb{1}\{p_t = M_i\}(M_i - y_t)\right|, \epsilon T\right)\right] \\
&\leqslant \sum_{i=1}^{k-1} \mathrm{CE}(T^{(i-1)}, T^{(i)}) + \mathrm{CE}(T^{(k-1)}, T) \\
&\leqslant \sum_{i=1}^{k-1}(\epsilon(T^{(i)} - T^{(i-1)}) + C\sqrt{T_{i-1}}) \\
&\qquad\qquad + \epsilon(T - T^{(k-1)}) + C\sqrt{T_{k-1}} \\
&= \epsilon T + \sum_{i=1}^{k} C\sqrt{T_{i-1}} \\
&= \epsilon T + C(\sqrt{T_0} + \sqrt{2T_0} + \sqrt{4T_0} + \ldots + \sqrt{2^{k-1}T_0}) \\
&\leqslant \epsilon T + C\sqrt{T_0} \cdot \frac{\sqrt{2^k} - 1}{\sqrt{2} - 1}
\end{aligned}
$$

$$
\begin{aligned}
&\leqslant \epsilon T + C \cdot \frac{\sqrt{2^k T_0}}{\sqrt{2}-1}. \\
&\leqslant \epsilon T + C'\sqrt{T},
\end{aligned}
$$

where $C' = C \cdot 2/(\sqrt{2}-1)$. The final inequality holds since for $k \geqslant 3$,

$$\sqrt{2^k T_0} \leqslant \sqrt{4(2^{k-1}-1)T_0} = \sqrt{4T^{(k-1)}} < 2\sqrt{T}.$$

Dividing by T and taking ϵ to the left-hand-size, we get that for all $T > T^{(2)}$,

$$\mathbb{E}\left[\max\left(\sum_{i=1}^{m}\left|\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\left\{p_t = M_i\right\}(M_i - y_t)\right| - \epsilon, 0\right)\right] \leqslant C'\sqrt{T} = O(1/\sqrt{T}),$$

as needed. □

8.C Generalization of POTC-Cal to bounded outputs

If the output is bounded instead of binary (see Remark 8.1), then POTC-Cal can be modified as follows. The forecaster maintains p_i^T as in the original algorithm, but these are now the mean of the v_t values instead of y_t values. The choice of the index i and the final forecast (p_{t0}, p_{t1}) is made identically to the original POTC-Cal. Finally, the forecaster plays

$$p_t = p_{t0} \text{ if } v_t \leqslant r_i, \text{ and } p_t = p_{t1} \text{ if } v_t > r_i. \tag{8.19}$$

Note that $r_i = l_{i+1}$ is the right (left) endpoint of interval i (interval $i+1$), and thus a natural threshold for deciding which of the two intervals to play.

Lemmas 8.1 and 8.2 hold for this modified setup and algorithm, and thus the $O(1/T)$ rate showed by Theorem 8.1 also holds. Lemma 8.1 goes through since the set of equations (8.6) can be modified as follows:

$$\left|d_i^{t+1} - d_i^t\right| = \left|e_i^{t+1} - e_i^t\right| = \left|\frac{v_{t+1} - p_i^t}{N_i^t + 1}\right| \leqslant \frac{1}{N_i^{t+1}}.$$

In the proof of Lemma 8.2, we assumed without loss of generality that $y_{t+1} = 0$. This assumption can be modified to $v_{t+1} \leqslant r_i$ in keeping with the forecaster's updated strategy (8.19). The case $e_i^{t+1} < d_i^{t+1}$ goes through since it is a consequence of the set of equations (8.6). For the case $e_i^{t+1} \geqslant d_i^{t+1}$, we have

$$
\begin{aligned}
N_i^{t+1}\max(d_i^{t+1}, e_i^{t+1}) &= N_i^t p_i^t + v_{t+1} - N_i^{t+1} r_i \\
&= N_i^t e_i^t + (v_{t+1} - r_i) \leqslant N_i^t \max(d_i^t, e_i^t),
\end{aligned}
$$

where the last inequality follows by the case assumption $v_{t+1} \leqslant r_i$.

Chapter 9

Nested conformal prediction and quantile out-of-bag ensemble methods

This chapter is based on Gupta et al. (2022).

Conformal prediction is a popular tool for providing valid prediction sets for classification and regression problems, without relying on any distributional assumptions on the data. While the traditional description of conformal prediction starts with a nonconformity score, we provide an alternate (but equivalent) view that starts with a sequence of nested sets and calibrates them to find a valid prediction set. The nested framework subsumes all nonconformity scores, including recent proposals based on quantile regression and density estimation. While these ideas were originally derived based on sample splitting, our framework seamlessly extends them to other aggregation schemes like cross-conformal, jackknife+ and out-of-bag methods. We use the framework to derive a new algorithm (QOOB, pronounced cube) that combines four ideas: quantile regression, cross-conformalization, ensemble methods and out-of-bag predictions. We develop a computationally efficient implementation of cross-conformal, that is also used by QOOB. In a detailed numerical investigation, QOOB performs either the best or close to the best on all simulated and real datasets.

9.1 Introduction

Traditional machine learning algorithms provide point predictions, such as mean estimates for regression and class labels for classification, without conveying uncertainty or confidence. However, sensitive applications like medicine and finance often require valid uncertainty estimates. In this work, we discuss quantification of predictive uncertainty through predictive inference, wherein we provide prediction sets rather than point predictions.

Formally, the problem of distribution-free predictive inference is described as follows: given dataset $D_n \equiv \{(X_i, Y_i)\}_{i=1}^n$ drawn i.i.d. from $P_{XY} = P_X \times P_{Y|X}$ on $\mathcal{X} \times \mathcal{Y}$[1], and $X_{n+1} \sim P_X$,

[1] The spaces $\mathcal{X}$ and $\mathcal{Y}$ are without restriction. For example, take $\mathcal{X} \equiv \mathbb{R}^d$ and let $\mathcal{Y}$ be a subset of $\mathbb{R}$, or a discrete space such as in multiclass classification. Though it is not formally required, it may be helpful think of $\mathcal{Y}$ as a totally ordered set or a metric space.

we must construct a prediction set $C(D_n, \alpha, X_{n+1}) \equiv C(X_{n+1})$ for Y_{n+1} that satisfies:

$$\text{for any distribution } P_{XY},\ \mathbb{P}(Y_{n+1} \in C(X_{n+1})) \geqslant 1 - \alpha. \tag{9.1}$$

Here the probability is taken over all $n+1$ points and $\alpha \in (0,1)$ is a predefined confidence level. As long as (9.1) is true, the 'size' of $C(X_{n+1})$ conveys how certain we are about the prediction at X_{n+1}. Methods with property (9.1) are called *marginally* valid to differentiate them from *conditional* validity:

$$\forall P_{XY},\ \mathbb{P}(Y_{n+1} \in C(X_{n+1}) \mid X_{n+1} = x) \geqslant 1 - \alpha,\ \text{for } P_X\text{-almost all } x.$$

Conditional validity is known to be impossible without assumptions on P_{XY}; see Balasubramanian et al. (2014, Section 2.6.1) and Barber et al. (2020). If $Y_{n+1} \in C(X_{n+1})$, we say that $C(X_{n+1})$ covers Y_{n+1}, and we often refer to marginal (conditional) validity as marginal (conditional) coverage. In this work, we develop methods that are (provably) marginally valid, but have reasonable conditional coverage in practice, using conformal prediction.

Conformal prediction is a universal framework for constructing marginally valid prediction sets. It acts like a wrapper around any prediction algorithm; in other words, any black-box prediction algorithm can be "conformalized" to produce valid prediction sets instead of point predictions. We refer the reader to the works of Vovk et al. (2005a) and Balasubramanian et al. (2014) for details on the original framework. In this work, we provide an alternate and equivalent viewpoint for accomplishing the same goals, called *nested conformal prediction.*

Conformal prediction starts from a nonconformity score. As we will see with explicit examples later, these nonconformity scores are often rooted in some underlying geometric intuition about how a good prediction set may be discovered from the data. Nested conformal acknowledges this geometric intuition and makes it explicit: instead of starting from a score, it instead starts from a sequence of all possible prediction sets $\{\mathcal{F}_t(x)\}_{t\in\mathcal{T}}$ for some ordered set $\mathcal{T}$. Just as we suppressed the dependence of set $C(\cdot)$ on the labeled data D_n in property (9.1), here too $\mathcal{F}_t(\cdot)$ will actually depend on D_n but we suppress this for notational simplicity.

These prediction sets are 'nested', that is, for every $t_1 \leqslant t_2 \in \mathcal{T}$, we require that $\mathcal{F}_{t_1}(x) \subseteq \mathcal{F}_{t_2}(x)$; also $\mathcal{F}_{\inf \mathcal{T}} = \varnothing$ and $\mathcal{F}_{\sup \mathcal{T}} = \mathcal{Y}$. Thus large values of t correspond to larger prediction sets. Given a tolerance $\alpha \in [0,1]$, we wish to identify the smallest $t \in \mathcal{T}$ such that

$$\mathbb{P}(Y \in \mathcal{F}_t(X)) \geqslant 1 - \alpha.$$

In a nutshell, nested conformal learns a data-dependent mapping $\alpha \rightarrow t(\alpha)$ using the conformal principle. Note that the mapping must be decreasing; lower tolerance values α naturally lead to larger prediction sets.

We now briefly describe the steps involved in split/inductive conformal prediction Papadopoulos et al., 2002, Lei et al., 2018, Balasubramanian et al., 2014, Chapter 2.3, and use it to illustrate the nested principle. First, split D_n into a training set $D_1 \equiv \{(X_i, Y_i)\}_{1\leqslant i\leqslant m}$ and a calibration set $D_2 \equiv \{(X_i, Y_i)\}_{m<i\leqslant n}$. Using D_1, construct an estimate $\widehat{\mu}(\cdot)$ of the conditional mean of Y given X. Then construct the nonconformity score as the residuals of $\widehat{\mu}$ on D_2: $r_i := |Y_i - \widehat{\mu}(X_i)|$, for $i \in D_2$. Finally, define

$$C(X_{n+1}) = \Big\{ y \in \mathbb{R} : |y - \widehat{\mu}(X_{n+1})| < Q_{1-\alpha}(\{r_i\}_{i\in D_2}) \Big\},$$

where $Q_{1-\alpha}(A)$ for a finite set A represents the $(1-\alpha)$-th quantile of elements in A. Due to the exchangeability of order statistics, $C(\cdot)$ can be shown to be marginally valid (see Proposition 9.1).

We now give an alternate derivation of the above set using nested conformal:

1. After learning $\widehat{\mu}$ using D_1 (as done before), construct a sequence of nested prediction sets corresponding to symmetric intervals around $\widehat{\mu}(\cdot)$:
$$\{\mathcal{F}_t(\cdot)\}_{t\geqslant 0} := \{[\widehat{\mu}(\cdot)-t, \widehat{\mu}(\cdot)+t] : t \geqslant 0\}.$$
Note that $\mathcal{F}_t(\cdot)$ is a random set since it is based on $\widehat{\mu}(\cdot)$ which is random through D_1. It is clear that regardless of $\widehat{\mu}$, for any distribution of (X, Y), and any $\alpha \in [0,1]$, there exists a (minimal) $t = t(\alpha)$ such that $\mathbb{P}(Y \in \mathcal{F}_t(X)) \geqslant 1-\alpha$. Hence we can rewrite our nested family as
$$\Big\{[\widehat{\mu}(\cdot)-t, \widehat{\mu}(\cdot)+t] : t \geqslant 0\Big\} = \Big\{[\widehat{\mu}(\cdot)-t(\alpha), \widehat{\mu}(\cdot)+t(\alpha)] : \alpha \in [0,1]\Big\}.$$

2. The only issue now is that we do not know the map $\alpha \mapsto t(\alpha)$, that is, given α we do not know which of these prediction intervals to use. Hence we use the calibration data to "estimate" the map $\alpha \to t(\alpha)$. This is done by finding the smallest t such that $\mathcal{F}_t(X_i)$ contains Y_i for at least $1-\alpha$ fraction of the calibration points (X_i, Y_i) (we provide formal details later). Because the sequence $\{\mathcal{F}_t(\cdot)\}_{t\geqslant 0}$ is increasing in t, finding the smallest t leads to the smallest prediction set within the nested family.

Embedding nonconformity scores into our nested framework allows for easy comparison between the geometric intuition of the scores; see Table 9.1. Further, this interpretation enables us to extend these nonconformity scores beyond the split/inductive conformal setting that they were originally derived in. Specifically, we seamlessly derive cross-conformal, jackknife+ and OOB versions of these methods, including our new method called QOOB (pronounced cube).

A final reason that the assumption of nestedness is natural is the fact that the optimal prediction sets are nested: Suppose $Z_1, \ldots, Z_n$ are exchangeable random variables with a common distribution that has density $p(\cdot)$ with respect to some underlying measure. The "oracle" prediction set (Lei et al., 2013) for a future observation Z_{n+1} is given by the *level set* of the density with valid coverage, that is, $\{z : p(z) \geqslant t(\alpha)\}$ with $t(\alpha)$ defined by largest t such that $\mathbb{P}(p(Z_{n+1}) \geqslant t) \geqslant 1-\alpha$. Because $\{z : p(z) \geqslant t\}$ is decreasing with t, $\{z : p(z) \geqslant t(\alpha)\}$ is decreasing with $\alpha \in [0,1]$, forming a nested sequence of sets. See 9.F for more details.

9.1.1 Organization and contributions

For simplicity, our discussion focuses on the regression setting: $\mathcal{Y} = \mathbb{R}$. However, all ideas are easily extended to other prediction settings including classification. The chapter is organized as follows:

1. In Section 9.2, we formalize the earlier discussion and present split/inductive conformal (Papadopoulos et al., 2002; Lei et al., 2018) in the language of nested conformal prediction, and translate various conformity scores developed in the literature for split conformal into nested prediction sets.

2. In Section 9.3, we rephrase the jackknife+ (Barber et al., 2021) and cross-conformal prediction (Vovk, 2015) in terms of the nested framework. This allows the jackknife+ to use many recent score functions, such as those based on quantiles, which were originally developed and deployed in the split framework. In Section 9.3.3 we provide an efficient implementation of cross-conformal that matches the jackknife+ prediction time for a large class of nested sets that includes all standard nested sets.
3. In Section 9.4, we extend the Out-of-Bag conformal (Johansson et al., 2014) and jackknife+ after bootstrap (Kim et al., 2020) methods to our nested framework. These are based on ensemble methods such as random forests, and are relatively computationally efficient because only a single ensemble needs to be built.
4. In Section 9.5, we consolidate the ideas developed in this work to construct a novel conformal method called QOOB (Quantile Out-of-Bag, pronounced cube), that is both computationally and statistically efficient. QOOB combines four ideas: quantile regression, cross-conformalization, ensemble methods and out-of-bag predictions. Section 9.6 demonstrates QOOB's strong empirical performance.

In 9.A, we show that nested conformal is equivalent to the standard conformal prediction based on nonconformity scores. We also formulate full transductive conformal prediction in the nested framework. In 9.B we derive K-fold variants of jackknife+/cross-conformal in the nested framework and in 9.C, we develop the other aggregated conformal methods of subsampling and bootstrap in the nested framework. In 9.D, we discuss cross-conformal computation and the jackknife+ in the case when our nested sequence could contain empty sets. This is a subtle but important issue to address when extending these methods to quantile-based nested sets of Romano et al. (2019), and thus relevant to QOOB as well. 9.E contains all proofs.

9.2 Split conformal based on nested prediction sets

In the introduction, we showed that in a simple regression setup with the nonconformity scores as held-out residuals, split conformal intervals can be naturally expressed in terms of nested sets. Below, we introduce the general nested framework and recover the usual split conformal method with general scores using this framework. We show how existing nonconformity scores in literature exhibit natural re-interpretations in the nested framework. The following description of split conformal follows descriptions by Papadopoulos et al. (2002) and Lei et al. (2018) but rewrites it in terms of nested sets.

Suppose $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}, i \in [n]$ denotes the training dataset. Let $[n] = \mathcal{I}_1 \cup \mathcal{I}_2$ be a partition of $[n]$. For $\mathcal{T} \subseteq \mathbb{R}$ and each $x \in \mathcal{X}$, let $\{\mathcal{F}_t(x)\}_{t\in\mathcal{T}}$ (with $\mathcal{F}_t(x) \subseteq \mathcal{Y}$) denote a nested sequence of sets constructed based on the first split of training data $\{(X_i, Y_i) : i \in \mathcal{I}_1\}$, that is, $\mathcal{F}_t(x) \subseteq \mathcal{F}_{t'}(x)$ for $t \leqslant t'$. The sets $\mathcal{F}_t(x)$ are not fixed but random through $\mathcal{I}_1$. We suppress this dependence for notational simplicity. Consider the score

$$r(x, y) \ := \ \inf\{t \in \mathcal{T} : \ y \in \mathcal{F}_t(x)\}, \tag{9.2}$$

where r is a mnemonic for "radius" and $r(x, y)$ can be informally thought of as the smallest "radius" of the set that captures y (and perhaps thinking of a multivariate response, that is

$\mathcal{Y} \subseteq \mathbb{R}^d$, and $\{\mathcal{F}_t(x)\}$ as representing appropriate balls/ellipsoids might help with that intuition). Define the scores for the second split of the training data $\{r_i = r(X_i, Y_i)\}_{i\in\mathcal{I}_2}$ and set

$$Q_{1-\alpha}(r, \mathcal{I}_2) \ := \ \lceil (1-\alpha)(1+1/|\mathcal{I}_2|)\rceil\text{-th quantile of } \{r_i\}_{i\in\mathcal{I}_2}.$$

(that is, $Q_{1-\alpha}(r, \mathcal{I}_2)$ is the $\lceil (1-\alpha)(1+1/|\mathcal{I}_2|)\rceil$-th largest element of the set $\{r_i\}_{i\in\mathcal{I}_2}$). The final prediction set is given by

$$C(x) \ := \ \mathcal{F}_{Q_{1-\alpha}(r,\mathcal{I}_2)}(x) = \{y \in \mathcal{Y} : \ r(x,y) \leqslant Q_{1-\alpha}(r, \mathcal{I}_2)\}. \tag{9.3}$$

The following well known sample coverage guarantee holds true (Papadopoulos et al., 2002; Lei et al., 2018).

Proposition 9.1. *If $\{(X_i, Y_i)\}_{i\in[n]\cup\{n+1\}}$ are exchangeable, then the prediction set $C(\cdot)$ in (9.3) satisfies*

$$\mathbb{P}\left(Y_{n+1} \in C(X_{n+1}) \mid \{(X_i, Y_i) : i \in \mathcal{I}_1\}\right) \ \geqslant \ 1-\alpha.$$

If the scores $\{r_i, i \in \mathcal{I}_2\}$ are almost surely distinct, then $C(\cdot)$ also satisfies

$$\mathbb{P}\left(Y_{n+1} \in C(X_{n+1}) \mid \{(X_i, Y_i) : i \in \mathcal{I}_1\}\right) \ \leqslant \ 1-\alpha + \frac{1}{|\mathcal{I}_2|+1}. \tag{9.4}$$

See 9.E.1 for a proof. Equation (9.2) is the key step that converts a sequence of nested sets $\{\mathcal{F}_t(x)\}_{t\in\mathcal{T}}$ into a nonconformity score r. Through two examples, we demonstrate how natural sequences of nested sets in fact give rise to standard nonconformity scores considered in literature, via equation (9.2).

1. **Split/Inductive Conformal (Papadopoulos et al., 2002; Lei et al., 2018).** Let $\widehat{\mu}(\cdot)$ be an estimator of the regression function $\mathbb{E}[Y|X]$ based on $(X_i, Y_i), i \in \mathcal{I}_1$, and consider nested sets corresponding to symmetric intervals around the mean estimate:

   $$\mathcal{F}_t(x) \ := \ [\widehat{\mu}(x) - t, \widehat{\mu}(x) + t], \ t \in \mathcal{T} = \mathbb{R}^+.$$

   Observe now that

   $$\begin{aligned} \inf\{t \geqslant 0 : y \in \mathcal{F}_t(x)\} &= \inf\{t \geqslant 0 : \widehat{\mu}(x) - t \leqslant y \leqslant \widehat{\mu}(x) + t\} \\ &= \inf\{t \geqslant 0 : \ -t \leqslant y - \widehat{\mu}(x) \leqslant t\} \ = \ |y - \widehat{\mu}(x)|, \end{aligned}$$

   which is exactly the nonconformity score of split conformal.

2. **Conformalized Quantiles (Romano et al., 2019).** For any $\beta \in (0,1)$, let the function $q_\beta(\cdot)$ be the conditional quantile function. Specifically, for each x, define $q_\beta(x) := \sup\{a : \mathbb{P}(Y \leqslant a \mid X = x) \leqslant \beta\}$. Let $\widehat{q}_{\alpha/2}(\cdot), \widehat{q}_{1-\alpha/2}(\cdot)$ be any conditional quantile estimators based on $(X_i, Y_i), i \in \mathcal{I}_1$. If the quantile estimates are good, we hope that $\mathbb{P}(Y \in [\widehat{q}_{\alpha/2}(X), \widehat{q}_{1-\alpha/2}(X)]) \approx 1-\alpha$, but this cannot be guaranteed in a distribution-free or assumption lean manner. However, it may be possible to achieve this with a symmetric expansion or shrinkage of the interval $[\widehat{q}_{\alpha/2}(X), \widehat{q}_{1-\alpha/2}(X)]$ (assuming $\widehat{q}_{\alpha/2}(X) \leqslant \widehat{q}_{1-\alpha/2}(X)$). Following the intuition, consider

   $$\mathcal{F}_t(x) \ := \ [\widehat{q}_{\alpha/2}(x) - t, \widehat{q}_{1-\alpha/2}(x) + t], \ t \in \mathbb{R}. \tag{9.5}$$

Note that the sets in (9.5) are increasing in t if $\widehat{q}_{\alpha/2}(x) \leqslant \widehat{q}_{1-\alpha/2}(x)$, and

$$\begin{aligned}\inf\{t \in \mathbb{R} : \ y \in \mathcal{F}_t(x)\} &= \inf\{t \in \mathbb{R} : \ \widehat{q}_{\alpha/2}(x) - t \leqslant y \leqslant \widehat{q}_{1-\alpha/2}(x) + t\} \\ &= \max\{\widehat{q}_{\alpha/2}(x) - y, y - \widehat{q}_{1-\alpha/2}(x)\}.\end{aligned}$$

Hence $r(X_i, Y_i) = \max\{\widehat{q}_{\alpha/2}(X_i) - Y_i, \ Y_i - \widehat{q}_{1-\alpha/2}(X_i)\}$ for $i \in \mathcal{I}_2$. This recovers exactly the nonconformity score proposed by Romano et al. (2019).

We believe that it is more intuitive to start with the shape of the predictive set, like we did above, than a nonconformity score. In this sense, nested conformal is a formalized technique to go from statistical/geometric intuition about the shape of the prediction set to a nonconformity score. See Table 9.1 for more translations between scores and nested sets.

Table 9.1: Examples from the literature covered by nested conformal framework. The methods listed are split conformal, locally weighted conformal, CQR, CQR-m, CQR-r, distributional conformal and conditional level-set conformal. Functions $\widehat{q}_a$ represents a conditional quantile estimate at level a, and $\widehat{f}$ represents a conditional density estimate.

Reference	$\mathcal{F}_t(x)$	$\mathcal{T}$	Estimates
Lei et al. (2018)	$[\widehat{\mu}(x) - t, \widehat{\mu}(x) + t]$	$[0, \infty)$	$\widehat{\mu}$
Lei et al. (2018)	$[\widehat{\mu}(x) - t\widehat{\sigma}(x), \widehat{\mu}(x) + t\widehat{\sigma}(x)]$	$[0, \infty)$	$\widehat{\mu}, \widehat{\sigma}$
Romano et al. (2019)	$[\widehat{q}_{\alpha/2}(x) - t, \widehat{q}_{1-\alpha/2}(x) + t]$	$(-\infty, \infty)$	$\widehat{q}_{\alpha/2}, \widehat{q}_{1-\alpha/2}$
Kivaranovic et al. (2020)	$(1 + t)[\widehat{q}_{\alpha/2}(x), \widehat{q}_{1-\alpha/2}(x)] - t\widehat{q}_{1/2}(x)$	$(-\infty, \infty)$	$\widehat{q}_{\alpha/2}, \widehat{q}_{1-\alpha/2}, \widehat{q}_{1/2}$
Sesia and Candès (2020)	$[\widehat{q}_{\alpha/2}(x), \widehat{q}_{1-\alpha/2}(x)] \pm t(\widehat{q}_{1-\alpha/2}(x) - \widehat{q}_{\alpha/2}(x))$	$(-1/2, \infty)$	$\widehat{q}_{\alpha/2}, \widehat{q}_{1-\alpha/2}$
Chernozhukov et al. (2021)	$[\widehat{q}_t(x), \widehat{q}_{1-t}(x)]$	$(0, 1/2)$	$\{\widehat{q}_\alpha\}_{\alpha \in [0,1]}$
Izbicki et al. (2020)	$\{y : \widehat{f}(y\|x) \geqslant \check{t}_\delta(x)\}$[2]	$[0, 1]$	$\widehat{f}$

Split conformal prediction methods are often thought of as being statistically inefficient because they only make use of one split of the data for training the base algorithm, while the rest is held-out for calibration. Recently many extensions have been proposed (Carlsson et al., 2014; Vovk, 2015; Johansson et al., 2014; Boström et al., 2017; Barber et al., 2021; Kim et al., 2020) that make use all of the data for training. All of these methods can be rephrased easily in terms of nested sets; we do so in Sections 9.3 and 9.4. This understanding also allows us to develop our novel method QOOB in Section 9.5.

9.3 Cross-conformal and Jackknife+ using nested sets

In the previous section, we used a part of training data to construct the nested sets and the remaining part to calibrate them for finite sample validity. This procedure, although computationally efficient, can be statistically inefficient due to the reduction of the sample size used for calibrating. Instead of splitting into two parts, it is statistically more efficient to split the data into multiple parts. In this section, we describe such versions of nested conformal prediction

[2] $\check{t}_\delta(x)$ is an estimator of $t_\delta(x)$, where $t_\delta(x)$ is defined the largest t such that $\mathbb{P}(f(Y|X) \geqslant t_\delta(X)\big|X = x) \geqslant 1 - \delta$; see (Izbicki et al., 2020, Definition 3.3) for details.

sets and prove their validity. These versions in the score-based conformal framework are called cross-conformal prediction and the jackknife+, and were developed by Vovk (2015) and Barber et al. (2021), but the latter only for a specific score function.

9.3.1 Rephrasing leave-one-out cross-conformal using nested sets

We now derive leave-one-out cross-conformal in the language of nested prediction sets. Suppose $\{\mathcal{F}_t^{-i}(x)\}_{t\in\mathcal{T}}$ for each $x \in \mathcal{X}$, $i \in [n]$ denotes a collection of nested sets constructed based only on $\{(X_j, Y_j)\}_{j\in[n]\setminus\{i\}}$. We assume that $\{\mathcal{F}_t^{-i}(x)\}_{t\in\mathcal{T}}$ is constructed invariantly to permutations of the input points $\{(X_j, Y_j)\}_{j\in[n]\setminus\{i\}}$; note that this is also required in the original description of cross-conformal (Vovk, 2015). The i-th nonconformity score r_i induced by these nested sets is defined as $r_i(x, y) = \inf\{t \in \mathcal{T} : y \in \mathcal{F}_t^{-i}(x)\}$. The leave-one-out cross-conformal prediction set is given by

$$C^{\text{LOO}}(x) := \left\{ y \in \mathbb{R} : \sum_{i=1}^{n} \mathbb{1}\{r_i(X_i, Y_i) < r_i(x, y)\} < (1-\alpha)(n+1) \right\}. \tag{9.6}$$

For instance, given a conditional mean estimator $\widehat{\mu}^{-i}(\cdot)$ trained on $\{(X_j, Y_j)\}_{j\in[n]\setminus\{i\}}$, we can consider the nested sets $\mathcal{F}_t^{-i}(x) = [\widehat{\mu}^{-i}(x) - t, \widehat{\mu}^{-i}(x) + t]$ to realize the absolute deviation residual function $r_i(x, y) = |y - \widehat{\mu}^{-i}(x)|$. We now state the coverage guarantee that $C^{\text{LOO}}(\cdot)$ satisfies.

Theorem 9.1. *If $\{(X_i, Y_i)\}_{i\in[n+1]}$ are exchangeable and the sets $\mathcal{F}_t^{-i}(x)$ constructed based on $\{(X_j, Y_j)\}_{j\in[n]\setminus\{i\}}$ are invariant to their ordering, then*

$$\mathbb{P}(Y_{n+1} \in C^{LOO}(X_{n+1})) \geqslant 1 - 2\alpha.$$

See 9.E.2 for a proof of Theorem 9.1, which follows the proof of Theorem 1 in Barber et al. (2021) except with the new residual defined based on nested sets. In particular, Theorem 9.1 applies when the nested sets are constructed using conditional quantile estimators as in the conformalized quantile example discussed in Section 9.2. The discussion in this section can be generalization to cross-conformal and the CV+ methods of Vovk (2015) and Barber et al. (2021), which construct K-fold splits of the data and require training an algorithm only n/K times (instead of n times in the leave-one-out case). These are discussed in the nested framework in 9.B.

In a regression setting, one may be interested in constructing prediction sets that are intervals (since they are easily interpretable), whereas $C^{\text{LOO}}(x)$ need not be an interval in general. Also, it is not immediately evident how one would algorithmically compute the prediction set defined in (9.6) without trying out all possible values $y \in \mathcal{Y}$. We discuss these concerns in Sections 9.3.2 and 9.3.3.

9.3.2 Jackknife+ and other prediction intervals that contain $C^{\mathrm{LOO}}(x)$

In a regression setting, prediction intervals may be more interpretable or 'actionable' than prediction sets that are not intervals. To this end, intervals that contain $C^{\mathrm{LOO}}(x)$ are good candidates for prediction intervals since they inherit the coverage validity of Theorem 9.1. For the residual function $r_i(x,y) = |y - \widehat{\mu}^{-i}(x)|$, Barber et al. (2021) provided an interval that always contains $C^{\mathrm{LOO}}(x)$, called the jackknife+ prediction interval. In this section, we discuss when the jackknife+ interval can be defined for general nested sets. Whenever jackknife+ can be defined, we argue that another interval can be defined that contains $C^{\mathrm{LOO}}(x)$ and is guaranteed to be no longer in width than the jackknife+ interval.

For general nonconformity scores, an analog of the jackknife+ interval may not exist. However, in the special case when the nested sets $\mathcal{F}_t^{-i}(x)$ are themselves either intervals or empty sets, an analog of the jackknife+ interval can be derived. Note that all the examples listed in Table 9.1 (except for the last one) result in $\mathcal{F}_t(x)$ being either a nonempty interval or the empty set. For clarity of exposition, we discuss the empty case separately in 9.D. Below, suppose $\mathcal{F}^{-i}_{r_i(X_i,Y_i)}(x)$ is a nonempty interval and define the notation

$$[\ell_i(x), u_i(x)] \;:=\; \mathcal{F}^{-i}_{r_i(X_i,Y_i)}(x).$$

With this notation, the cross-conformal prediction set can be re-written as

$$\begin{aligned} C^{\mathrm{LOO}}(x) &= \left\{ y : \sum_{i=1}^{n} \mathbb{1}\{y \notin [\ell_i(x), u_i(x)]\} < (1-\alpha)(n+1) \right\} \\ &= \left\{ y : \alpha(n+1) - 1 < \sum_{i=1}^{n} \mathbb{1}\{y \in [\ell_i(x), u_i(x)]\} \right\}. \end{aligned} \tag{9.7}$$

Suppose $y < q^-_{n,\alpha}(\{\ell_i(x)\})$, where $q^-_{n,\alpha}(\{\ell_i(x)\})$ denotes the $\lfloor \alpha(n+1) \rfloor$-th smallest value of $\{\ell_i(x)\}_{i=1}^n$. Clearly,

$$\sum_{i=1}^{n} \mathbb{1}\{y \in [\ell_i(x), u_i(x)]\} \leqslant \sum_{i=1}^{n} \mathbb{1}\{y \geqslant l_i(x)\} \leqslant \lfloor \alpha(n+1) \rfloor - 1,$$

and hence $y \notin C^{\mathrm{LOO}}(x)$. Similarly it can be shown that if $y > q^+_{n,\alpha}(\{u_i(x)\})$ (where $q^+_{n,\alpha}(\{u_i(x)\})$ denotes the $\lceil (1-\alpha)(n+1) \rceil$-th smallest value of $\{u_i(x)\}_{i=1}^n$), $y \notin C^{\mathrm{LOO}}(x)$. Hence, defining the jackknife+ prediction interval as

$$C^{\mathrm{JP}}(x) \;:=\; [q^-_{n,\alpha}(\{\ell_i(x)\}),\, q^+_{n,\alpha}(\{u_i(x)\})], \tag{9.8}$$

we conclude

$$C^{\mathrm{LOO}}(x) \;\subseteq\; C^{\mathrm{JP}}(x) \quad \text{for all} \quad x \in \mathcal{X}. \tag{9.9}$$

However, there exists an even shorter interval containing $C^{\mathrm{LOO}}(x)$: its convex hull; this does not require the nested sets to be intervals. The convex hull of $C^{\mathrm{LOO}}(x)$ is defined as the smallest interval containing itself. Hence,

$$C^{\mathrm{LOO}}(x) \;\subseteq\; \mathrm{Conv}(C^{\mathrm{LOO}}(x)) \;\subseteq\; C^{\mathrm{JP}}(x). \tag{9.10}$$

Because of (9.10), the coverage guarantee from Theorem 9.1 continues to hold for $\mathrm{Conv}(C^{\mathrm{LOO}}(x))$ and $C^{\mathrm{JP}}(x)$. Interestingly, $C^{\mathrm{LOO}}(x)$ can be empty but $C^{\mathrm{JP}}(x)$ is non-empty if each $\mathcal{F}^{-i}_{r_i(X_i,Y_i)}(x)$ is non-empty (in particular it contains the medians of $\{\ell_i(x)\}$ and $\{u_i(x)\}$). Further, $\mathrm{Conv}(C^{\mathrm{LOO}}(x))$ can be a strictly smaller interval than $C^{\mathrm{JP}}(x)$; see Section 9.6.4 for details.

9.3.3 Efficient computation of the cross-conformal prediction set

Equation (9.6) defines $C^{\mathrm{LOO}}(x)$ implicitly, and does not address the question of how to compute the mathematically defined prediction set efficiently. If the nested sets $\mathcal{F}_t^{-i}(x)$ are themselves guaranteed to either be intervals or empty sets, jackknife+ seems like a computationally feasible alternative since it just relies on the quantiles $q^-_{n,\alpha}(\{\ell_i(x)\}),\ q^+_{n,\alpha}(\{u_i(x)\})$ which can be computed efficiently. However, it turns out that $C^{\mathrm{LOO}}(x)$, $\mathrm{Conv}(C^{\mathrm{LOO}}(x))$, and $C^{\mathrm{JP}}(x)$ can all be computed in near linear in n time. In this section, we provide an algorithm for near linear time computation of the aforementioned prediction sets. We will assume for simplicity that $\mathcal{F}_t^{-i}(x)$ is always an interval; the empty case is discussed separately in 9.D.

First, notice that the inclusion in (9.6) need not be ascertained for every $y \in \mathcal{Y}$ but only for a finite set of values in $\mathcal{Y}$. These values are exactly the ones corresponding to the end-points of the intervals produced by each training point $\mathcal{F}^{-i}_{r_i(X_i,Y_i)}(x) = [\ell_i(x), u_i(x)]$. This is because none of the indicators $\mathbb{1}\{r_i(X_i,Y_i) < r_i(x,y)\}$ change value between two consecutive interval end-points. Since $\ell_i(x)$ and $u_i(x)$ can be repeated, we define the bag of all these values (see footnote[3] for $\wr \cdot \wr$ notation):

$$\mathcal{Y}^x := \bigcup_{i=1}^{n} \wr \ell_i(x), u_i(x) \wr. \tag{9.11}$$

Thus we only need to verify the condition

$$\sum_{i=1}^{n} \mathbb{1}\{y \in [\ell_i(x), u_i(x)]\} > \alpha(n+1) - 1 \tag{9.12}$$

for the $2n$ values of $y \in \mathcal{Y}^x$ and construct the prediction sets suitably. Done naively, verifying (9.12) itself is an $O(n)$ operation for an overall time of $O(n^2)$. However, (9.12) can be verified for all $y \in \mathcal{Y}^x$ in one pass on the sorted $\mathcal{Y}^x$ for a running time of $O(n \log n)$; we describe how to do so.

Let the sorted order of the points $\mathcal{Y}^x$ be $y_1^x \leqslant y_2^x \leqslant \ldots \leqslant y^x_{|\mathcal{Y}^x|}$. If $\mathcal{Y}^x$ contains repeated elements, we require that the left end-points ℓ_i come earlier in the sorted order than the right end-points u_i for the repeated elements. Also define the bag of indicators $\mathcal{S}^x$ with elements $s_1^x \leqslant s_2^x \leqslant \ldots \leqslant s^x_{|\mathcal{Y}^x|}$, where

$$s_i^x := \begin{cases} 1 & \text{if } y_i^x \text{ corresponds to a left end-point} \\ 0 & \text{if } y_i^x \text{ corresponds to a right end-point.} \end{cases} \tag{9.13}$$

[3] A bag denoted by $\wr \cdot \wr$ is an unordered set with potentially repeated elements. Bag unions respect the number of occurrences of the elements, eg $\wr 1,1,2 \wr \cup \wr 1,3 \wr = \wr 1,1,1,2,3 \wr$.

Algorithm 9.1 Efficient cross-conformal style aggregation

```
Require: Training data $\{(X_i, Y_i)\}_{i=1}^n$, desired coverage level $\alpha$; $\mathcal{Y}^x$ and $\mathcal{S}^x$ computed as defined
    in equations (9.11), (9.13) using the training data $\{(X_i, Y_i)\}_{i=1}^n$, test point $x$, and any sequence
    of nested sets $\mathcal{F}_t^{-i}(\cdot)$.
Ensure: Prediction set $C^x \subseteq \mathcal{Y}$
threshold $\leftarrow \alpha(n+1) - 1$; if threshold $< 0$, then return $\mathbb{R}$ and stop
$C^x \leftarrow \varnothing$, count $\leftarrow 0$, left_endpoint $\leftarrow 0$
for $i \leftarrow 1$ to $|\mathcal{Y}^x|$ do
    if $s_i^x = 1$ then
        count $\leftarrow$ count $+ 1$
        if count $>$ threshold and count $- 1 \leqslant$ threshold then
            left_endpoint $\leftarrow y_i^x$
        end if
    else
        if count $>$ threshold and count $- 1 \leqslant$ threshold then
            $C^x \leftarrow C^x \cup \{[$left_endpoint$, y_i^x]\}$
        end if
        count $\leftarrow$ count $- 1$
    end if
end for
return $C^x$
```

Given $\mathcal{Y}^x$ and $\mathcal{S}^x$, Algorithm 9.1 describes how to compute the cross-conformal prediction set in one pass (thus time $O(n)$) for every test point. Thus the runtime (including the sorting) is $O(n \log n)$ time to compute the predictions $\mathcal{F}_{r_i(X_i,Y_i)}^{-i}(x)$ for every i. If each prediction takes time $\leqslant T$, the overall time is $O(n \log n) + Tn$, which is the same as jackknife+.[4]

Proposition 9.2. *Algorithm 9.1 correctly computes the cross-conformal prediction set 9.6 given $\mathcal{Y}^x$ and $\mathcal{S}^x$.*

The proof of the proposition is in 9.E.4. The proof proceeds through a step-wise description of the algorithm that makes it transparent how the algorithm verifies (9.12) for every value of $y \in \mathcal{Y}^x$.

9.4 Extending ensemble based out-of-bag conformal methods using nested sets

Cross-conformal, jackknife+, and their K-fold versions perform multiple splits of the data and for every training point (X_i, Y_i), a residual function r_i is defined using a set of training points

[4] For jackknife+, using quick-select, we could obtain $O(n) + Tn$ randomized, but the testing time Tn usually dominates the additional $n \log n$ required to sort.

that does not include (X_i, Y_i). In the previous section, our description required training the base algorithm multiple times on different splits of the data. Often each of these individual algorithms is itself an ensemble algorithm (such as random forests). As described in this section, an ensemble algorithm naturally provide multiple (random) splits of the data from a single run and need not be re-trained on different splits to produce conformal prediction sets. This makes the conformal procedure computationally efficient. At the same time, like cross-conformal, the conformal prediction sets produced here are often shorter than split conformal because they use all of the training data for prediction. In a series of interesting papers (Johansson et al., 2014; Boström et al., 2017; Linusson et al., 2019; Kim et al., 2020), many authors have exhibited promising empirical evidence that these ensemble algorithms improve the width of prediction sets without paying a computational cost. We call this the *OOB-conformal* method (short for out-of-bag). Linusson et al. (2019) provided an extensive empirical comparison of OOB-conformal to other conformal methods but without formal validity guarantees.

We now describe the procedure formally within the nested conformal framework, thus extending it instantly to residual functions that have hitherto not been considered. Our procedure can be seen as a generalization of the OOB-conformal method (Linusson et al., 2019) or the jackknife+ after bootstrap method (Kim et al., 2020):

1. Let $\{M_j\}_{j=1}^K$ denote $K \geqslant 1$ independent and identically distributed random sets drawn uniformly from $\{M : M \subset [n], |M| = m\}$. This is the same as subsampling. Alternatively $\{M_j\}_{j=1}^K$ could be i.i.d. random bags, where each bag is obtained by drawing m samples with replacement from $[n]$. This procedure corresponds to bootstrap.
2. For every $i \in [n]$, define
$$M_{-i} := \{j : i \notin M_j\},$$
which contains the indices of the training sets that are *out-of-bag* for the i-th data point.
3. The idea now is to use an ensemble learning method that, for every i, aggregates $|M_{-i}|$ many predictions to identify a single collection of nested sets $\{\mathcal{F}_t^{-i}(x)\}_{t\in\mathcal{T}}$. For instance, one can obtain an estimate $\widehat{\mu}_j(\cdot)$ of the conditional mean based on the training data corresponding to M_j, for every j, and then construct
$$\mathcal{F}_t^{-i}(x) = [\widehat{\mu}_{-i}(x) - t, \widehat{\mu}_{-i}(x) + t],$$
where $\widehat{\mu}_{-i}(\cdot)$ is some combination (such as the mean) of $\{\widehat{\mu}_j(\cdot)\}_{\{j:i\notin M_j\}}$.
4. The remaining conformalization procedure is identical to $C^{\text{LOO}}(x)$ described in Section 9.3. Define the residual score $r_i(x,y) := \inf\left\{t \in \mathcal{T} : \, y \in \mathcal{F}_t^{-i}(x)\right\}$.

Using the cross-conformal scheme, the prediction set for any $x \in \mathcal{X}$ is given as

$$C^{\text{OOB}}(x) := \left\{y : \sum_{i=1}^{n} \mathbb{1}\{r_i(X_i, Y_i) < r_i(x,y)\} < (1-\alpha)(n+1)\right\}. \tag{9.14}$$

If $\mathcal{F}_t^{-i}(x)$ is an interval for all $1 \leqslant i \leqslant n$ and $x \in \mathcal{X}$, then following the discussion in Section 9.3.2, we could also derive a jackknife+ style prediction interval that is guaranteed to be non-empty:

$$C^{\text{OOB-JP}}(x) := [q_{n,\alpha}^{-}(\ell_i(x)), q_{n,\alpha}^{+}(u_i(x))]. \tag{9.15}$$

If $\mathcal{F}_t^{-i}(x)$ could further contain empty sets, a jackknife+ interval can still be derived following the discussion in 9.D, but we skip these details here. Once again, we have that for every $x \in \mathcal{X}$, $C^{\text{OOB}}(x) \subseteq C^{\text{OOB-JP}}(x)$; see Equation (9.9) for details. The computational discussion of Section 9.3.3 extends to C^{OOB}.

Recently, Kim et al. (2020) provided a $1 - 2\alpha$ coverage guarantee of the OOB-conformal method when $\mathcal{F}_t^{-i}(x) = [\widehat{\mu}_{-i}(x) - t, \widehat{\mu}_{-i}(x) + t]$ where $\widehat{\mu}_{-i}(\cdot)$ represents the aggregation of conditional mean estimate from $\{M_j\}_{i \notin M_j}$. We generalize their result to any sequence of nested sets and extend it to the cross-conformal style aggregation scheme. In order to obtain a coverage guarantee, the conformal method must ensure a certain exchangeability requirement is satisfied. To do so, the argument of Kim et al. (2020) required the number of random resamples K to itself be drawn randomly from a binomial distribution. We assert the same requirement in the following theorem (proved in 9.E.3).

Theorem 9.2. *Fix a permutation invariant ensemble technique that constructs sets $\{\mathcal{F}_t^{-i}\}_{t \in \mathcal{T}}$ given a collection of subsets of $[n]$. Fix integers $\widetilde{K}, m \geqslant 1$ and let*

$$K \sim \text{Binomial}\left(\widetilde{K}, \left(1 - \frac{1}{n+1}\right)^m\right) \quad (\textit{in the case of bagging}), \text{ or,}$$
$$K \sim \text{Binomial}\left(\widetilde{K}, 1 - \frac{m}{n+1}\right) \quad (\textit{in the case of subsampling}).$$

Then $\mathbb{P}(Y_{n+1} \in C^{OOB}(X_{n+1})) \geqslant 1 - 2\alpha$.

Because $C^{\text{OOB}}(x) \subseteq C^{\text{OOB-JP}}(x)$ for every $x \in \mathcal{X}$, the validity guarantee continues to hold for $C^{\text{OOB-JP}}(\cdot)$. While we can only prove a $1 - 2\alpha$ coverage guarantee, it has been observed empirically that the OOB-conformal method with regression forests as the ensemble scheme and nested sets $\{[\widehat{\mu}(x) - t\widehat{\sigma}(x), \widehat{\mu}(x) + t\widehat{\sigma}(x)]\}_{t \in \mathbb{R}^+}$ satisfies $1 - \alpha$ coverage while providing the shortest prediction sets on average (Boström et al., 2017). On the other hand, the best empirically performing nested sets are the ones introduced by Romano et al. (2019): $\{[\widehat{q}_\beta(x) - t, \widehat{q}_\beta(x) + t]\}_{t \in \mathbb{R}}$ (for an appropriately chosen β). Using nested conformal, we show how these these two ideas can be seamlessly integrated: quantile based nested sets with an OOB-style aggregation scheme. In Section 9.5 we formally develop our novel method QOOB, and in Section 9.6 we empirically verify that it achieves competitive results in terms of the length of prediction sets.

9.5 QOOB: A novel conformal method using nested sets

The nested conformal interpretation naturally separates the design of conformal methods into two complementary aspects:

(a) identifying an information efficient nonconformity score based on a set of nested intervals, and

(b) performing sample efficient aggregation of the nonconformity scores while maintaining validity guarantees.

In this section, we leverage this dichotomy to merge two threads of ideas in the conformal literature and develop a novel conformal method that empirically achieves state-of-the-art results in terms of the width of prediction sets.

First, we review what is known on aspect (b). While split-conformal based methods are computationally efficient, they lose sample efficiency due to sample splitting. Aggregated conformal methods such as cross-conformal, jackknife+, and OOB-conformal do not have this drawback and are the methods of choice for computationally feasible and sample efficient prediction sets. Among all aggregation techniques, the OOB-conformal method has been observed empirically to be the best aggregation scheme which uses all the training data efficiently (Boström et al., 2017).

Next, we consider aspect (a), the design of the nested sets. The nested sets considered by Boström et al. (2017) are $\{[\widehat{\mu}(x)-t\widehat{\sigma}(x), \widehat{\mu}(x)+t\widehat{\sigma}(x)]\}_{t\in\mathbb{R}^+}$ based on mean and variance estimates obtained using out-of-bag trees. On the other hand, it has been demonstrated that nested sets based on quantile estimates $\widehat{q}_s(x)$ given by $\{[\widehat{q}_\beta(x) - t, \widehat{q}_{1-\beta}(x) + t]\}_{t\in\mathbb{R}}$ perform better than those based on mean-variance estimates in the split conformal setting (Romano et al., 2019; Sesia and Candès, 2020).

Building on these insights, we make the following suggestion: Quantile Out-of-Bag (QOOB) conformal; pronounced "cube" conformal. This method works in the following way. First, a quantile regression based on random forest (Meinshausen, 2006) with T trees is learnt by subsampling or bagging the training data T times. Next, the out-of-bag trees for every training point (X_i, Y_i) are used to learn a quantile estimator function $\widehat{q}_s^{-i}(\cdot)$ for $s = \beta$ and $s = 1 - \beta$. Here $\beta = k\alpha$ for some constant k. Now for every i and some $x \in \mathcal{X}$, we define the nested sets as

$$\mathcal{F}_t^{-i}(x) \;:=\; [\widehat{q}_\beta^{-i}(x) - t, \widehat{q}_{1-\beta}^{-i}(x) + t].$$

The nonconformity scores based on these nested sets are aggregated to provide a prediction set as described by $C^{\mathrm{OOB}}(x)$ in (9.14) of Section 9.4. Algorithm 9.2 describes QOOB procedurally. Following Section 9.3.3, the aggregation step of QOOB (line 13, Algorithm 9.2) can be performed in time $O(n \log n)$.

Since QOOB is a special case of OOB-conformal, it inherits an assumption-free $1 - 2\alpha$ coverage guarantee from Theorem 9.2 if K is drawn from an appropriate binomial distribution as described in the theorem. In practice, we typically obtain $1 - \alpha$ coverage with a fixed K. In Section 9.6, we empirically demonstrate that QOOB achieves state-of-the-art performance on multiple real-world datasets. We also discuss three aspects of our method:

(a) how to select the nominal quantile level $\beta = k\alpha$,

(b) the effect of the number of trees T on the performance, and

(c) the performance of the jackknife+ version of our method (QOOB-JP), which corresponds to the OOB-JP style aggregation (equation (9.15)) of quantile-based nonconformity scores.

Algorithm 9.2 Quantile Out-of-Bag conformal (QOOB)

```
Require: Training data $\{(X_i, Y_i)\}_{i=1}^n$, test point $x$, desired coverage level $\alpha$, number of trees $T$,
    nominal quantile level $\beta$ (default $= 2\alpha$)
Ensure: Prediction set $C^x \subseteq \mathcal{Y}$
$\{M_j\}_{j=1}^T \leftarrow$ training bags drawn independently from $[n]$ using subsampling or bootstrap
$\{M_{-i}\}_{i=1}^n \leftarrow \{j : i \notin M_j\}$
for $j \leftarrow 1$ to $T$ do
    $\phi_j \leftarrow$ Quantile regression trees learnt using the data-points in $M_j$
                    (this step could include subsampling of features)
end for
for $i \leftarrow 1$ to $n$ do
    $\Phi_{-i} \leftarrow \{\phi_j : j \in M_{-i}\}$
    $\widehat{q}^{-i}(\cdot) \leftarrow$ quantile regression forest using the trees $\Phi_{-i}$
    $\mathcal{F}_t^{-i}(\cdot) \leftarrow [\widehat{q}_\beta^{-i}(\cdot) - t, \widehat{q}_{1-\beta}^{-i}(\cdot) + t]$
    $r_i(\cdot,\cdot) \leftarrow ((x,y) \to \inf\{t \in \mathcal{T} : y \in \mathcal{F}_t^{-i}(x)\})$
end for
$C^x \leftarrow$ OOB prediction set defined in Equation (9.14); (call Algorithm 9.1 with $\mathcal{Y}^x$, $\mathcal{S}^x$
    computed using $\mathcal{F}_t^{-i}(\cdot)$, the training data $\{(X_i, Y_i)\}_{i=1}^n$ and test point $x$ as described in
    equations (9.11), (9.13))
return $C^x$
```

9.6 Numerical comparisons

We compare several methods discussed in this chapter using synthetic and real datasets. MATLAB code to execute QOOB and reproduce the experiments in this section is provided at https://github.com/AIgen/QOOB. Some experiments on synthetic data are discussed in Section 9.6.5; the rest of this section discusses results on real datasets. We use the following six datasets from the UCI repository: blog feedback, concrete strength, superconductivity, news popularity, kernel performance and protein structure. Metadata and links for these datasets are provided in 9.G. In order to assess the coverage and width properties, we construct multiple version of each of these three datasets. For each dataset, we obtain 100 versions by independently drawing 1000 data points randomly (without replacement) from the full dataset. Then we split each such version into two parts: training and testing of sizes 768[5] and 232 respectively. Hence corresponding to each of the six datasets, we get 100 different datasets with 768 training and 232 testing points.

For each conformal method, we report the following two metrics:

- *Mean-width*: For a prediction set $C(x) \subseteq \mathcal{Y} = \mathbb{R}$ its width is defined as its Lebesgue measure. For instance, if $C(x)$ is an interval, then the width is its length and if $C(x)$ is a union of two or more disjoint intervals, then the width is the sum of the lengths of these

[5] the number of training points is divisible by many factors, which is useful for creating a varying number of folds for K-fold methods

disjoint intervals. We report the average over the mean-width given by

$$\text{Ave-Mean-Width} := \frac{1}{100}\sum_{b=1}^{100}\left(\frac{1}{232}\sum_{i=1}^{232}\text{width}(C_b(X_i^b))\right). \tag{9.16}$$

Here $C_b(\cdot)$ is a prediction set learnt from the b-th version of a dataset. The outer mean is the average over 100 versions of a dataset. The inner mean is the average of the width over the testing points in a particular version of a dataset.

- *Mean-coverage*: We have proved finite-sample coverage guarantees for all our methods and to verify (empirically) this property, we also report the average over the mean-coverage given by

$$\text{Ave-Mean-Coverage} := \frac{1}{100}\sum_{b=1}^{100}\left(\frac{1}{232}\sum_{i=1}^{232}\mathbb{1}\{Y_i^b \in C_b(X_i^b)\}\right). \tag{9.17}$$

In addition to reporting the average over versions of a dataset, we also report the estimated standard deviation of the average (to guage the fluctuations). In the rest of the discussion, the qualification 'average' may be skipped for succinctness, but all reports and conclusions are to be understood as comments on the average value for mean-width and mean-coverage.

Random forest (RF) based regressors perform well across different conformal methods and will be used as the base regressor in our experiments, with varying T, the number of trees. Each tree is trained on an independently drawn bootstrap sample from the training set (containing about $(1 - 1/e)100\% \approx 63.2\%$ of all training points). The numerical comparisons will use the following methods:

1. SC-T: Split conformal (Papadopoulos et al., 2002; Lei et al., 2018) with nested sets $\{[\widehat{\mu}(x) - t, \widehat{\mu}(x) + t]\}_{t\in\mathbb{R}^+}$ and T trees.
2. Split-CQR-T (2α): Split conformalized quantile regression (Romano et al., 2019) with T trees and nominal quantile level 2α. This corresponds to the nested sets $\{[\widehat{q}_{2\alpha}^{-i}(x) - t, \widehat{q}_{1-2\alpha}^{-i}(x) + t]\}_{t\in\mathbb{R}}$. Quantile conformal methods require the nominal quantile level to be set carefully, as also noted by Sesia and Candès (2020). In our experiments, we observe that Split-CQR-T performs well at the nominal quantile level 2α. This is discussed more in Section 9.6.1.
3. 8-fold-CC-T: 8-fold cross-conformal (Vovk, 2015; Barber et al., 2021) with T trees learnt for every fold and the nested sets $\{[\widehat{\mu}(x) - t, \widehat{\mu}(x) + t]\}_{t\in\mathbb{R}^+}$. Leave-one-out cross-conformal is computationally expensive if T trees are to be trained for each fold, and does not lead to significantly improved performance compared to OOB-CC in our experiments. Hence we did not report a detailed comparison across all datasets.
4. OOB-CC-T: OOB-cross-conformal (Johansson et al., 2014; Kim et al., 2020) with T trees. This method considers the nested sets $\{[\widehat{\mu}(x) - t, \widehat{\mu}(x) + t]\}_{t\in\mathbb{R}^+}$ where $\widehat{\mu}$ is the average of the mean-predictions for x on out-of-bag trees.
5. OOB-NCC-T: OOB-normalized-cross-conformal (Boström et al., 2017) with T trees. This method considers nested sets $\{[\widehat{\mu}(x) - t\widehat{\sigma}(x), \widehat{\mu}(x) + t\widehat{\sigma}(x)]\}_{t\in\mathbb{R}^+}$ where $\widehat{\sigma}(x)$ is the standard deviation of mean-predictions for x on out-of-bag trees.

6. QOOB-T (2α): OOB-quantile-cross-conformal with T trees and nominal quantile level $\beta = 2\alpha$. This is our proposed method. In our experiments, we observe that QOOB-T performs well at the nominal quantile level 2α. We elaborate more on the nominal quantile selection in Section 9.6.1.

Table 9.2: Mean-width (9.16) of conformal methods with regression forests ($\alpha = 0.1$). Average values across 100 simulations are reported with the standard deviation in brackets.

Method	Blog	Protein	Concrete	News	Kernel	Superconductivity
SC-100	25.54 (0.71)	16.88 (0.08)	22.29 (0.14)	12491.84 (348.07)	452.71 (5.10)	54.46 (0.37)
Split-CQR-100 (2α)	**12.22** (0.35)	14.20 (0.09)	21.45 (0.12)	**7468.15** (136.93)	**295.49** (3.09)	39.59 (0.27)
8-fold-CC-100	24.83 (0.44)	16.42 (0.05)	19.23 (0.04)	12461.40 (263.54)	411.81 (3.4299)	50.30 (0.24)
OOB-CC-100	24.76 (0.50)	16.38 (0.04)	18.69 (0.03)	12357.58 (213.72)	402.97 (3.13)	49.31 (0.24)
OOB-NCC-100	20.31 (0.42)	14.87 (0.05)	18.66 (0.06)	11500.22 (320.91)	353.35 (2.95)	39.55 (0.22)
QOOB-100 (2α)	14.43 (0.38)	**13.74** (0.05)	**18.19** (0.05)	7941.19 (89.21)	300.04 (2.70)	**37.04** (0.18)

Table 9.3: Mean-coverage (9.17) of conformal methods with regression forests ($\alpha = 0.1$). The standard deviation of these average mean-widths are zero upto two significant digits.

Method	Blog	Protein	Concrete	News	Kernel	Superconductivity
SC-100	0.90	0.90	0.90	0.90	0.90	0.90
Split-CQR-100 (2α)	0.91	0.90	0.90	0.90	0.90	0.90
8-fold-CC-100	0.91	0.91	0.91	0.91	0.91	0.90
OOB-CC-100	0.90	0.91	0.91	0.91	0.91	0.90
OOB-NCC-100	0.92	0.91	0.91	0.92	0.93	0.91
QOOB-100 (2α)	0.92	0.91	0.92	0.91	0.93	0.91

Tables 9.2 and 9.3 report the mean-width and mean-coverage that these conformal methods achieve on 6 datasets. Here, the number of trees T is set to 100 for all the methods. We draw the following conclusions:

- Our novel method QOOB achieves the shortest or close to the shortest mean-width compared to other methods while satisfying the $1 - \alpha$ coverage guarantee. The closest competitor is Split-CQR. As we further investigate in Section 9.6.2, even on datasets where Split-CQR performs better than QOOB, if the number of trees are increased beyond 100, QOOB shows a decrease in mean-width while Split-CQR does not improve. For example, on the kernel dataset, QOOB outperforms Split-CQR at 400 trees.

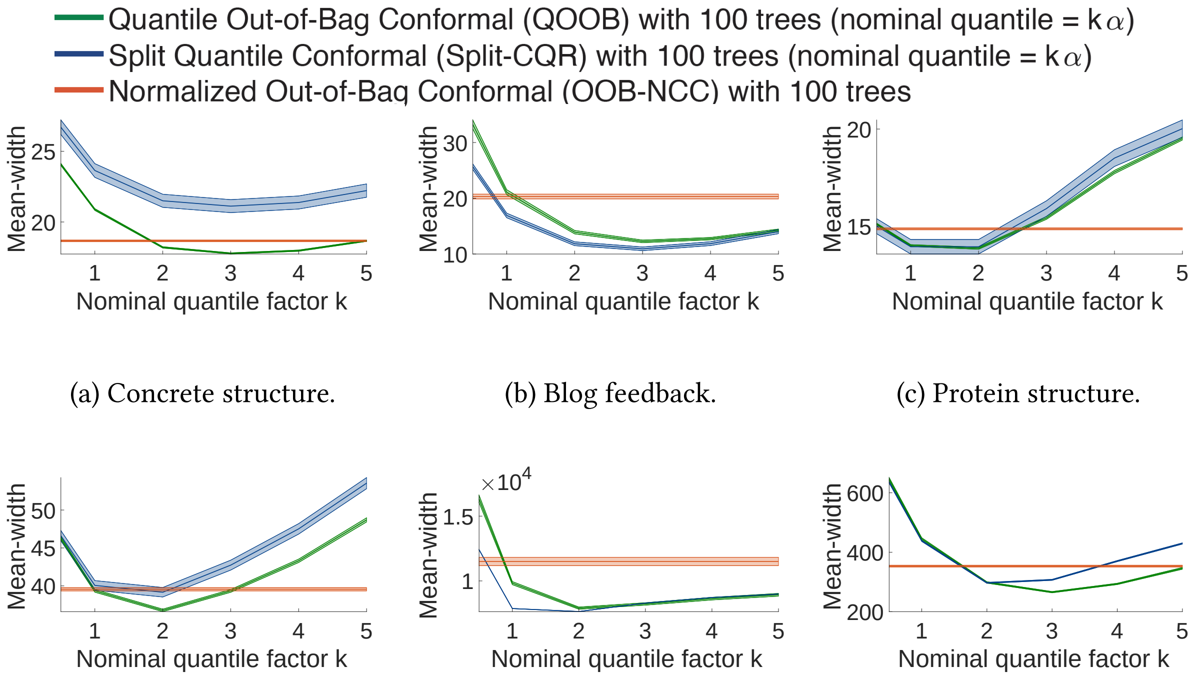
- In Table 9.2, QOOB typically has low values for the standard deviation of the average-mean-width across all methods. This entails more reliability to our method, which may be desirable in some applications. In Sections 9.6.1 and 9.6.2, we observe that this property is true across different number of trees and nominal quantile levels as well.
- On every dataset, QOOB achieves coverage higher than the prescribed value of $1 - \alpha$, with a margin of 1-3%. Surprisingly this is true even if it shortest mean-width among all methods. It may be possible to further improve the performance of QOOB in terms of mean-width by investigating the cause for this over-coverage.
- OOB-CC does better than 8-fold-CC with faster running times. Thus, to develop QOOB, we chose to work with out-of-bag conformal.

We now present additional experiments to demonstrate the following key insights into the behavior of QOOB:

- In Section 9.6.1, we discuss the significant impact that nominal quantile selection has on the performance of QOOB and Split-CQR. We observe that 2α is an appropriate nominal quantile recommendation for both methods.
- In Section 9.6.2, we show that increasing the number of trees T leads to decreasing mean-widths for QOOB, while this is not true for its closest competitor Split-CQR. QOOB also outperforms other competing OOB methods across different values for the number of trees T.
- In Section 9.6.3, we compare QOOB and Split-CQR in the small sample size (small n) regime where we expect sample splitting methods to lose statistical efficiency. We confirm that QOOB significnatly outperforms Split-CQR on all six datasets we have considered for $n \leqslant 100$.
- In Section 9.6.4, we compare the related methods of cross-conformal and jackknife+ and demonstrate that there exist settings where cross-conformal leads to shorter intervals compared to jackknife+, while having a similar computational cost (as discussed in Section 9.3.3).
- In Section 9.6.5, we demonstrate that QOOB achieves conditional coverage on a synthetic dataset. We use the data distribution designed by Romano et al. (2019) for demonstrating the conditional coverage of Split-CQR.

9.6.1 Nominal quantile selection has a significant effect on QOOB

QOOB and Split-CQR both use nominal quantiles $\widehat{q}_\beta$, $\widehat{q}_{1-\beta}$ from a learnt quantile prediction model. In the case of Split-CQR, as observed by Romano et al. (2019) and Sesia and Candès (2020), tuning β leads to improved performance. We perform a comparison of QOOB and Split-CQR at different values of β. Figure 9.1 reports mean-widths for QOOB-100 ($k\alpha$) and Split-CQR-100 ($k\alpha$), with OOB-NCC-100 as a fixed baseline (that does not vary with k). We observe that the nominal quantile level significantly affects the performance of Split-CQR and QOOB. Both methods perform well at the nominal quantile of about 2α. We encourage a more detailed study on the theoretical and empirical aspects of nominal quantile selection in future work. We also note that



(a) Concrete structure.

(b) Blog feedback.

(c) Protein structure.

(d) Superconductivity.

(e) News popularity.

(f) Kernel performance.

Figure 9.1: QOOB and Split-CQR are sensitive to the nominal quantile level $\beta = k\alpha$. At $\beta \approx 2\alpha$, QOOB performs better than OOB-NCC for all datasets (OOB-NCC does not require nominal quantile tuning and is a constant baseline). For the plots above, $\alpha = 0.1$. All methods plotted have empirical mean-coverage at least $1 - \alpha$. The mean-width values are averaged over 100 subsamples. The shaded area denotes ± 1 std-dev for the average of mean-width.

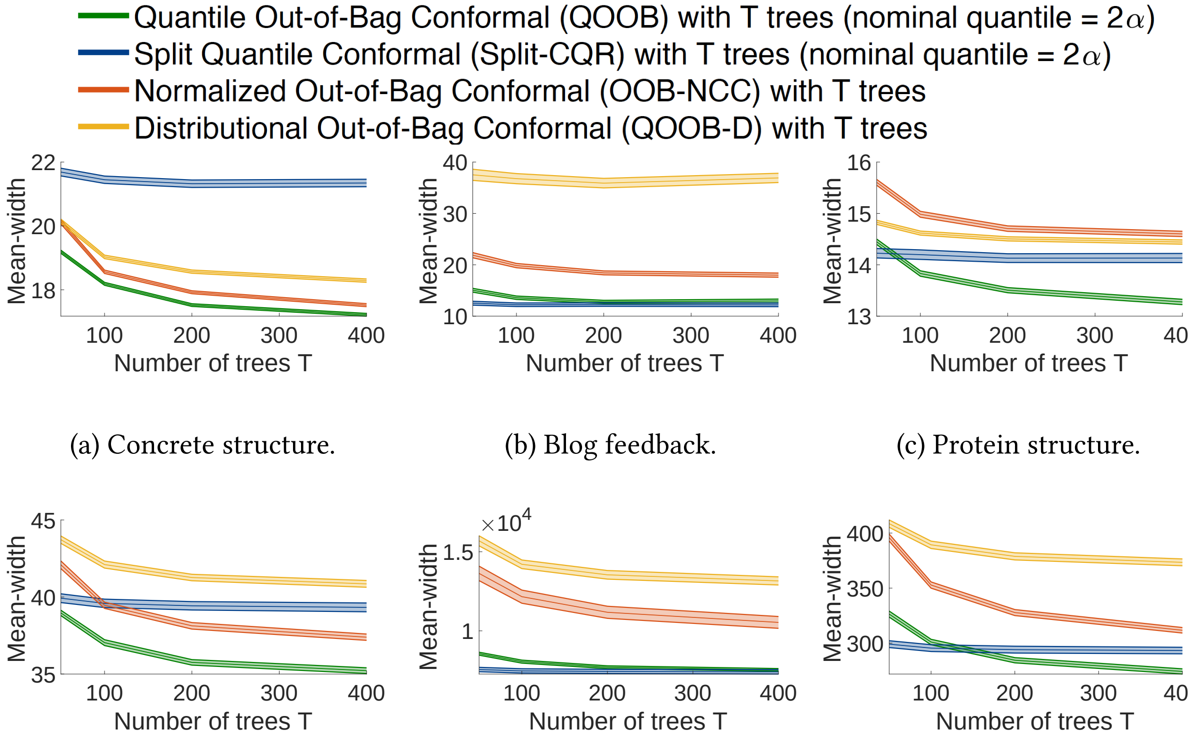
for all values of k, QOOB typically has smaller standard deviation of the average-mean-width compared to Split-CQR, implying more reliability in the predictions.

9.6.2 QOOB has shorter prediction intervals as we increase the number of trees

In this experiment, we investigate the performance of the competitive conformal methods from Table 9.2 as the number of trees T are varied. For QOOB and Split-CQR, we fix the quantile level to $\beta = 2\alpha$. We also compare with OOB-NCC and another quantile based OOB method described as follows. Like QOOB, suppose we are given a quantile estimator $\widehat{q}_s(\cdot)$. Consider the quantile-based construction of nested sets suggested by Chernozhukov et al. (2021):

$$\mathcal{F}_t(x) = [\widehat{q}_t(x), \widehat{q}_{1-t}(x)]_{t \in (0,1/2)}.$$

Using these nested sets and the OOB conformal scheme (Section 9.4) leads to the QOOB-D method (for 'distributional' conformal prediction as the original authors called it). Since QOOB-D



(a) Concrete structure.

(b) Blog feedback.

(c) Protein structure.

(d) Superconductivity.

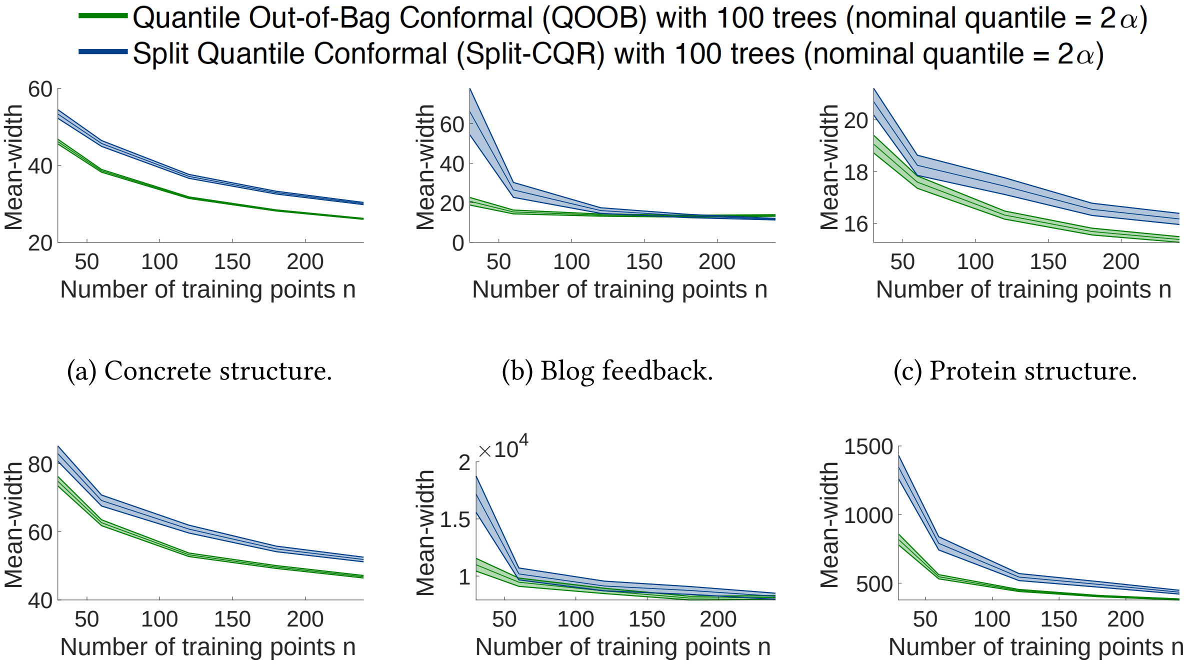
(e) News popularity.

(f) Kernel performance.

Figure 9.2: The performance of QOOB (2α) improves with increasing number of trees T, while the performance of Split-CQR (2α) does not. QOOB (2α) beats every other method except Split-CQR (2α) for all values of T. For the plots above, $\alpha = 0.1$ and all methods plotted have empirical mean-coverage at least $1 - \alpha$. The mean-width values are averaged over 100 iterations. The shaded area denotes ± 1 std-dev for the average of mean-width.

does not require nominal quantile selection, we considered this method as a possible solution to the nominal quantile problem of QOOB and Split-CQR (Section 9.6.1). The results are reported in Figure 9.2 for T ranging from 50 to 400.

We observe that with increasing T, QOOB continues to show improving performance in terms of the width of prediction intervals. Notably, this is not true for Split-CQR, which does not show improving performance beyond 100 trees. In the results reported in Table 9.2, we noted that Split-CQR-100 outperformed QOOB-100 on the blog feedback, news popularity and kernel performance datasets. However, from Figure 9.2 we observe that for $T = 400$, QOOB performs almost the same as Split-CQR on blog feedback and news popularity, and in fact does significantly better than Split-CQR on kernel performance. Further, QOOB shows lower values for the standard deviation of the average-mean-width. The QOOB-D method performs worse than QOOB for every dataset, and hence we did not report it in the other comparisons in this chapter.



(a) Concrete structure.

(b) Blog feedback.

(c) Protein structure.

(d) Superconductivity.

(e) News popularity.

(f) Kernel performance.

Figure 9.3: The performance of QOOB and Split-CQR with varying number of training points n. QOOB has shorter mean-width than Split-CQR across datasets for small n and also smaller standard-deviation of the average mean-width. For the plots above, $\alpha = 0.1$ and all methods plotted have empirical mean-coverage at least $1 - \alpha$. The mean-width values are averaged over 100 iterations. The shaded area denotes ± 1 std-dev for the average of mean-width.

9.6.3 QOOB outperforms Split-CQR at small sample sizes

QOOB needs n times more computation than Split-CQR to produce prediction intervals, since one needs to make n individual predictions. If fast prediction time is desired, our experiments in Sections 9.6.1 and 9.6.2 indicate that Split-CQR is a competitive quick alternative. However, here we demonstrate that at the small sample regime, QOOB significantly outperforms Split-CQR on all six datasets that we have considered.

To make this comparison, we subsample the datasets to a smaller sample size and consider the mean-width and mean-coverage properties of QOOB (2α) and Split-CQR (2α) with $T = 100$. Figure 9.3 contains the results with n ranging from 30 to 240. We observe that at small n, QOOB does significantly better than Split-CQR. This behavior is expected since at smaller values of n, the statistical loss due to sample splitting is most pronounced. Since the overall computation time decreases as n decreases, QOOB is a significantly better alternative in the small sample regime on all fronts.

9.6.4 Cross-conformal outperforms jackknife+

Cross-conformal prediction sets are always smaller than the corresponding jackknife+ prediction sets by construction; see Section 9.3.2 and (9.9). However, the fact that cross-conformal may not give an interval might be of practical importance. In this subsection, we show that the jackknife+ prediction interval can sometimes be strictly larger than the smallest interval containing the cross-conformal prediction set (this is the convex hull of the cross-conformal prediction set and we call it QOOB-Conv).

Table 9.4 reports the performance of QOOB, QOOB-JP, and QOOB-Conv on the blog feedback dataset. Here QOOB-JP refers to the OOB-JP version (9.15). For each of these, we set the nominal quantile level to 0.5 instead of 2α as suggested earlier (this led to the most pronounced difference in mean-widths).

Table 9.4: Mean-width of $C^{\text{OOB}}(x)$, $\text{Conv}(C^{\text{OOB}}(x))$, and $C^{\text{OOB-JP}}(x)$ for the blog feedback dataset with QOOB method. The base quantile estimator is quantile regression forests, and $\alpha = 0.1$. Average values across 100 simulations are reported with the standard deviation in brackets .

Method	Mean-width	Mean-coverage
QOOB-100 (β =0.5)	**14.67 (0.246)**	0.908 (0.002)
QOOB-Conv-100 (β =0.5)	14.73 (0.249)	0.908 (0.002)
QOOB-JP-100 (β =0.5)	15.36 (0.248)	0.911 (0.002)

While this is a specific setting, our goal is to provide a proof of existence. In other settings, cross-conformal style aggregation and jackknife+ style aggregation may have identical prediction sets. However, because the cross-conformal prediction set as well as its convex hull can be computed in nearly the same time (see Section 9.3.3) and have the same marginal validity guarantee, one should always prefer cross-conformal over jackknife+.

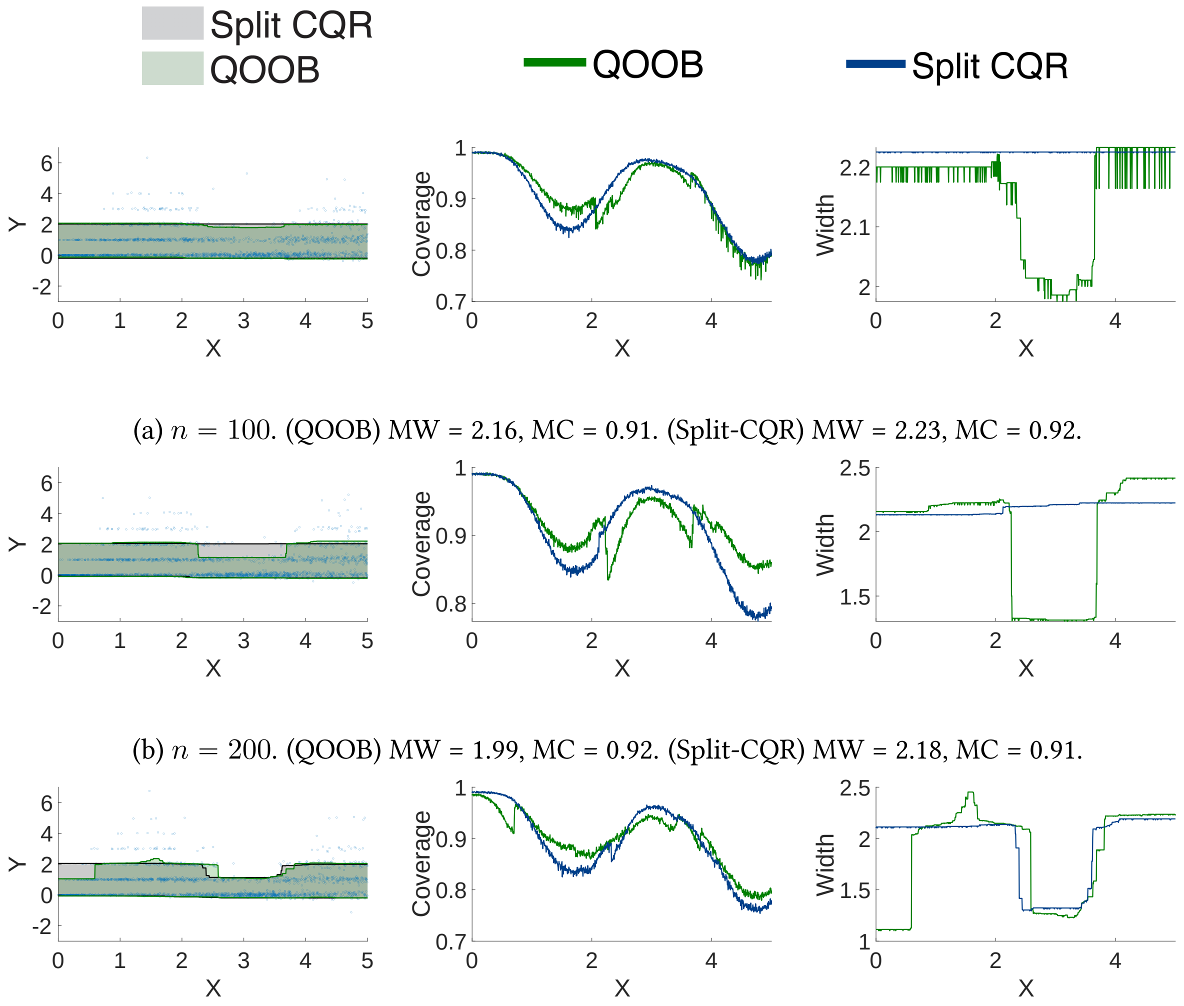
9.6.5 QOOB demonstrates conditional coverage empirically

To demonstrate that Split-CQR exhibits conditional coverage, Romano et al. (2019, Appendix B) designed the following data-generating distribution for P_{XY}:

$$\begin{aligned}&\epsilon_1 \sim N(0,1), \epsilon_2 \sim N(0,1), u \sim \text{Unif}[0,1], \quad \text{and} \quad X \sim \text{Unif}[0,1],\\ &Y \sim \text{Pois}(\sin^2(X) + 0.1) + 0.03X\epsilon_1 + 25\mathbf{1}\{u < 0.01\}\epsilon_2.\end{aligned}$$

We use the same distribution to demonstrate the conditional coverage of QOOB. Additionally, we performed the experiments at a small sample size ($n \leqslant 300$) to understand the effect of sample size on both methods (the original experiments had $n = 2000$, for which QOOB and Split-CQR perform identically). Figure 9.4 summarizes the results.

For this experiment, the number of trees T are set to 100 for both methods. To choose the nominal quantile level, we first ran the python notebook at https://github.com/yromano/cqr to



(a) $n = 100$. (QOOB) MW = 2.16, MC = 0.91. (Split-CQR) MW = 2.23, MC = 0.92.

(b) $n = 200$. (QOOB) MW = 1.99, MC = 0.92. (Split-CQR) MW = 2.18, MC = 0.91.

(c) $n = 300$. (QOOB) MW = 1.86, MC = 0.89. (Split-CQR) MW = 1.94, MC = 0.89.

Figure 9.4: The performance of QOOB-100 and Split-CQR-100 on synthetic data with varying number of training points n ($\alpha = 0.1$). MW refers to mean-width and MC refers to mean-coverage. QOOB shows conditional coverage at smaller values of n than Split-CQR. Section 9.6.5 contains more experimental details.

reproduce the original experiments performed by Romano et al. (2019). Their code first learns a nominal quantile level for Split-CQR by cross-validating. On executing their code, we typically observed values near 0.1 for $\alpha = 0.1$ and hence we picked this nominal quantile level for our experiments as well (for both Split-CQR and QOOB). For our simple 1-dimensional distribution, deeper trees lead to wide prediction sets. This was also observed in the original Split-CQR experiments. To rectify this, the minimum number of training data-points in the tree leaves was set to 40; we do this in our experiments as well.

9.7 Conclusion

We introduced an alternative framework to score-based conformal prediction which is based on a sequence of nested prediction sets. We argued that the nested conformal prediction framework is more natural and intuitive. We demonstrated how to translate a variety of existing nonconformity scores into nested prediction sets. We showed how cross-conformal prediction, the jackknife+, and out-of-bag conformal can be described in our nested framework. The interpretation provided by nested conformal opens up new procedures to practitioners. We propose one such procedure — QOOB — which uses quantile regression forests to perform out-of-bag conformal prediction. We proposed an efficient cross-conformalization algorithm (Algorithm 9.1) that makes cross-conformal as efficient as jackknife+. QOOB relies on this efficient cross-conformalization procedure. We demonstrated empirically that QOOB achieves state-of-the-art performance on multiple real-world datasets.

QOOB has certain limitations. First, without making additional assumptions, we can only guarantee $1-2\alpha$ coverage for QOOB. On the other hand, we observe that in practice QOOB has coverage slightly larger than $1-\alpha$. Second, QOOB is designed specifically for real-valued responses; extending it to classification and other settings would be interesting. Third, QOOB is computationally intensive compared to the competitive alternative Split-CQR (at least when sample-sizes are high; see Section 9.6.3 for more details). We leave the resolution of these limitations to future work.

Appendices for Chapter 9

9.A Equivalence between score-based conformal prediction and nested conformal

We show that every instance of score-based conformal prediction can be cast in terms of nested conformal prediction, and vice versa. 9.A.1 argues this fact for non-transductive conformal methods which are the focus of the main chapter. 9.A.2 describes full transductive conformal prediction using nested sets and argues the equivalence in that setting as well.

9.A.1 Equivalence for non-transductive conformal methods

Non-transductive conformal methods define a score function $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ using some part of the training data (a split, a leave-one-out set, a fold, a subsample, etc). Thus unlike full (tranductive) conformal, the r does not depend on the point (x, y) that it is applied to (of course the value $r(x, y)$ depends on (x, y) but not r itself).

Given a nested family, $\{\mathcal{F}_t(\cdot)\}_{t \in \mathbb{R}}$ learnt on some part of the data, a nonconformity score can be constructed per equation (9.2). We now argue the other direction. Given any nonconformity score r and an $x \in \mathcal{X}$, consider the family of nested sets $\{\mathcal{F}_t(x)\}_{t \in \mathbb{R}}$ defined as:

$$\mathcal{F}_t(x) := \{y \in \mathcal{Y} : r(x, y) \leqslant t\}.$$

Clearly, $y \in \mathcal{F}_t(x)$ if and only if $r(x, y) \leqslant t$. Hence,

$$\inf \{t \in \mathcal{T} : \ y \in \mathcal{F}_t(x)\} = \inf \{t \in \mathcal{T} : \ r(x, y) \leqslant t\} = r(x, y).$$

Thus, for any nonconformity score r, there exists a family of nested sets that recovers it.

The randomness in r (through the data on which it is learnt) is included in the conformal validity guarantees of split conformal, cross-conformal, OOB conformal, etc. Similarly, the guarantees based on nested conformal implicitly include the randomness in $\{\mathcal{F}_t(\cdot)\}_{t \in \mathcal{T}}$.

9.A.2 An equivalent formulation of full transductive conformal using the language of nested sets

For simpler exposition, in this subsection, we skip the qualification ‘transductive’ and just say ‘conformal’. Each instance of conformal refers to transductive conformal. We follow the description of a conformal prediction set as defined by Balasubramanian et al. (2014). Conformal can be defined for any space $\mathcal{Z}$; in the predictive inference setting of this chapter, one can think of $z = (x, y) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.

A (non-)conformity N-measure is a measurable function A that assigns every sequence $(z_1, \ldots, z_N)$ of N examples to a corresponding sequence $(\alpha_1, \ldots, \alpha_N)$ of N real numbers that is equivariant with respect to permutations, meaning that for any permutation $\pi : [N] \to [N]$,

$$(\alpha_1, \ldots, \alpha_N) = A(z_1, \ldots, z_N) \quad \Rightarrow \quad (\alpha_{\pi(1)}, \ldots, \alpha_{\pi(N)}) = A(z_{\pi(1)}, \ldots, z_{\pi(N)}).$$

For some training set $z_1, z_2, \ldots, z_n$ and a candidate point z, define the nonconformity score as

$$(\alpha_1^z, \ldots, \alpha_{n+1}^z) := A(z_1, \ldots, z_n, z).$$

For each z, define

$$p^z := \frac{|\{i \in [n+1] : \alpha_i^z \geqslant \alpha_{n+1}^z\}|}{n+1}.$$

Then the *conformal prediction set* determined by A as a nonconformity measure is defined by

$$\Gamma^\alpha(z_1, \ldots, z_n) := \{z : p^z > \alpha\}. \tag{9.18}$$

In predictive inference, we have a fixed x and wish to learn a prediction set for y. This set takes the form

$$\{y : p^{(x,y)} > \alpha\}.$$

If the training and test-data are exchangeable, it can be shown that the above prediction set is marginally valid at level α (Proposition 1.2 (Balasubramanian et al., 2014)).

The nested (transductive) conformal predictor starts with a nested sequence of sets. For any $N \geqslant 1$ and sequence $(z_1, \ldots, z_N)$, let $\{\mathcal{F}_t(z_1, \ldots, z_N)\}_{t \in \mathcal{T}}$ be a sequence of nested sets that are invariant to permutations of indices. For observations $Z_1, \ldots, Z_n$ and a possible future z, define the scores

$$\begin{aligned} r_i^z &:= \inf\{t \in \mathcal{T} : Z_i \in \mathcal{F}_t(\{Z_1, \ldots, Z_n, z\})\}, \quad i \in [n], \\ r_{n+1}^z &:= \inf\{t \in \mathcal{T} : z \in \mathcal{F}_t(\{Z_1, \ldots, Z_n, z\})\}. \end{aligned}$$

The nested conformal predictor is then given by

$$C_\alpha(Z_1, \ldots, Z_n) := \{z : p^z > \alpha\}, \quad \text{where} \quad p^z := \frac{|\{i \in [n+1] : r_i^z \geqslant r_{n+1}^z\}|}{n+1}.$$

It is clear that nested conformal prediction is a special case of (transductive) conformal prediction with scores defined based on nested sets rather than a function A. Below, we prove that the converse also holds.

Proposition 9.3. *Suppose $\{\Gamma^\alpha(Z_1, \ldots, Z_n)\}_{\alpha \in [0,1]}$ represents a conformal prediction set. Then there exists a nested sequence such that the nested conformal set matches with the conformal prediction set.*

Proof of Proposition 9.3. For any $N \geqslant 1$ and $w_1, \ldots, w_N$, define $\mathcal{F}_t(w_1, \ldots, w_N) := \Gamma^{1-t}(w_1, \ldots, w_N)$ for $t \in [0,1]$. From the definition (9.18), it is clear that $\mathcal{F}_t$ is increasing in $t \in [0,1]$. For notational convenience, set $\mathcal{Z} := \{Z_1, \ldots, Z_n, z\}$. Now define scores

$$\begin{aligned} r_i^z &:= \inf\{t \in [0,1] : Z_i \in \mathcal{F}_t(\mathcal{Z}\backslash\{Z_i\})\}, \quad \text{for} \quad i \in [n], \\ r_{n+1}^z &:= \inf\{t \in [0,1] : z \in \mathcal{F}_t(\mathcal{Z}\backslash\{z\})\}. \end{aligned}$$

The statement $Z_i \in \mathcal{F}_t(\mathcal{Z}\backslash\{Z_i\})$ is equivalent to

$$\frac{|\{j \in [n+1] : \alpha_j^z \geqslant \alpha_i^z\}|}{n+1} > 1 - t.$$

This equivalence follows from the fact that A is a nonconformity score and hence equivariant to permutations. Therefore,

$$\begin{aligned} r_i^z &= 1 - \frac{|\{j \in [n+1] : \alpha_j^z \geqslant \alpha_i^z\}|}{n+1} \\ &= \frac{n - |\{j \in [n+1]\backslash\{i\} : \alpha_j^z \geqslant \alpha_i^z\}|}{n+1} = \frac{|\{j \in [n+1]\backslash\{i\} : \alpha_j^z < \alpha_i^z\}|}{n+1}. \end{aligned}$$

Because $(r_1^z, \ldots, r_{n+1}^z)$ is an increasing ranking transformation of $(\alpha_1^z, \ldots, \alpha_{n+1}^z)$, the p^z definitions based on (α_i^z) or (r_i^z) are equal. Hence for any conformal prediction set there exists an equivalent nested conformal prediction set. □

Proposition 1.3 of Balasubramanian et al., 2014 shows that conformal prediction is universal in a particular sense (informally, any valid scheme for producing assumption-free confidence sets can be replaced by a conformal prediction scheme that is at least as efficient). Since everything that can be accomplished via nested conformal prediction can also be done via conformal prediction and vice versa, nested conformal prediction is also universal in the same sense.

9.B K-fold cross-conformal and CV+ using nested sets

In Section 9.3 we rephrased leave-one-out cross-conformal and jackknife+ in the nested framework. In this section, we will now describe their K-fold versions.

9.B.1 Extending K-fold cross-conformal using nested sets

Suppose $S_1, \ldots, S_K$ denotes a disjoint partition of $\{1, 2, \ldots, n\}$ such that $|S_1| = |S_2| = \cdots = |S_K|$. For exchangeability, this equality of sizes is very important. Let $m = n/K$ (assume m is an integer). Let $\{\mathcal{F}_t^{-S_k}(x)\}_{t \in \mathcal{T}}$ be a sequence of nested sets computed based on $\{1, 2, \ldots, n\}\backslash S_k$. Define the score

$$r_i(x, y) \;:=\; \inf\left\{t \in \mathcal{T} : y \in \mathcal{F}_t^{-S_{k(i)}}(x)\right\},$$

where $k(i) \in [K]$ is such that $i \in S_{k(i)}$. The cross-conformal prediction set is now defined as

$$C_K^{\text{cross}}(x) \;:=\; \left\{y : \sum_{i=1}^{n} \mathbb{1}\{r_i(X_i, Y_i) < r_i(x, y)\} < (1-\alpha)(n+1)\right\}.$$

It is clear that if $K = n$ then $C_K^{\text{cross}}(x) = C^{\text{LOO}}(x)$ for every x. The following result proves the validity of $C_K^{\text{cross}}(\cdot)$ as an extension of Theorem 4 of Barber et al. (2021). This clearly reduces to Theorem 9.1 if $K = n$.

Theorem 9.3. *If $(X_i, Y_i), i \in [n] \cup \{n+1\}$ are exchangeable and sets $\mathcal{F}_t^{-S_k}(x)$ constructed based on $\{(X_i, Y_i) : i \in [n]\backslash S_k\}$ are invariant to their ordering, then*

$$\mathbb{P}(Y_{n+1} \in C_K^{cross}(X_{n+1})) \;\geqslant\; 1 - 2\alpha - \min\left\{\frac{1-K/n}{K+1}, \frac{2(K-1)(1-\alpha)}{n+K}\right\}.$$

See 9.E.2 for a proof. Although the construction of $C_K^{\text{cross}}(x)$ is based on a K fold split of the data. The form is exactly the same as that of $C^{\text{LOO}}(x)$ in (9.6). Hence the computation of $C_K^{\text{cross}}(x)$ can be done based on the discussion in Section 9.3.3. In particular, if each of the nested sets $\mathcal{F}_t(x)$ are either intervals or empty sets, the $C_K^{\text{cross}}(x)$ aggregation step (after computing the residuals) can be performed in time $O(n \log n)$.

9.B.2 Extending CV+ using nested sets

The prediction sets $C^{\text{cross}}(x)$ and $C_K^{\text{cross}}(x)$ are defined implicitly and are in general not intervals. The sets $C^{\text{cross}}(x)$ and $C_K^{\text{cross}}(x)$ can be written in terms of nested sets as

$$C_K^{\text{cross}}(x) \;:=\; \left\{y : \sum_{i=1}^{n} \mathbb{1}\left\{y \notin \mathcal{F}_{r_i(X_i,Y_i)}^{-S_{k(i)}}(x)\right\} < (1-\alpha)(n+1)\right\}.$$

In this subsection, we show that there exists an explicit interval that always contains $C_K^{\text{cross}}(x)$ whenever $\{\mathcal{F}_t(x)\}_{t\in\mathcal{T}}$ is a collection of nested intervals (instead of just nested sets). This is a generalization of the CV+ interval defined by Barber et al. (2021). The discussion of this subsection can be extended to the case whenever the nested sets are either intervals or the empty set just like we did for leave-one-out cross-conformal and jackknife+ in 9.D.

If each $\mathcal{F}_t(x)$ is an interval, we can write $\mathcal{F}_{r_i(X_i,Y_i)}^{-S_{k(i)}}(x) = [\ell_i(x), u_i(x)]$ for some $\ell_i(\cdot), u_i(\cdot)$. Using this notation, we can write $C_K^{\text{cross}}(x)$ as

$$C_K^{\text{cross}}(x) \;:=\; \left\{y : \sum_{i=1}^{n} \mathbb{1}\{y \notin [\ell_i(x), u_i(x)]\} < (1-\alpha)(n+1)\right\}.$$

Following the same arguments as in Section 9.3.2 we can define the CV+ prediction interval as follows

$$C_K^{\text{CV+}}(x) \;:=\; [q_{n,\alpha}^{-}(\ell_i(x)), q_{n,\alpha}^{+}(u_i(x))] \;\supseteq\; C_K^{\text{cross}}(x), \tag{9.19}$$

where $q^-_{n,\alpha}(\ell_i(x))$ denotes the $\lfloor\alpha(n+1)\rfloor$-th smallest value of $\{\ell_i(x)\}_{i=1}^n$. and $q^+_{n,\alpha}(u_i(x))$ denotes the $\lceil(1-\alpha)(n+1)\rceil$-th smallest value of $\{u_i(x)\}_{i=1}^n$. For $K = n$, $C_K^{\text{CV+}}(x)$ reduces to the jackknife+ prediction interval $C^{\text{JP}}(x)$. $C^{\text{CV+}}(x)$ and $C^{\text{JP}}(x)$ are always non-empty intervals if each of the $\mathcal{F}_t(x)$ are non-empty intervals. Because $C_K^{\text{cross}}(x) \subseteq C_K^{\text{CV+}}(x)$ for all x, we obtain a validity guarantee from Theorem 9.3: for all $2 \leqslant K \leqslant n$,

$$\mathbb{P}\left(Y_{n+1} \in C_K^{\text{CV+}}(X_{n+1})\right) \geqslant 1 - 2\alpha - \min\left\{\frac{1-K/n}{K+1}, \frac{2(K-1)(1-\alpha)}{n+K}\right\}.$$

Note that the convex hull of the K-fold cross-conformal prediction set is also an interval smaller than the CV+ interval

$$C_K^{\text{cross}}(x) \subseteq \text{Conv}(C_K^{\text{cross}}(x)) \subseteq C_K^{\text{CV+}}(x).$$

In some cases, the containment above can be strict, and hence we recommend the K-fold cross conformal or its convex hull over CV+.

9.C Nested conformal based on multiple repetitions of splits

In Section 9.2, we described the nested conformal version of split conformal which is based on one particular split of the data into two parts, and in 9.B we discussed partitions of the data into $2 \leqslant K \leqslant n$ parts. In practice, however, to reduce the additional variance due to randomization, one might wish to consider several (say M) different splits of data into two parts combine these predictions. Lei et al. (2018) discuss a combination of M split conformal prediction sets based on Bonferroni correction and in this section, we consider an alternative combination method that we call *subsampling conformal.* The same idea can also be used for cross-conformal version where the partition of the data into K folds can be repeatedly performed M times. The methods to be discussed are related to those proposed by Carlsson et al. (2014), Vovk (2015), and Linusson et al. (2017) and Linusson et al. (2019), but these papers do not provide validity results for their methods.

9.C.1 Subsampling conformal based on nested prediction sets

Fix a number $K \geqslant 1$ of subsamples. Let $M_1, M_2, \ldots, M_K$ denote independent and identically distributed random sets drawn uniformly from $\{M : M \subset [n]\}$; one can also restrict to $\{M : M \subset [n], |M| = m\}$ for some $m \geqslant 1$. For each set M_k, define the p-value for the new prediction y at X_{n+1} as

$$p_k^y(x) := \frac{|\{i \in M_k^c : r_k(x,y) \leqslant r_k(X_i,Y_i)\}| + 1}{|M_k^c| + 1},$$

where the scores $r_k(X_i,Y_i)$ and $r_k(x,y)$ are computed as

$$r_k(X_i,Y_i) := \inf\{t \in \mathcal{T} : Y_i \in \mathcal{F}_t^{M_k}(X_i)\}, \quad i \in M_k^c,$$

$$r_k(x, y) := \inf\{t \in \mathcal{T} : y \in \mathcal{F}_t^{M_k}(x)\},$$

based on nested sets $\{\mathcal{F}_t^{M_k}(x)\}_{t\in\mathcal{T}}$ computed based on observations in M_k. Define the prediction set as

$$C_K^{\texttt{subsamp}}(x) := \left\{y : \frac{1}{K}\sum_{k=1}^{K} \frac{|\{i \in M_k^c : r_k(x, y) \leqslant r_k(X_i, Y_i)\}| + 1}{|M_k^c| + 1} > \alpha\right\}.$$

It is clear that for $K = 1$, $C_K^{\texttt{subsamp}}(x)$ is same as the split conformal prediction set discussed in Section 9.2. The following results proves the validity of $C_K^{\texttt{subsamp}}(\cdot)$.

Theorem 9.4. *If* (X_i, Y_i), $i \in [n] \cup \{n + 1\}$ *are exchangeable, then for any* $\alpha \in [0, 1]$ *and* $K \geqslant 1$, $\mathbb{P}\left(Y_{n+1} \notin C_K^{\texttt{subsamp}}(X_{n+1})\right) \leqslant \min\{2, K\}\alpha$.

See 9.E.6 for a proof. Note that we can write $p_k^y(x)$ as $p^y(x; M_k)$ by adding the argument for observations used in computing the nested sets. Using this notation, we can write for K large

$$\frac{1}{K}\sum_{k=1}^{K} p^y(x; M_k) \;\approx\; \mathbb{E}_M[p^y(x; M)], \tag{9.20}$$

where the expectation is taken with respect to the random "variable" M drawn uniformly from a collection of subsets of $[n]$ such as $\{S : S \subset [n]\}$ or $\{S : S \subset [n], |S| = m\}$ for some $m \geqslant 1$. Because any uniformly drawn element in $\{S : S \subset [n]\}$ can be obtained by sampling from $[n]$ without replacement (subsampling), the above combination of prediction intervals can be thought as subbagging introduced in (Bühlmann and Yu, 2002).

Lei et al. (2018, Section 2.3) combine the p-values $p_k^y(x)$ by taking the minimum. They define the set

$$C_K^{\texttt{split}}(x) := \bigcap_{k=1}^{K}\{y : p_k^y(x) > \alpha/K\} \;=\; \left\{y : K \min_{1\leqslant k\leqslant K} p_k^y(x) > \alpha\right\}.$$

Because $p_1^y(x), p_2^y(x), \ldots, p_K^y(x)$ are independent and identically distributed (conditional on the data), averaging is a natural stabilizer than the minimum; all the p-values should get equal contribution towards the stabilizer but the minimum places all its weight on one p-value.

Vovk (2015, Appendix B) describes a version of $C_K^{\texttt{subsamp}}(\cdot)$ using bootstrap samples instead of subsamples and this corresponds to bagging. We consider this version in the following subsection.

9.C.2 Bootstrap conformal based on nested prediction sets

The subsampling prediction set $C_K^{\texttt{subsamp}}(x)$ is based on sets M_k obtained by sampling without replacement. Statistically a more popular alternative is to form sets M_k by sampling with replacement, which corresponds to bootstrap.

Let $B_1, \ldots, B_K$ denote independent and identically distributed bags (of size m) obtained by random sampling with replacement from $[n] = \{1, 2, \ldots, n\}$. For each $1 \leqslant k \leqslant K$, consider scores

$$\begin{aligned} r_k(X_i, Y_i) &:= \inf\{t \in \mathcal{T} : Y_i \in \mathcal{F}_t^{B_k}(X_i)\}, \quad i \in B_k^c, \\ r_k(x, y) &:= \inf\{t \in \mathcal{T} : y \in \mathcal{F}_t^{B_k}(x)\}, \end{aligned}$$

based on nested sets $\{\mathcal{F}_t^{B_k}(x)\}_{t \in \mathcal{T}}$ computed based on observations $(X_i, Y_i), i \in B_k$; B_k should be thought of as a bag rather than a set of observations because of repititions of indices. Consider the prediction interval

$$C_{\alpha,K}^{\text{boot}}(x) := \left\{ y : \frac{1}{K} \sum_{k=1}^{K} \frac{|\{i \in [n] \backslash B_k : r_k(x, y) \leqslant r_k(X_i, Y_i)\}| + 1}{|[n] \backslash B_k| + 1} > \alpha \right\}.$$

This combination of prediction interval based on bootstrap sampling is a version of bagging and was considered in Vovk (2015, Appendix B). The following result proves a validity bound for $C_{\alpha,K}^{\text{boot}}(X_{n+1})$.

Theorem 9.5. *If $(X_i, Y_i), i \in [n] \cup \{n+1\}$ are exchangeable, then for any $\alpha \in [0, 1]$ and $K \geqslant 1$, $\mathbb{P}(Y_{n+1} \notin C_{\alpha,K}^{boot}(X_{n+1})) \leqslant \min\{2, K\}\alpha$.*

See 9.E.6 for a proof. Carlsson et al. (2014, Proposition 1) provide a similar result in the context of aggregated conformal prediction but require an additional consistent sampling assumption. The computation of the subsampling and the bootstrap conformal prediction sets is no different from that of cross-conformal and the techniques discussed in susbection 9.3.3 are still applicable.

Linusson et al. (2019) demonstrated that aggregated conformal methods tend to be conservative. We also observed this in our simulations. Because of this, we did not present these methods in our experiments.

9.D Cross-conformal and Jackknife+ if $\mathcal{F}_t(x)$ could be empty

The definition of cross-conformal (9.6) is agnostic to the interval interpretation through $\mathcal{F}_{r_i(X_i,Y_i)}^{-i}(x)$ since $r_i(X_i, Y_i)$ and $r_i(x, y)$ are well-defined irrespective of whether $\mathcal{F}_t(x)$ is an interval or not. However, the discussion in Sections 9.3.2 and 9.3.3 indicates that the interval interpretation is useful for interpretability as well as to be able to compute $C^{\text{LOO}}(x)$ efficiently. In these subsections, we assumed that $\mathcal{F}_t(x)$ is always an interval. However there exist realistic scenarios in which $\mathcal{F}_t(x)$ is always either an interval or an empty set. Fortunately, it turns out that the discussion about jackknife+ and efficient cross-conformal computation can be generalized to this scenario as well.

9.D.1 When can $\mathcal{F}_t(x)$ be empty?

Consider the quantile estimate based set entailed by the CQR formulation of Romano et al. (2019): $\mathcal{F}_t(x) = [\widehat{q}_{\alpha/2}(x) - t, \widehat{q}_{1-\alpha/2}(x) + t]$. $\mathcal{F}_t(x)$ is implicitly defined as the empty set if

$t < 0.5(\widehat{q}_{\alpha/2}(x) - \widehat{q}_{1-\alpha/2}(x))$. Notice that since t can be negative, the problem we are considering is different from the quantile crossing problem which has separately been discussed by Romano et al. (2019, Section 6), and may occur even if the quantile estimates satisfy $\widehat{q}_{\alpha/2}(x) \leqslant \widehat{q}_{1-\alpha/2}(x)$ for every x. In the cross-conformal or jackknife+ setting, $\mathcal{F}^{-i}_{r_i(X_i,Y_i)}(x)$ is empty for a test point x if

$$r_i(X_i, Y_i) < 0.5(\widehat{q}^{-i}_{\alpha/2}(x) - \widehat{q}^{-i}_{1-\alpha/2}(x)),$$

where $\widehat{q}^{-i}$ are the quantile estimates learnt leaving (X_i, Y_i) out. If the above is true, it implies that $r_i(X_i, Y_i) < r_i(x, y)$ for every possible y. From the conformal perspective, the interpretation is that (x, y) is more 'non-conforming' than (X_i, Y_i) for every $y \in \mathcal{Y}$. In our experiments, we observe this does occur occasionally for cross-conformal (or out-of-bag conformal described in Section 9.4) with quantile-based nested sets. In hindsight, it seems reasonable that this would happen at least once across multiple training and test points.

9.D.2 Jackknife+ and efficiently computing $C^{\text{LOO}}(x)$ in the presence of empty sets

Suppose $\mathcal{F}_t(x)$ is an interval whenever it is non-empty. Define

$$\Lambda_x := \{i : \mathcal{F}^{-i}_{r_i(X_i,Y_i)}(x) \text{ is not empty}\},$$

Equivalently we can write $\Lambda_x := \{i : \exists y, y \in [\ell_i(x), u_i(x)]\}$. The key observation of this section is that for jackknife+ and the computation, only the points in Λ_x need to be considered. To see this, we re-write the interval definition of the cross-conformal prediction (9.7):

$$\begin{aligned} C^{\text{LOO}}(x) &= \left\{ y : \alpha(n+1) - 1 < \sum_{i=1}^{n} \mathbb{1}\{y \in [\ell_i(x), r_i(x)]\} \right\} \\ &= \left\{ y : \alpha(n+1) - 1 < \sum_{i \in \Lambda_x} \mathbb{1}\{y \in [\ell_i(x), r_i(x)]\} \right\}, \end{aligned} \tag{9.21}$$

since if $i \notin \Lambda_x$, no y satisfies $y \in [\ell_i(x), u_i(x)]$. Following the same discussion as in Section 9.3.2, we can define the jackknife+ prediction interval as

$$C^{\text{JP}}(x) \ := \ [q^-_{n,\alpha}(\{\ell_i(x)\}_{i\in\Lambda_x}), \ -q^-_{n,\alpha}(\{-u_i(x)\}_{i\in\Lambda_x})]$$

where $q^-_{n,\alpha}(\{\ell_i(x)\}_{i\in\Lambda_x})$ denotes the $\lfloor\alpha(n+1)\rfloor$-th smallest value of $\{\ell_i(x)\}_{i\in\Lambda_x}$ and $q^-_{n,\alpha}(\{-u_i(x)\}_{i\in\Lambda_x})$ denotes the $\lfloor\alpha(n+1)\rfloor$-th smallest value of $\{-u_i(x)\}_{i\in\Lambda_x}$ (if $|\Lambda_x| = n$ the above definition can be verified to be exactly the same as the one provided in equation (9.8)). It may be possible that $\lfloor\alpha(n+1)\rfloor > |\Lambda_x|$ in which case the jackknife+ interval (and the cross-conformal prediction set) should be defined to be empty. The $1 - 2\alpha$ coverage guarantee continues to hold marginally even though we may sometimes return empty sets.

To understand the computational aspect, we note that the discussion of Section 9.3.3 continues to hold with $\mathcal{Y}^x$ (equation (9.11)) redefined to only include the intervals end-points for intervals which are defined:

$$\mathcal{Y}^x := \bigcup_{i \in \Lambda_x} \wr \ell_i(x), u_i(x) \wr, \tag{9.22}$$

and $\mathcal{S}^x$ defined only for these points. This also follows from the re-definition of $C^{\text{LOO}}(x)$ in (9.21). With the definition above, Algorithm 9.1 works generally for the case where $\mathcal{F}_t(x)$ could be an interval or an empty set.

9.E Proofs

9.E.1 Proofs of results in Section 9.2

Proof of Proposition 9.1. Set $r_{n+1} := r(X_{n+1}, Y_{n+1})$. By the construction of the prediction interval, we have

$$Y_{n+1} \in C(X_{n+1}) \quad \text{if and only if} \quad r_{n+1} \leqslant Q_{1-\alpha}(r, \mathcal{I}_2).$$

Hence

$$\mathbb{P}(Y_{n+1} \in C(X_{n+1}) \big| \{(X_i, Y_i) : i \in \mathcal{I}_1\}) = \mathbb{P}(r_{n+1} \leqslant Q_{1-\alpha}(r, \mathcal{I}_2) \big| \{(X_i, Y_i) : i \in \mathcal{I}_1\}).$$

Exchangeability of $(X_i, Y_i), i \in [n] \cup \{n+1\}$ implies the exchangeability of $(X_i, Y_i), i \in \mathcal{I}_2 \cup \{n+1\}$ conditional on $(X_i, Y_i), i \in \mathcal{I}_1$. This in turn implies that $r_i, i \in \mathcal{I}_2 \cup \{n+1\}$ are also exchangeable (conditional on the first split of the training data) and thus Lemma 2 of Romano et al. (2019) yields

$$\mathbb{P}(r_{n+1} \leqslant Q_{1-\alpha}(r, \mathcal{I}_2) | \{(X_i, Y_i) : \ i \in \mathcal{I}_1\}) \geqslant 1 - \alpha,$$

and the assumption of almost sure distinctness of $r_1, \ldots, r_n$ implies (9.4). □

9.E.2 Proof of Theorem 9.1

Define the matrix $D \in \mathbb{R}^{(n+1)\times(n+1)}$ with entries

$$D_{i,j} := \begin{cases} +\infty, & \text{if } i = j, \\ r_{(i,j)}(X_i, Y_i), & \text{if } i \neq j, \end{cases}$$

where $r_{(i,j)}(x, y) := \inf\{t \in \mathcal{T} : \ y \in \mathcal{F}_t^{-(i,j)}(x)\}$, with $\mathcal{F}_t^{-(i,j)}(x)$ defined analogues to $\mathcal{F}_t^{-i}(x)$ computed based on $\{(X_k, Y_k) : \ k \in [n+1] \backslash \{i, j\}\}$. It is clear that $D_{i,n+1} = r_i(X_i, Y_i)$, and $D_{n+1,i} = r_i(X_{n+1}, Y_{n+1})$. Therefore,

$$Y_{n+1} \notin C^{\text{LOO}}(X_{n+1}) \quad \text{if and only if} \quad (1-\alpha)(n+1) \leqslant \sum_{i=1}^{n+1} \mathbb{1}\{D_{i,n+1} < D_{n+1,i}\},$$

which holds if and only if $n+1 \in \mathcal{I}(D)$, with $\mathcal{I}(D)$ is defined as in (9.23). Hence from Theorem 9.6 the result is proved, if its assumption is verified. This assumption follows from the fact that $\mathcal{F}_t^{-(i,j)}(x)$ treats its training data symmetrically. □

9.E.3 Proof of Theorem 9.2

The OOB-conformal procedure treats the training data $\{(X_i, Y_i)\}_{i\in[n]}$ exchangeably but not the test point (X_{n+1}, Y_{n+1}). To prove a validity guarantee, we first lift the OOB-conformal procedure to one that treats all points $\{(X_i, Y_i)\}_{i\in[n+1]}$ exchangeably. This is the reason we require a random value of K, as will be evident shortly.

The lifted OOB-conformal method is described as follows. Construct a collection of sets $\{\widetilde{M}_i\}_{i=1}^{\widetilde{K}}$, where each $\widetilde{M}_i$ is independently drawn using bagging or subsampling m samples from $[n+1]$ (instead of $[n]$). Following this, for every $(i,j) \in [n+1] \times [n+1]$ with $i \neq j$ define $\{\mathcal{F}_t^{-(i,j)}\}_{t\in\mathcal{T}}$ as the sequence of nested sets learnt by ensembling samples $M_{-(i,j)} := \{M_k : i, j \notin M_k\}$. The nested sets $\{\mathcal{F}_t^{-(i,j)}\}_{t\in\mathcal{T}}$ are then used to compute residuals on (X_i, Y_i) and (X_j, Y_j): we define a matrix $D \in \mathbb{R}^{(n+1)\times(n+1)}$ with entries

$$D_{i,j} := \begin{cases} +\infty, & \text{if } i = j, \\ r_{(i,j)}(X_i, Y_i), & \text{if } i \neq j, \end{cases}$$

where $r_{(i,j)}(x, y) := \inf\{t \in \mathcal{T} : \ y \in \mathcal{F}_t^{-(i,j)}(x)\}$.

We will now invoke Theorem 9.6 for the exchangeable random variables $\{Z_i = (X_i, Y_i)\}_{i=1}^{n+1}$. The assumption of Theorem 9.6 holds for the elements $D_{i,j}$ since the ensemble method we use to learn $\{\mathcal{F}_t^{-(i,j)}\}_{t\in\mathcal{T}}$ treats all random variables apart from Z_i, Z_j symmetrically. Thus for every $j \in [n+1]$,

$$P(j \notin \mathcal{I}(D)) \leqslant 2\alpha - \frac{1}{n+1} \leqslant 2\alpha,$$

for $\mathcal{I}(D)$ defined in (9.23). We will now argue that for $i \in [n]$, $D_{i,n+1} = r_i(X_i, Y_i)$, and $D_{n+1,i} = r_i(X_{n+1}, Y_{n+1})$. Notice that for every $j \in [\widetilde{K}]$,

$$\text{if we use bagging: } P(n+1 \notin \widetilde{M}_j) = \left(1 - \frac{1}{n+1}\right)^m;$$

$$\text{if we use subsampling: } P(n+1 \notin \widetilde{M}_j) = \left(1 - \frac{m}{n+1}\right).$$

and so $K = |\{j : n+1 \notin \widetilde{M}_j\}| \sim \text{Bin}(\widetilde{K}, (1 - \frac{1}{n+1})^m)$ for bagging and $K \sim \text{Bin}(\widetilde{K}, 1 - \frac{m}{n+1})$ for subsampling. Evidently, we can conclude that conditioned on the set $\{j : n+1 \notin \widetilde{M}_j\}$, $\{M_j\}_{j=1}^K \overset{d}{=} \{\widetilde{M}_j : n+1 \notin \widetilde{M}_j\}$. In other words, the OOB-conformal bagging or subsampling procedure is embedded in its lifted version. Therefore,

$$\begin{aligned} P(Y_{n+1} \notin C^{\text{OOB}}(X_{n+1})) &= P((1-\alpha)(n+1) \leqslant \sum_{i=1}^{n+1} \mathbb{1}\{D_{i,n+1} < D_{n+1,i}\}) \\ &= P(n+1 \in \mathcal{I}(D)) \qquad\qquad \text{(per definition (9.23))}, \end{aligned}$$

which as we have shown happens with probability at most 2α. This completes the proof. □

9.E.4 Proof of Proposition 9.2

To see why Algorithm 9.1 works, we describe it step by step along with variable definitions. To simplify understanding, assume that $\mathcal{Y}^x$ does not contain repeated elements (Algorithm 9.1 continues to correctly compute the cross-conformal prediction set even if this is not true; the requirement mentioned on the ordering of elements in $\mathcal{Y}^x$ before definition (9.13) is crucial for Algorithm 9.1 to remain correct with repeated elements). As we make a single pass over $\mathcal{Y}^x$ in sorted order, at every iteration i, when we are on line 6 or line 10, the variable *count* stores the number of training points that are more nonconforming than (x, y_i^x); *count* is increased by 1 whenever a left end-point is seen (line 5) and is decreased by 1 after a right end-point is seen (line 13). Thus *count* correctly computes the left hand side of condition (9.12) for the current value of $y \in \mathcal{Y}^x$. The rest of the algorithm compares the value in *count* to the right hand side of condition (9.12), which is stored in *threshold*, to compute the prediction set C^x.

If *count* is strictly larger than $\alpha(n+1) - 1$, then by (9.12), $y_i^x \in C^{\text{LOO}}(x)$. If this were not true for y_{i-1}^x (as checked in line 6), then for every $y \in (y_{i-1}^x, y_i^x)$, we have $y \notin C^{\text{LOO}}(x)$. Hence y_i^x is a left end-point for one of the intervals in $C^{\text{LOO}}(x)$. We store this value of y in $left\_endpoint$ until the right end-point is discovered (line 7). Next, if we are at a right end-point, if the current value of *count* is larger than $\alpha(n+1) - 1$, and if *count* is at most $\alpha(n+1) - 1$ after the interval ends (condition on line 10), then $y_i^x \in C^{\text{LOO}}(x)$ and for every $y \in (y_i^x, y_{i+1}^x)$, $y \notin C^{\text{LOO}}(x)$. Thus y_i^x is a right end-point for some interval in C^x, with the left end-point given by the current value of $left\_endpoint$. We update C^x accordingly in line 11. □

9.E.5 Auxiliary lemmas used in 9.E.2

For any matrix $A \in \mathbb{R}^{N \times N}$ and $\alpha \in [0, 1]$, define

$$\mathcal{I}(A) := \left\{ i \in [N] : \sum_{j=1}^{N} \mathbb{1}\{A_{ji} < A_{ij}\} \geqslant (1-\alpha)N \right\}. \tag{9.23}$$

Lemma 9.1 (Section 5.3 of Barber et al. (2021)). *For any matrix A,*

$$\frac{|\mathcal{I}(A)|}{N} \leqslant 2\alpha - \frac{1}{N}.$$

Lemma 9.2 (Section 5.2 of Barber et al. (2021)). *If A is a matrix of random variables such that for any permutation matrix Π, $A \overset{d}{=} \Pi A \Pi^\top$, then for all $1 \leqslant j \leqslant N$,*

$$\mathbb{P}(j \in \mathcal{I}(A)) = \frac{\mathbb{E}[|\mathcal{I}(A)|]}{N} \leqslant 2\alpha - \frac{1}{N}.$$

Remark 9.1. The condition $A \overset{d}{=} \Pi A \Pi^\top$ (for any permutation matrix Π) is equivalent to $(A_{i,j}) \overset{d}{=} (A_{\pi(i),\pi(j)})$ for any permutation $\pi : [N] \to [N]$

Consider the following form of matrices:

$$A_{i,j} = \begin{cases} +\infty, & \text{if } i = j, \\ \mathbb{G}(Z_i, \{Z_1, \ldots, Z_N\} \backslash \{Z_i, Z_j\}), & \text{if } i \neq j, \end{cases} \tag{9.24}$$

for exchangeable random variables $Z_1, \ldots, Z_N$.

Lemma 9.3. *If $Z_1, \ldots, Z_N$ are exchangeable and $\mathbb{G}(\cdot, \cdot)$ treats the elements of its second argument symmetrically, then the matrix A defined by (9.24) satisfies*

$$A \overset{d}{=} \Pi A \Pi^\top,$$

for any permutation matrix Π.

Proof. Observe that for any i, $A_{i,i} = A_{\pi(i),\pi(i)}$ deterministically. For any $i \neq j$, and $\pi(i) = k \neq \pi(j) = \ell$,

$$\begin{aligned} A_{i,j} &:= \mathbb{G}(Z_i, \{Z_1, \ldots, Z_N\} \backslash \{Z_i, Z_j\}), \\ A_{k,\ell} &:= \mathbb{G}(Z_k, \{Z_1, \ldots, Z_N\} \backslash \{Z_k, Z_\ell\}). \end{aligned}$$

Exchangeability of $Z_1, \ldots, Z_N$ implies that for any permutation $\pi : [N] \to [N]$ and any function F that depends symmetrically on $Z_1, \ldots, Z_N$

$$F(Z_i, Z_j) \overset{d}{=} F(Z_{\pi(i)}, Z_{\pi(j)}).$$

The result follows by taking $F(Z_i, Z_j) := \mathbb{G}(Z_i, \{Z_1, \ldots, Z_N\} \backslash \{Z_i, Z_j\})$. □

Theorem 9.6. *If $\mathbb{G}(\cdot, \cdot)$ is a function that treats the elements of its second argument symmetrically, then for any set of exchangeable random variables $Z_1, \ldots, Z_N$, and matrix A defined via (9.24), we have*

$$\mathbb{P}(j \in \mathcal{I}(A)) \leqslant 2\alpha - \frac{1}{N} \quad \textit{for all} \quad j \in [N].$$

Proof. The proof follows by combining Lemmas 9.1, 9.2, 9.3. □

9.E.6 Proofs of results in 9.C

Proof of Theorem 9.4. Conditional on the randomness of M_k, $p_k^{Y_{n+1}}(X_{n+1})$ is a valid p-value and hence conditional on $M_1, \ldots, M_K$, $(2/K) \sum_{k=1}^K p_k^{Y_{n+1}}(X_{n+1})$ is a valid p-value, by Proposition 18 of Vovk and Wang (2018). Therefore, for any $\alpha \in [0, 1]$,

$$\mathbb{P}\left(\frac{1}{K} \sum_{k=1}^{K} p_k^{Y_{n+1}}(X_{n+1}) \leqslant \alpha \right) \leqslant \min\{2, K\}\alpha, \tag{9.25}$$

which implies the result. (The factor 2 follows from (Vovk and Wang, 2018) for $K \geqslant 2$ and for $K = 1$ the factor 2 is not necessary because $p_1^{Y_{n+1}}(X_{n+1})$ is a valid p-value.) □

Proof of Theorem 9.5. Fix $1 \leqslant k \leqslant K$. Conditional on B_k, the scores $r_k(X_i, Y_i), i \in B_k^c, r_k(X_{n+1}, Y_{n+1})$ are exchangeable because $\{(X_i, Y_i) : i \in B_k^c \cup \{n+1\}\}$ are exchangeable by the assumption. Therefore, $p_k^{Y_{n+1}}(X_{n+1})$ is a valid p-value conditional on the bag $(X_i, Y_i), i \in B_k$, where

$$p_k^y(x) := \frac{|\{i \in [n]\backslash B_k : r_k(x,y) \leqslant r_k(X_i, Y_i)\}| + 1}{|[n]\backslash B_k| + 1}.$$

This is similar to the conclusion of Proposition 9.1 where we only need exchangeability of $(X_i, Y_i), i \in \mathcal{I}_2 \cup \{n+1\}$. The result now follows from Proposition 18 of Vovk and Wang (2018). □

9.F Imitating the optimal conditionally-valid prediction set

Prediction sets that are intervals may not be suitable (or well-defined) unless $\mathcal{Y}$ is a totally ordered set. For example, $\mathcal{Y}$ is not totally ordered for classification problems. Furthermore, even if $\mathcal{Y}$ is ordered, it is well known (see introduction of Lei et al. (2013)) that the optimal conditionally-valid prediction regions are level sets of conditional densities (with respect to an appropriate underlying measure), which need not be intervals. Formally, suppose $P_{Y|X}$ has a density $p(y|x)$ with respect to some measure μ on $\mathcal{Y}$. For a given miscoverage level $\alpha \in [0,1]$ we wish to identify an optimal set $C \subseteq \mathcal{Y}$ that satisfies $1-\alpha$ coverage for $Y \mid X = x$, ie

$$\int_{y \in C} p(y|x) dy \geqslant 1 - \alpha.$$

For simplicity, suppose that the conditional density $p(\cdot|x)$ is injective. Then, it is easy to see that the smallest set (with respect to the measure μ) that satisfies coverage $1-\alpha$ must correspond to an upper level set $\{y \in \mathcal{Y} : p(y|x) \geqslant t\}$ for some $t \geqslant 0$. In particular, the appropriate value of t depends on x and α, and is given by

$$t_\alpha(x) := \sup \left\{ t \geqslant 0 : \int_{p(y|x) \geqslant t} p(y|x) dy \geqslant 1 - \alpha \right\}. \tag{9.26}$$

Clearly the set $\{y \in \mathcal{Y} : p(y|x) \geqslant t_\alpha(x)\}$ satisfies $1-\alpha$ coverage and is the smallest set to do so. If an oracle provides us access to $p(y|x)$ for every x, y, we may thus compute the optimal prediction set at level α as

$$C_\alpha^{\texttt{oracle}}(x) \ := \ \{y \in \mathcal{Y} : \ p(y|x) \geqslant t_\alpha(x)\}. \tag{9.27}$$

Note that the prediction regions $\left\{C_\delta^{\texttt{oracle}}(x)\right\}_{\delta \in [0,1]}$ form a sequence of nested sets. This motivates us to imitate/approximate $C_\alpha^{\texttt{oracle}}(x)$ through the nested framework as follows. Let $\widehat{p}(\cdot|x)$ be any estimator of the conditional density, and let $\widehat{g}_x(t)$ be defined as

$$\widehat{g}_x(t) \ := \ \int_{y:\, \widehat{p}(y|x) \geqslant t} \widehat{p}(y|x) dy,$$

which represents the estimated conditional probability of a level set with threshold t. Now, in our effort to mimic (9.26), for any $\delta \in [0,1]$, define $\widehat{t}_\delta(x)$ as the plugin threshold estimator:

$$\widehat{t}_\delta(x) \ := \ \sup\left\{t \geqslant 0 : \ \widehat{g}_x(t) \geqslant 1-\delta\right\}. \tag{9.28}$$

Last, define the nested sets $\{\mathcal{F}_\delta(x)\}_{\delta\in[0,1]}$ as

$$\mathcal{F}_\delta(x) := \{y : \ \widehat{p}(y|x) \geqslant \widehat{t}_\delta(x)\}.$$

It is clear that for any x, the function $\delta \mapsto \widehat{t}_\delta(x)$ is monotonically decreasing and hence for any x, the sets $\{\mathcal{F}_\delta(x)\}_{\delta\in[0,1]}$ are nested. It is also clear that if $\widehat{p}(\cdot|x) = p(\cdot|x)$, then $\mathcal{F}_\alpha(x) = C_\alpha^{\texttt{oracle}}(x)$. Following this, we conjecture that if $\widehat{p}(\cdot|x)$ is consistent for $p(\cdot|x)$ in supremum norm, then $\mu(\mathcal{F}_\alpha(x)\Delta C_\alpha^{\texttt{oracle}}(x)) \to 0$ as $n \to \infty$. (The notation $A\Delta B$ represents symmetric difference.)

An important distinguishing fact about nested sets $\mathcal{F}_\delta(x)$, in comparison with the examples in Section 9.2, is that these are not intervals in general, and are also useful for prediction in the context of classification.

Applying nested split-conformal method from (9.3) to the nested sets above yields the prediction set

$$\widehat{C}_\alpha^{\texttt{oracle}}(x) \ := \ \mathcal{F}_{\widehat{\delta}(\alpha)}(x),$$

where $\widehat{\delta}(\alpha)$ is obtained from the second split $\mathcal{I}_2$ of the data. Unlike the definition in Section 9.2, $\widehat{\delta}(\alpha)$ here is given by the equation

$$1-\widehat{\delta}(\alpha) = \lceil(1-\alpha)(1+1/|\mathcal{I}_2|)\rceil - \text{th quantile of } 1-\delta(X_i,Y_i), i \in \mathcal{I}_2,$$

where $\delta(X_i,Y_i) := \sup\{\delta \in [0,1] : \ \widehat{p}(Y_i|X_i) \geqslant \widehat{t}_\delta(X_i)\}$. This difference is because the nested sets here are decreasing in δ instead of increasing as in Section 9.2. Proposition 9.1 readily yields the validity guarantee

$$\mathbb{P}(Y_{n+1} \in \widehat{C}_\alpha^{\texttt{oracle}}(X_{n+1})) \geqslant 1-\alpha.$$

It is important to realize that the prediction set $\widehat{C}_\alpha^{\texttt{oracle}}(X_{n+1})$ is only marginally valid, although it is motivated through the optimal conditionally valid prediction regions.

Izbicki et al., 2020 propose a prediction set similar to $\widehat{C}_\alpha^{\texttt{oracle}}$ but use an alternative estimate of $t_\delta(x)$ via a notion of "profile distance" between points, which they defined as

$$d(x,x') := \int_0^\infty (\widehat{g}_x(t) - \widehat{g}_{x'}(t))^2 dt.$$

For every x, let $\mathcal{N}_m(x;d)$ represent the indices of the first m nearest neighbors of x with respect to the profile distance d. The prediction set described by Izbicki et al. (2020) is

$$\check{t}_\delta(x) := (1-\delta)\text{-th quantile of } \{\widehat{p}(y_i|x_i) : i \in \mathcal{N}_m(x;d)\}. \tag{9.29}$$

Formulating their procedure in terms of the nested sets

$$\mathcal{F}_\delta^{\text{ISS}}(x) \ := \ \{y \in \mathcal{Y} : \ \widehat{p}(y|x) \geqslant \check{t}_\delta(x)\},$$

Table 9.5: Meta-data for the datasets used in our experiments. N refers to the total number of data-points from which we create 100 versions by independently drawing 1000 data points randomly (as described in the beginning of Section 9.6). d refers to the feature dimension.

Dataset	N	d	**URL (http://archive.ics.uci.edu/ml/datasets/*)**
Blog	52397	280	BlogFeedback
Protein	45730	8	Physicochemical+Properties+of+Protein+Tertiary+Structure
Concrete	1030	8	Concrete+Compressive+Strength
News	39644	59	Online+News+Popularity
Kernel[6]	241600	14	SGEMM+GPU+kernel+performance
Superconductivity	21263	81	Superconductivty+Data

Proposition 9.1 readily yields marginal validity.

We conjecture an improvement over the proposal of Izbicki et al. (2020). Because the optimal prediction set (9.27) depends directly on $t_\delta(x)$, it is more important to combine information from those x_i's which are close to x in terms of $t_\delta(x)$. For example, we may define a "revised" profile distance as

$$\widetilde{d}(x, x') := \int_0^1 (\widehat{t}_\delta(x) - \widehat{t}_\delta(x'))^2 w(\delta) d\delta,$$

where the function $w(\cdot)$ provides more weight on values close to zero. (We use such w because often one is interested in coverage levels close to 1 or equivalently small values of the miscoverage level α.) Using this new distance, we can use various alternative estimators of the threshold, for example using $\widetilde{d}$ in (9.29), or kernel-smoothed variants of (9.28) and (9.29).

In this chapter, we focus on the problem of valid prediction region in the context of regression in which case one often wants to report an interval. For this reason, we leave the discussion of optimal prediction regions discussed herein at this stage, although a more detailed enquiry in this direction would be fruitful for complicated response spaces.

9.G Additional information on experiments

Details for the datasets used in our experiments are provided in Table 9.5. All our experiments were conducted in MATLAB using the TreeBagger class for training regression forests and quantile regression forests. Default parameters were used for all datasets apart from the synthetic dataset of Section 9.6.5.

[6] The GPU kernel dataset contains four output variables corresponding to four measurements of the same entity. The output variable is the average of these values.

Chapter 10

Conclusion and future work

We have shown that ML classifiers can be calibrated, provably and efficiently, using held-out data. Apart from the specific methods and findings reported in the papers/chapters of this thesis, we point out some themes that recur across chapters.

- **Nonparametric binning followed by parametric scaling works well in practice.** While scaling techniques like Platt, beta, temperature, and matrix scaling are popular, we showed in Chapter 3 that they can fail for some distributions. In practice, the best performance is often obtained by first scaling, and then discretizing using fixed-width binning, histogram binning, or isotonic regression. Chapter 6 presents compelling experiments in support of this observation (in the online calibration setting). Further, if the binning step is done on a held-out (independent) batch of data that is identically distributed as the test data, then distribution-free calibration guarantees can be established, such as those in Chapter 4, making them suitable for risk-sensitive domains like medicine.
- **The validation data can be used for more than hyperparameter tuning.** Holding out a small subset of the data (aka the validation data) has proven to be useful for providing good estimates of the unknown test error. The success of post-hoc calibration shows that the traditional train-validate split can actually be repurposed as a train-(validate + calibrate) split. Experiments in Chapters 3, 4, 5, and 6 provide further backing to this observation. While training on validation data makes us prone to overfitting since we can no longer reliably estimate test error, post-hoc calibrating on validation data improves calibration without sacrificing generalization accuracy or other metrics of interest. We expect there to be further interesting ways to utilize the statistical information present in validation data.
- **Calibration as a post-hoc "correction" step.** It is impossible to learn the true data-generating distribution.[1] Thus good forecasts are not perfect but have some desirable properties like small log loss, misclassification error, squared error, or pinball loss. At a

[1] For that matter, even assuming that data comes from a distribution is a useful, but unfalsifiable—and thus unscientific, model of the world. However, notions of calibration can typically be formalized without talking about distributions since they typically boil down to tallying forecasts with actual observed "counts" of an event.

high level, when we calibrate, we retain the good properties of the forecasts but perform a "correction" for something the forecast is not good at. In Chapter 7, we demonstrated this through the concept of parity calibration where we target the up-down event in a regression time-series. We showed that the performance of real-world domain experts can be drastically improved for parity calibration using post-hoc techniques. In Chapter 5, we showed that while calibrating the full multiclass prediction vector is challenging, simpler and useful binary-calibration-like properties can be achieved in multiclass settings.

- **Post-hoc calibration gives adaptivity to distribution drifts and shifts.** In Chapter 3, we discussed post-hoc calibration under covariate shift if certain "importance weights" can be estimated well. In Chapter 6 we demonstrated adaptivity in miscellaneous distribution drift scenarios (including covariate and label drift), in an online supervised learning setting.

We now turn to discuss a few problems in calibration that, to the best of our knowledge, are open and of interest to the community. The question "how do we achieve calibration?" was the focus of this thesis and has been studied extensively elsewhere as well. While there are certainly interesting calibration methods yet to be discovered, the pressing academic question today is, "why calibration?" Answers to this question could revolve around the following considerations.

- **Calibration as a tool for task-specific deployment of ML models.** Often, the downstream user of the ML model is separated from the engineering or research team that built the ML model. Since the scores of a calibrated model exhibit an easy-to-interpret statistical property, it gives the downstream user more control when using the model for miscellaneous tasks. The bullet point "Calibration as a post-hoc correction step…" in the previous list mentioned a related point: calibration can seen as a way of adapting a complex model for specific simpler tasks. These high-level ideas are formalized to some extent using the frameworks of checking rules (Sandroni et al., 2003) and defensive forecasting (Vovk et al., 2005b). However, these frameworks remain theoretical ideas and have not had impact outside a narrow community.
- **Calibration, decision-making, and humans.** If classification scores have no probabilistic meaning, they are uninterpretable for decision-makers, especially if the decision-maker lacks ML expertise. Consider how you would interpret a forecasted chance of rain of 30% or a disease risk score of 5% if these scores are not calibrated. However, calibration is insufficient on its own. Decision-makers are human beings. Their statistical biases and prior views affect the decision-making process and need to be taken into account (Vodrahalli et al., 2022; Benz and Rodriguez, 2023). A weaker property than calibration, called monotonicity has also been considered in this context (Wang et al., 2022). A related question to decision-making is the effect of calibration (or broadly, uncertainty quantification) on trust in the ML model (Yin et al., 2019). HCI (Human-Computer-Interaction) studies on the interplay between calibration, monotonicity, uncertainty quantification, decision-making, and trust are of interest.
- **Calibrated models for robust ML pipelines.** Often, the output of an ML model is not directly used for decision-making, but becomes the input for another model. Such an ML *pipeline* may involve multiple models one after another. Srivastava et al. (2020) give an example of the problem of detecting if a paper receipt is fraudulent or genuine. This could

be done via a three-stage ML pipeline: text localization, then optical character recognition, and then fraud detection. If different models in the pipeline are trained by different entities, it could lead to unexpected behavior (e.g. see Bansal et al. (2019)). Calibrating a model "standardizes" it in a certain sense, and such standardization could enable the model's deployment in practice to be more robust. Theoretical and empirical studies on the role of calibration in ML pipelines would be of significant interest to ML practitioners.

We conclude by emphasizing two key takeaways of this thesis. One, post-hoc calibration is essentially "free" if validation data has already been held out, as is often the case. Two, binning based calibration methods are reliable and almost parameter-free. We recommend post-hoc calibrating as the last step of training a classification model, and hope that this practice becomes commonplace over the years.

Bibliography

[1] Joseph Abbate, Rory Conlin, and Egemen Kolemen. “Data-driven profile prediction for DIII-D”. In: *Nuclear Fusion* 61.4 (2021), p. 046027 (cit. on p. 188).

[2] Joseph Abbate, Rory Conlin, Ricardo Shousha, Keith Erickson, and Egemen Kolemen. “A general infrastructure for data-driven control design and implementation in tokamaks”. In: *Journal of Plasma Physics* 89.1 (2023), p. 895890102 (cit. on p. 187).

[3] Jacob Abernethy, Peter L Bartlett, and Elad Hazan. “Blackwell approachability and no-regret learning are equivalent”. In: *Conference on Learning Theory*. 2011 (cit. on pp. 203, 210, 213).

[4] Alekh Agarwal, Ofer Dekel, and Lin Xiao. “Optimal Algorithms for Online Convex Optimization with Multi-Point Bandit Feedback.” In: *Conference on Learning Theory*. 2010 (cit. on p. 216).

[5] Mohammad Ahsanullah, Valery B Nevzorov, and Mohammad Shakil. *An introduction to order statistics*. Vol. 8. Springer, 2013 (cit. on p. 81).

[6] Amr Alexandari, Anshul Kundaje, and Avanti Shrikumar. “Adapting to Label Shift with Bias-Corrected Calibration”. In: *International Conference on Machine Learning*. 2020 (cit. on p. 41).

[7] Mari-Liis Allikivi and Meelis Kull. “Non-parametric Bayesian isotonic calibration: Fighting over-confidence in binary classification”. In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD*. 2020 (cit. on p. 12).

[8] Barry C Arnold, Narayanaswamy Balakrishnan, and Haikady Navada Nagaraja. *A first course in order statistics*. SIAM, 2008 (cit. on pp. 68, 81, 82).

[9] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. “Tuning Bandit Algorithms in Stochastic Environments”. In: *International Conference on Algorithmic Learning Theory*. 2007 (cit. on pp. 35, 63).

[10] Miriam Ayer, H Daniel Brunk, George M Ewing, William T Reid, and Edward Silverman. “An empirical distribution function for sampling with incomplete information”. In: *The Annals of Mathematical Statistics* (1955), pp. 641–647 (cit. on p. 11).

[11] Yossi Azar, Andrei Z Broder, Anna R Karlin, and Eli Upfal. “Balanced allocations”. In: *Symposium on Theory of Computing*. 1994 (cit. on pp. 206, 207).

[12] Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. *Conformal prediction for reliable machine learning: theory, adaptations and applications.* Newnes, 2014 (cit. on pp. 228, 251, 252).

[13] Gagan Bansal, Besmira Nushi, Ece Kamar, Daniel S Weld, Walter S Lasecki, and Eric Horvitz. "Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff". In: *AAAI Conference on Artificial Intelligence.* 2019 (cit. on p. 268).

[14] Rina Foygel Barber. "Is distribution-free inference possible for binary regression?" In: *Electronic Journal of Statistics* 14.2 (2020), pp. 3487–3524 (cit. on pp. 15, 27, 29, 31).

[15] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, Ryan J Tibshirani, et al. "Predictive inference with the jackknife+". In: *Annals of Statistics* 49.1 (2021), pp. 486–507 (cit. on pp. 24, 230, 232–234, 241, 253, 260).

[16] Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. "The limits of distribution-free conditional predictive inference". In: *Information and Inference: A Journal of the IMA* (Aug. 2020). ISSN: 2049-8772. DOI: 10.1093/imaiai/iaaa017. URL: https://doi.org/10.1093/imaiai/iaaa017 (cit. on p. 228).

[17] Richard E Barlow. *Statistical inference under order restrictions; the theory and application of isotonic regression.* Tech. rep. 1972 (cit. on p. 11).

[18] Richard E Barlow and Hugh D Brunk. "The isotonic regression problem and its dual". In: *Journal of the American Statistical Association* 67.337 (1972), pp. 140–147 (cit. on p. 11).

[19] Nina L Corvelo Benz and Manuel Gomez Rodriguez. "Human-Aligned Calibration for AI-Assisted Decision Making". In: *arXiv preprint arXiv:2306.00074* (2023) (cit. on p. 267).

[20] Steffen Bickel, Michael Brückner, and Tobias Scheffer. "Discriminative Learning for Differing Training and Test Distributions". In: *International Conference on Machine Learning.* 2007 (cit. on p. 39).

[21] Jock A Blackard and Denis J Dean. "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables". In: *Computers and Electronics in Agriculture* 24.3 (1999), pp. 131–151 (cit. on p. 112).

[22] David Blackwell. "An analog of the minimax theorem for vector payoffs." In: *Pacific Journal of Mathematics* 6.1 (1956), pp. 1–8 (cit. on pp. 150, 205, 210, 213).

[23] Henrik Boström, Henrik Linusson, Tuve Löfström, and Ulf Johansson. "Accelerating difficulty estimation for conformal regression forests". In: *Annals of Mathematics and Artificial Intelligence* 81.1-2 (2017), pp. 125–144 (cit. on pp. 232, 237–239, 241).

[24] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32 (cit. on p. 79).

[25] Leo Breiman. *Classification and regression trees.* Routledge, 2017 (cit. on p. 126).

[26] Glenn W Brier. "Verification of forecasts expressed in terms of probability". In: *Monthly Weather Review* 78.1 (1950), pp. 1–3 (cit. on pp. 28, 40, 201).

[27] Jochen Bröcker. "Reliability, sufficiency, and the decomposition of proper scores". In: *Quarterly Journal of the Royal Meteorological Society* 135.643 (2009), pp. 1512–1519 (cit. on pp. 146, 148, 191).

[28] Jochen Bröcker. "Estimating reliability and resolution of probability forecasts through decomposition of the empirical score". In: *Climate dynamics* 39.3-4 (2012), pp. 655–667 (cit. on pp. 40, 71).

[29] Andrei Broder and Michael Mitzenmacher. "Using multiple hash functions to improve IP lookups". In: *IEEE International Conference on Computer Communications*. 2001 (cit. on p. 207).

[30] Peter Bühlmann and Bin Yu. "Analyzing bagging". In: *The Annals of Statistics* 30.4 (2002), pp. 927–961 (cit. on p. 255).

[31] Lars Carlsson, Martin Eklund, and Ulf Norinder. "Aggregated conformal prediction". In: *International Conference on Artificial Intelligence Applications and Innovations*. 2014 (cit. on pp. 232, 254, 256).

[32] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006 (cit. on pp. 204, 219).

[33] Ian Char, Youngseog Chung, Mark Boyer, Egemen Kolemen, and Jeff Schneider. "A Model-Based Reinforcement Learning Approach for Beta Control". In: *APS Division of Plasma Physics Meeting Abstracts*. Vol. 2021. 2021, pp. 11–150 (cit. on p. 188).

[34] Bertrand Charpentier, Oliver Borchert, Daniel Zügner, Simon Geisler, and Stephan Günnemann. "Natural Posterior Network: Deep Bayesian Predictive Uncertainty for Exponential Family Distributions". In: *International Conference on Learning Representations*. 2022 (cit. on p. 175).

[35] Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. "Distributional conformal prediction". In: *Proceedings of the National Academy of Sciences* 118.48 (2021) (cit. on pp. 232, 244).

[36] Youngseog Chung, Ian Char, Han Guo, Jeff Schneider, and Willie Neiswanger. "Uncertainty toolbox: an open-source library for assessing, visualizing, and improving uncertainty quantification". In: *arXiv preprint arXiv:2109.10254* (2021) (cit. on pp. 180, 190).

[37] Youngseog Chung, Willie Neiswanger, Ian Char, and Jeff Schneider. "Beyond pinball loss: Quantile methods for calibrated uncertainty quantification". In: *Advances in Neural Information Processing Systems* (2021) (cit. on p. 175).

[38] Youngseog Chung, Aaron Rumack, and Chirag Gupta. "Parity calibration". In: *Conference on Uncertainty in Artificial Intelligence*. 2023 (cit. on pp. 7, 173).

[39] Charles J Clopper and Egon S Pearson. "The use of confidence or fiducial limits illustrated in the case of the binomial". In: *Biometrika* 26.4 (1934), pp. 404–413 (cit. on p. 89).

[40] Richard Cole, Bruce M Maggs, Friedhelm Meyer auf der Heide, Michael Mitzenmacher, Andréa W Richa, Klaus Schröder, Ramesh K Sitaraman, and Berthold Vöcking. "Randomized protocols for low-congestion circuit routing in multistage interconnection networks". In: *Symposium on Theory of Computing*. 1998 (cit. on p. 207).

[41] Gert de Cooman and Jasper De Bock. "Randomness is inherently imprecise". In: *International Journal of Approximate Reasoning* 141 (2022), pp. 28–68 (cit. on p. 205).

[42] Thomas M Cover. "Universal portfolios". In: *Mathematical finance* 1.1 (1991), pp. 1–29 (cit. on p. 4).

[43] Estee Y Cramer, Yuxin Huang, Yijin Wang, Evan L Ray, Matthew Cornell, Johannes Bracher, Andrea Brennen, Alvaro J Castro Rivadeneira, Aaron Gerding, Katie House, Dasuni Jayawardena, Abdul H Kanji, Ayush Khandelwal, Khoa Le, Jarad Niemi, Ariane Stark, Apurv Shah, Nutcha Wattanachit, Martha W Zorn, Nicholas G Reich, and US

COVID-19 Forecast Hub Consortium. “The United States COVID-19 Forecast Hub dataset”. In: *medRxiv* (2021). DOI: 10.1101/2021.11.04.21265886 (cit. on p. 181).

[44] Estee Y. Cramer, Evan L. Ray, Velma K. Lopez, et al., and Nicholas G. Reich. “Evaluation of individual and ensemble probabilistic forecasts of COVID-19 mortality in the United States”. In: *Proceedings of the National Academy of Sciences* 119.15 (2022), e2113561119. DOI: 10.1073/pnas.2113561119 (cit. on p. 181).

[45] Peng Cui, Wenbo Hu, and Jun Zhu. “Calibrated Reliable Regression using Maximum Mean Discrepancy”. In: *Advances in Neural Information Processing Systems.* 2020 (cit. on p. 175).

[46] Ran Dai, Hyebin Song, Rina Foygel Barber, and Garvesh Raskutti. “The bias of isotonic regression”. In: *Electronic Journal of Statistics* 14.1 (2020), p. 801 (cit. on p. 67).

[47] A Philip Dawid. “The well-calibrated Bayesian”. In: *Journal of the American Statistical Association* 77.379 (1982), pp. 605–610 (cit. on pp. 3, 6, 28, 40, 64, 139, 177, 202).

[48] A Philip Dawid. “Comment: The Impossibility of Inductive Inference”. In: *Journal of the American Statistical Association* 80.390 (1985), pp. 340–341 (cit. on pp. 6, 147, 202).

[49] A Philip Dawid. “Probability forecasting”. In: *Wiley StatsRef: Statistics Reference Online* (2014) (cit. on p. 40).

[50] A Philip Dawid and Vladimir Vovk. “Prequential probability: principles and properties”. In: *Bernoulli* (1999), pp. 125–162 (cit. on pp. 16, 19).

[51] Morris H DeGroot and Stephen E Fienberg. *Assessing Probability Assessors: Calibration and Refinement.* Tech. rep. Carnegie Mellon University, 1981 (cit. on pp. 140, 177, 180, 190).

[52] Morris H DeGroot and Stephen E Fienberg. “The comparison and evaluation of forecasters”. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 32.1-2 (1983), pp. 12–22 (cit. on pp. 17, 40).

[53] Luc Devroye et al. “The equivalence of weak, strong and complete convergence in L_1 for kernel density estimates”. In: *The Annals of Statistics* 11.3 (1983), pp. 896–904 (cit. on p. 125).

[54] Luc Devroye. “Automatic pattern recognition: A study of the probability of error”. In: *IEEE Transactions on pattern analysis and machine intelligence* 10.4 (1988), pp. 530–543 (cit. on p. 126).

[55] Raaz Dwivedi, Ohad N Feldheim, Ori Gurel-Gurevich, and Aaditya Ramdas. “The power of online thinning in reducing discrepancy”. In: *Probability Theory and Related Fields* 174.1 (2019), pp. 103–131 (cit. on p. 207).

[56] M. P. Ershov. “Extension of Measures and Stochastic Equations”. In: *Theory of Probability & Its Applications* 19.3 (1975), pp. 431–444 (cit. on pp. 33, 48, 49).

[57] Meir Feder, Neri Merhav, and Michael Gutman. “Universal prediction of individual sequences”. In: *IEEE transactions on Information Theory* 38.4 (1992), pp. 1258–1270 (cit. on p. 4).

[58] Christopher AT Ferro and Thomas E Fricker. “A bias-corrected decomposition of the Brier score”. In: *Quarterly Journal of the Royal Meteorological Society* 138.668 (2012), pp. 1954–1960 (cit. on p. 40).

[59] Dean P Foster. “A proof of calibration via Blackwell’s approachability theorem”. In: *Games and Economic Behavior* 29.1-2 (1999), pp. 73–78 (cit. on pp. 6, 7, 149, 150, 164–166, 171, 172, 203, 204, 208, 210, 217, 219).

[60] Dean P Foster and Sergiu Hart. “Smooth calibration, leaky forecasts, finite recall, and nash dynamics”. In: *Games and Economic Behavior* 109 (2018), pp. 271–293 (cit. on p. 204).

[61] Dean P Foster and Sergiu Hart. “Forecast Hedging and Calibration”. In: *Journal of Political Economy* 129.12 (2021), pp. 3447–3490 (cit. on pp. 141, 204).

[62] Dean P Foster and Sergiu Hart. ““Calibeating”: beating forecasters at their own game”. In: *Theoretical Economics (to appear)* (2023) (cit. on pp. 22, 141, 147–150, 168).

[63] Dean P Foster and Rakesh Vohra. “Asymptotic calibration”. In: *Biometrika* 85.2 (June 1998), pp. 379–390. ISSN: 0006-3444 (cit. on pp. 3, 6, 140, 147, 149, 150, 165, 177, 203, 215).

[64] Dylan J Foster, Satyen Kale, Haipeng Luo, Mehryar Mohri, and Karthik Sridharan. “Logistic regression: The importance of being improper”. In: *Conference On Learning Theory.* 2018 (cit. on pp. 142, 145, 146, 166).

[65] Yoav Freund and Robert E Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139 (cit. on p. 4).

[66] Drew Fudenberg and David K Levine. “An easier way to calibrate”. In: *Games and Economic Behavior* 29.1-2 (1999), pp. 131–137 (cit. on pp. 3, 203).

[67] Saurabh Garg, Yifan Wu, Sivaraman Balakrishnan, and Zachary C Lipton. “A Unified View of Label Shift Estimation”. In: *Advances in Neural Information Processing Systems.* 2020 (cit. on p. 41).

[68] Isaac Gibbs and Emmanuel Candes. “Adaptive conformal inference under distribution shift”. In: *Advances in Neural Information Processing Systems.* 2021 (cit. on p. 24).

[69] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. “Probabilistic forecasts, calibration and sharpness”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.2 (2007), pp. 243–268 (cit. on pp. 65, 106, 148, 173, 174).

[70] Tilmann Gneiting and Adrian E Raftery. “Strictly proper scoring rules, prediction, and estimation”. In: *Journal of the American statistical Association* 102.477 (2007), pp. 359–378 (cit. on p. 146).

[71] Louis Gordon and Richard A Olshen. “Almost surely consistent nonparametric regression from recursive partitioning schemes”. In: *Journal of Multivariate Analysis* 15.2 (1984), pp. 147–163 (cit. on p. 126).

[72] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. “On Calibration of Modern Neural Networks”. In: *International Conference on Machine Learning.* 2017 (cit. on pp. 3, 4, 12–14, 18, 19, 28, 29, 31, 34, 40, 65, 67, 68, 79, 94, 96–98, 102, 104, 115, 119, 120, 139, 140, 155, 178).

[73] Chirag Gupta, Arun K Kuchibhotla, and Aaditya Ramdas. “Nested conformal prediction and quantile out-of-bag ensemble methods”. In: *Pattern Recognition* 127 (2022), p. 108496 (cit. on pp. 31, 227).

[74] Chirag Gupta, Aleksandr Podkopaev, and Aaditya Ramdas. “Distribution-free binary classification: prediction sets, confidence intervals and calibration”. In: *Advances in*

Neural Information Processing Systems. 2020 (cit. on pp. 3, 8, 9, 15, 26, 30, 65–69, 73, 76, 79, 86, 104, 111, 140, 147, 178, 206).

[75] Chirag Gupta and Aaditya Ramdas. "Distribution-free calibration guarantees for histogram binning without sample splitting". In: *International Conference on Machine Learning*. 2021 (cit. on pp. 10, 11, 18, 64, 94, 104, 105, 109, 111–113, 115, 122, 126, 131–133, 135, 136, 140, 158, 178, 206).

[76] Chirag Gupta and Aaditya Ramdas. "Faster online calibration without randomization: interval forecasts and the power of two choices". In: *Conference On Learning Theory*. 2022 (cit. on pp. 164, 165, 171, 201).

[77] Chirag Gupta and Aaditya Ramdas. "Top-label calibration and multiclass-to-binary reductions". In: *International Conference on Learning Representations*. 2022 (cit. on pp. 12, 14, 93, 206).

[78] Chirag Gupta and Aaditya Ramdas. "Online Platt Scaling with calibeating". In: *International Conference on Machine Learning*. 2023 (cit. on pp. 7, 139, 179, 199).

[79] Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. "Calibration of neural networks using splines". In: *International Conference on Learning Representations*. 2021 (cit. on pp. 12, 100, 104, 114).

[80] Sergiu Hart and Andreu Mas-Colell. "A simple adaptive procedure leading to correlated equilibrium". In: *Econometrica* 68.5 (2000), pp. 1127–1150 (cit. on p. 210).

[81] Elad Hazan. "Introduction to online convex optimization". In: *Foundations and Trends in Optimization* 2.3-4 (2016), pp. 157–325 (cit. on pp. 146, 162, 166).

[82] Elad Hazan, Amit Agarwal, and Satyen Kale. "Logarithmic regret algorithms for online convex optimization". In: *Machine Learning* 69.2 (2007), pp. 169–192 (cit. on pp. 142, 146).

[83] Elad Hazan and Comandur Seshadhri. "Efficient learning algorithms for changing environments". In: *International Conference on Machine Learning*. 2009 (cit. on p. 155).

[84] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016 (cit. on p. 97).

[85] Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. "Multicalibration: Calibration for the (computationally-identifiable) masses". In: *International Conference on Machine Learning*. 2018 (cit. on p. 79).

[86] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. "Deep Anomaly Detection with Outlier Exposure". In: *International Conference on Learning Representations*. 2019 (cit. on p. 41).

[87] Keisuke Hirano and Jack R Porter. "Impossibility results for nondifferentiable functionals". In: *Econometrica* 80.4 (2012), pp. 1769–1790 (cit. on p. 15).

[88] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780 (cit. on pp. 185, 192).

[89] Steven R Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. "Time-uniform, nonparametric, nonasymptotic confidence sequences". In: *The Annals of Statistics* 49.2 (2021), pp. 1055–1080 (cit. on pp. 37, 51, 62, 223).

[90] Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. "Time-uniform Chernoff bounds via nonnegative supermartingales". In: *Probability Surveys* 17 (2020), pp. 257–317 (cit. on pp. 37, 51).

[91] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. "Densely connected convolutional networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017 (cit. on p. 104).

[92] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex J. Smola. "Correcting Sample Selection Bias by Unlabeled Data". In: *Advances in Neural Information Processing Systems*. 2007 (cit. on p. 39).

[93] David Humphreys, G Ambrosino, Peter de Vries, Federico Felici, Sun H Kim, Gary Jackson, A Kallenbach, Egemen Kolemen, J Lister, D Moreau, et al. "Novel aspects of plasma control in ITER". In: *Physics of Plasmas* 22.2 (2015), p. 021806 (cit. on p. 187).

[94] Rafael Izbicki, Gilson Shimizu, and Rafael Stern. "Flexible distribution-free conditional predictive bands using density estimators". In: *International Conference on Artificial Intelligence and Statistics*. 2020 (cit. on pp. 232, 263, 264).

[95] Rémi Jézéquel, Pierre Gaillard, and Alessandro Rudi. "Efficient improper learning for online logistic regression". In: *Conference on Learning Theory*. 2020 (cit. on pp. 142, 145, 146, 166).

[96] Rémi Jézéquel, Pierre Gaillard, and Alessandro Rudi. "Mixability made efficient: Fast online multiclass logistic regression". In: *Advances in Neural Information Processing Systems*. 2021 (cit. on p. 155).

[97] Ulf Johansson, Henrik Boström, Tuve Löfström, and Henrik Linusson. "Regression conformal prediction with random forests". In: *Machine Learning* 97.1-2 (2014), pp. 155–176 (cit. on pp. 24, 230, 232, 237, 241).

[98] Sham M Kakade and Dean P Foster. "Deterministic calibration and Nash equilibrium". In: *Conference on Learning Theory*. 2004 (cit. on p. 204).

[99] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. "A Least-Squares Approach to Direct Importance Estimation". In: *Journal of Machine Learning Research* 10 (2009), pp. 1391–1445 (cit. on pp. 39, 58).

[100] Alex Kendall and Yarin Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems*. 2017 (cit. on p. 40).

[101] Byol Kim, Chen Xu, and Rina Barber. "Predictive inference is free with the jackknife+-after-bootstrap". In: *Advances in Neural Information Processing Systems*. 2020 (cit. on pp. 230, 232, 237, 238, 241).

[102] Danijel Kivaranovic, Kory D Johnson, and Hannes Leeb. "Adaptive, Distribution-Free Prediction Intervals for Deep Networks". In: *International Conference on Artificial Intelligence and Statistics*. 2020 (cit. on p. 232).

[103] Alex Krizhevsky. "Learning multiple layers of features from tiny images". In: *Technical Report, University of Toronto* (2009) (cit. on p. 97).

[104] Volodymyr Kuleshov and Stefano Ermon. "Estimating uncertainty online against an adversary". In: *AAAI Conference on Artificial Intelligence*. 2017 (cit. on pp. 141, 150).

[105] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. “Accurate Uncertainties for Deep Learning Using Calibrated Regression”. In: *International Conference on Machine Learning*. 2018 (cit. on pp. 40, 106, 175).

[106] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. “Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration”. In: *Advances in neural information processing systems*. 2019 (cit. on pp. 13, 14, 96, 102, 104, 115, 119, 122).

[107] Meelis Kull, Telmo M. Silva Filho, and Peter Flach. “Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration”. In: *Electronic Journal of Statistics* 11.2 (2017), pp. 5052–5080 (cit. on pp. 9, 12, 34, 79, 94, 100, 153, 162, 163, 178).

[108] Ananya Kumar, Percy S Liang, and Tengyu Ma. “Verified Uncertainty Calibration”. In: *Advances in Neural Information Processing Systems*. 2019 (cit. on pp. 14, 17, 37, 40, 41, 63, 67, 68, 71, 76, 86, 96, 120, 140, 150, 155, 157).

[109] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. “Trainable Calibration Measures for Neural Networks from Kernel Mean Embeddings”. In: *International Conference on Machine Learning*. 2018 (cit. on pp. 40, 96).

[110] Lars van der Laan, Ernesto Ulloa-Pérez, Marco Carone, and Alex Luedtke. “Causal isotonic calibration for heterogeneous treatment effects”. In: *International Conference on Machine Learning*. 2023 (cit. on p. 12).

[111] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *Advances in Neural Information Processing Systems*. 2017 (cit. on pp. 40, 79, 178, 193).

[112] Antonis Lambrou, Ilia Nouretdinov, and Harris Papadopoulos. “Inductive Venn prediction”. In: *Annals of Mathematics and Artificial Intelligence* 74.1-2 (2015), pp. 181–201 (cit. on pp. 40, 61).

[113] Donghwan Lee, Xinmeng Huang, Hamed Hassani, and Edgar Dobriban. “T-cal: An optimal test for the calibration of predictive models”. In: *arXiv preprint arXiv:2203.01850* (2022) (cit. on p. 19).

[114] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. “Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples”. In: *International Conference on Learning Representations*. 2018 (cit. on p. 41).

[115] Yonghoon Lee and Rina Barber. “Distribution-free inference for regression: discrete, continuous, and in between”. In: *Advances in Neural Information Processing Systems*. 2021 (cit. on p. 15).

[116] Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006 (cit. on p. 184).

[117] Ehud Lehrer. “Any inspection is manipulable”. In: *Econometrica* 69.5 (2001), pp. 1333–1347 (cit. on p. 204).

[118] Jing Lei, Max G’Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. “Distribution-free predictive inference for regression”. In: *Journal of the American Statistical Association* 113.523 (2018), pp. 1094–1111 (cit. on pp. 228–232, 241, 254, 255).

[119] Jing Lei, James Robins, and Larry Wasserman. "Distribution-free prediction sets". In: *Journal of the American Statistical Association* 108.501 (2013), pp. 278–287 (cit. on pp. 229, 262).

[120] Henrik Linusson, Ulf Johansson, and Henrik Boström. "Efficient conformal predictor ensembles". In: *Neurocomputing* (2019) (cit. on pp. 237, 254, 256).

[121] Henrik Linusson, Ulf Norinder, Henrik Boström, Ulf Johansson, and Tuve Löfström. "On the calibration of aggregated conformal predictors". In: *Symposium on Conformal and Probabilistic Prediction with Applications (COPA)*. 2017 (cit. on p. 254).

[122] Gábor Lugosi and Andrew Nobel. "Consistency of data-driven histogram methods for density estimation and classification". In: *Annals of Statistics* 24.2 (1996), pp. 687–706 (cit. on pp. 67, 126).

[123] James L Luxon. "A design retrospective of the DIII-D tokamak". In: *Nuclear Fusion* 42.5 (2002), p. 614 (cit. on pp. 188, 195).

[124] Shie Mannor and Vianney Perchet. "Approachability, fast and slow". In: *Conference on Learning Theory*. 2013 (cit. on pp. 203, 211, 213, 214).

[125] Shie Mannor and Gilles Stoltz. "A geometric proof of calibration". In: *Mathematics of Operations Research* 35.4 (2010), pp. 721–727 (cit. on pp. 203, 204).

[126] Charles Marx, Shengjia Zhao, Willie Neiswanger, and Stefano Ermon. "Modular conformal calibration". In: *International Conference on Machine Learning*. 2022 (cit. on p. 175).

[127] Jena Weather Station at Max Planck Institute for Biogeochemistry. "Jena Climate Data". In: (2016). URL: https://www.bgc-jena.mpg.de/wetter/ (cit. on p. 184).

[128] Daniel McFadden. "Conditional logit analysis of qualitative choice behavior". In: *Frontiers in Econometrics* (1974), pp. 105–142 (cit. on p. 201).

[129] Nicolai Meinshausen. "Quantile regression forests". In: *Journal of Machine Learning Research* 7.Jun (2006), pp. 983–999 (cit. on p. 239).

[130] Dimitrios Milios, Raffaello Camoriano, Pietro Michiardi, Lorenzo Rosasco, and Maurizio Filippone. "Dirichlet-based Gaussian processes for large-scale calibrated classification". In: *Advances in Neural Information Processing Systems*. 2018 (cit. on p. 40).

[131] Robert G Miller. "Statistical prediction by discriminant analysis". In: *Statistical Prediction by Discriminant Analysis*. Springer, 1962, pp. 1–54 (cit. on p. 67).

[132] Michael Mitzenmacher. "The power of two choices in randomized load balancing". PhD thesis. University of California, Berkeley, 1996 (cit. on p. 207).

[133] Michael Mitzenmacher, Andréa W Richa, and Ramesh Sitaraman. "The power of two random choices: A survey of techniques and results". In: *Combinatorial Optimization* 9 (2001), pp. 255–304 (cit. on p. 207).

[134] Edward Morse. *Nuclear Fusion*. Springer, 2018 (cit. on p. 187).

[135] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip HS Torr, and Puneet K Dokania. "Calibrating deep neural networks using focal loss". In: *Advances in Neural Information Processing Systems*. 2020 (cit. on pp. 96, 105, 106, 119).

[136] Allan H Murphy. "Scalar and vector partitions of the probability score: Part I. Two-state situation". In: *Journal of Applied Meteorology* 11.2 (1972), pp. 273–282 (cit. on p. 40).

[137] Allan H Murphy. “Scalar and vector partitions of the probability score: Part II. N-state situation”. In: *Journal of Applied Meteorology* 11.8 (1972), pp. 1183–1192 (cit. on p. 40).

[138] Allan H Murphy. “A new vector partition of the probability score”. In: *Journal of applied Meteorology* 12.4 (1973), pp. 595–600 (cit. on pp. 40, 148).

[139] Allan H Murphy and Edward S Epstein. “Verification of probabilistic predictions: A brief review”. In: *Journal of Applied Meteorology* 6.5 (1967), pp. 748–755 (cit. on pp. 28, 40).

[140] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. “Obtaining well calibrated probabilities using Bayesian binning”. In: *AAAI Conference on Artificial Intelligence*. 2015 (cit. on pp. 11, 17, 40, 67, 70, 79, 178).

[141] Mahdi Pakdaman Naeini and Gregory F Cooper. “Binary classifier calibration using an ensemble of near isotonic regression models”. In: *International Conference on Data Mining*. IEEE. 2016 (cit. on p. 12).

[142] Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. “Binary classifier calibration: Non-parametric approach”. In: *arXiv preprint arXiv:1401.3390* (2014) (cit. on pp. 12, 67).

[143] Gergely Neu and Nikita Zhivotovskiy. “Fast rates for online prediction with abstention”. In: *Conference on Learning Theory*. 2020 (cit. on p. 216).

[144] Alexandru Niculescu-Mizil and Rich Caruana. “Predicting Good Probabilities with Supervised Learning”. In: *International Conference on Machine Learning*. 2005 (cit. on pp. 4, 40, 67, 71, 89, 97, 122, 139, 140, 180, 190).

[145] David A Nix and Andreas S Weigend. “Estimating the mean and variance of the target probability distribution”. In: *International Conference on Neural Networks*. 1994 (cit. on pp. 178, 193).

[146] Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. “Measuring Calibration in Deep Learning.” In: *arXiv preprint arXiv:1904.01685* (2020) (cit. on pp. 17, 96).

[147] Andrew Nobel. “Histogram regression estimation using data-dependent partitions”. In: *The Annals of Statistics* 24.3 (1996), pp. 1084–1105 (cit. on p. 126).

[148] David Oakes. “Self-calibrating priors do not exist”. In: *Journal of the American Statistical Association* 80.390 (1985), pp. 339–339 (cit. on pp. 6, 147, 202).

[149] Harris Papadopoulos, Kostas Proedrou, Vladimir Vovk, and Alex Gammerman. “Inductive confidence machines for regression”. In: *European Conference on Machine Learning*. 2002 (cit. on pp. 24, 31, 228–231, 241).

[150] Sangdon Park, Osbert Bastani, James Weimer, and Insup Lee. “Calibrated Prediction with Covariate Shift via Unsupervised Domain Adaptation”. In: *International Conference on Artificial Intelligence and Statistics*. 2020 (cit. on p. 40).

[151] KR Parthasarathy and PK Bhattacharya. “Some limit theorems in regression theory”. In: *Sankhyā: The Indian Journal of Statistics, Series A* (1961), pp. 91–102 (cit. on p. 67).

[152] Kanil Patel, William Beluch, Bin Yang, Michael Pfeiffer, and Dan Zhang. “Multi-class uncertainty calibration via mutual information maximization-based binning”. In: *International Conference on Learning Representations*. 2021 (cit. on pp. 15, 103, 104).

[153] Vianney Perchet. “Approachability, regret and calibration: Implications and equivalences”. In: *Journal of Dynamics & Games* 1.2 (2014), p. 181 (cit. on pp. 150, 171).

[154] Vianney Perchet. “Exponential weight approachability, applications to calibration and regret minimization”. In: *Dynamic Games and Applications* 5.1 (2015), pp. 136–153 (cit. on pp. 203, 204).

[155] John C. Platt. “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods”. In: *Advances in Large Margin Classifiers.* MIT Press, 1999, pp. 61–74 (cit. on pp. 3, 4, 8, 9, 31, 34, 40, 67, 101, 122, 140, 178).

[156] Aleksandr Podkopaev and Aaditya Ramdas. “Distribution-free uncertainty quantification for classification under label shift”. In: *Conference on Uncertainty in Artificial Intelligence.* 2021 (cit. on p. 125).

[157] Teodora Popordanoska, Raphael Sayer, and Matthew Blaschko. “A consistent and differentiable lp canonical calibration error estimator”. In: *Advances in Neural Information Processing Systems.* 2022 (cit. on p. 19).

[158] Foster Provost and Pedro Domingos. “Tree induction for probability-based ranking”. In: *Machine learning* 52.3 (2003), pp. 199–215 (cit. on pp. 4, 79).

[159] Jian Qian, Ronan Fruit, Matteo Pirotta, and Alessandro Lazaric. “Concentration Inequalities for Multinoulli Random Variables”. In: *arXiv preprint arXiv:2001.11595* (2020) (cit. on p. 125).

[160] Mingda Qiao and Gregory Valiant. “Stronger calibration lower bounds via sidestepping”. In: *Symposium on Theory of Computing.* 2021 (cit. on pp. 203, 215).

[161] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning.* MIT Press, 2008 (cit. on p. 22).

[162] Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Richard Hartley, and Byron Boots. “Intra order-preserving functions for calibration of multi-class neural networks”. In: *Advances in Neural Information Processing Systems.* 2020 (cit. on pp. 14, 104).

[163] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. “Online learning: Beyond regret”. In: *Conference on Learning Theory.* 2011 (cit. on p. 204).

[164] Carl Edward Rasmussen. “Gaussian processes in machine learning”. In: *ML Summer School 2003 (Canberra, Australia).* Springer. 2004 (cit. on p. 178).

[165] Rebecca Roelofs, Nicholas Cain, Jonathon Shlens, and Michael C Mozer. “Mitigating bias in calibration error estimation”. In: *International Conference on Artificial Intelligence and Statistics.* 2022 (cit. on pp. 67, 68, 98, 104, 121, 150, 157).

[166] Yaniv Romano, Evan Patterson, and Emmanuel Candes. “Conformalized quantile regression”. In: *Advances in Neural Information Processing Systems.* 2019 (cit. on pp. 230–232, 238, 239, 241, 243, 247, 248, 256–258).

[167] Marco Saerens, Patrice Latinne, and Christine Decaestecker. “Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure”. In: *Neural computation* 14.1 (2002), pp. 21–41 (cit. on p. 41).

[168] Roshni Sahoo, Shengjia Zhao, Alyssa Chen, and Stefano Ermon. “Reliable decisions with threshold calibration”. In: *Advances in Neural Information Processing Systems.* 2021 (cit. on pp. 175, 176).

[169] Frederick Sanders. "On subjective probability forecasting". In: *Journal of Applied Meteorology* 2.2 (1963), pp. 191–201 (cit. on pp. 28, 40, 67).

[170] Alvaro Sandroni, Rann Smorodinsky, and Rakesh V Vohra. "Calibration with many checking rules". In: *Mathematics of Operations Research* 28.1 (2003), pp. 141–153 (cit. on pp. 204, 267).

[171] Jaemin Seo, Y-S Na, B Kim, CY Lee, MS Park, SJ Park, and YH Lee. "Feedforward beta control in the KSTAR tokamak by deep reinforcement learning". In: *Nuclear Fusion* 61.10 (2021), p. 106010 (cit. on p. 188).

[172] Seonguk Seo, Paul Hongsuck Seo, and Bohyung Han. "Learning for single-shot confidence calibration in deep neural networks through stochastic inferences". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019 (cit. on p. 40).

[173] Matteo Sesia and Emmanuel J Candès. "A comparison of some conformal quantile regression methods". In: *Stat* 9.1 (2020), e261 (cit. on pp. 232, 239, 241, 243).

[174] Glenn Shafer and Vladimir Vovk. *Game-Theoretic Foundations for Probability and Finance*. John Wiley & Sons, 2019 (cit. on p. 205).

[175] Ohad Shamir. "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 1703–1713 (cit. on p. 216).

[176] Hidetoshi Shimodaira. "Improving predictive inference under covariate shift by weighting the log-likelihood function". In: *Journal of Statistical Planning and Inference* 90.2 (2000), pp. 227–244 (cit. on p. 38).

[177] Hao Song, Tom Diethe, Meelis Kull, and Peter Flach. "Distribution calibration for regression". In: *International Conference on Machine Learning*. 2019 (cit. on p. 175).

[178] David J Spiegelhalter. "Probabilistic prediction in patient management and clinical trials". In: *Statistics in Medicine* 5.5 (1986), pp. 421–433 (cit. on p. 201).

[179] Megha Srivastava, Besmira Nushi, Ece Kamar, Shital Shah, and Eric Horvitz. "An empirical analysis of backward compatibility in machine learning systems". In: *International Conference on Knowledge Discovery & Data Mining*. 2020 (cit. on p. 267).

[180] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. "Conformal Prediction Under Covariate Shift". In: *Advances in Neural Information Processing Systems*. 2019 (cit. on p. 40).

[181] Ryan J Tibshirani, Holger Hoefling, and Robert Tibshirani. "Nearly-isotonic regression". In: *Technometrics* 53.1 (2011), pp. 54–61 (cit. on p. 12).

[182] Gia-Lac Tran, Edwin V Bonilla, John Cunningham, Pietro Michiardi, and Maurizio Filippone. "Calibrating Deep Convolutional Gaussian Processes". In: *International Conference on Artificial Intelligence and Statistics*. 2019 (cit. on p. 40).

[183] Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas B Schön. "Evaluating model calibration in classification". In: *International Conference on Artificial Intelligence and Statistics*. 2019 (cit. on pp. 12, 28, 40, 43, 93, 96, 125).

[184] Kaspar Valk and Meelis Kull. "Assuming Locally Equal Calibration Errors for Non-Parametric Multiclass Calibration". In: *Transactions on Machine Learning Research* (2023) (cit. on p. 11).

[185] Kailas Vodrahalli, Tobias Gerstenberg, and James Y Zou. "Uncalibrated models can improve human-ai collaboration". In: *Advances in Neural Information Processing Systems*. 2022 (cit. on p. 267).

[186] Vladimir Vovk. "Aggregating strategies". In: *Conference on Computational Learning Theory*. 1990 (cit. on p. 146).

[187] Vladimir Vovk. "A game of prediction with expert advice". In: *Conference on Computational Learning Theory*. 1995 (cit. on p. 4).

[188] Vladimir Vovk. "Cross-conformal predictors". In: *Annals of Mathematics and Artificial Intelligence* 74.1-2 (2015), pp. 9–28 (cit. on pp. 24, 230, 232, 233, 241, 254–256).

[189] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005 (cit. on pp. 27, 29, 31, 40, 61, 228).

[190] Vladimir Vovk and Ivan Petej. "Venn-Abers predictors". In: *Conference on Uncertainty in Artificial Intelligence*. 2014 (cit. on pp. 28, 40, 61, 62, 206).

[191] Vladimir Vovk, Ivan Petej, and Valentina Fedorova. "Large-scale probabilistic predictors with and without guarantees of validity". In: *Advances in Neural Information Processing Systems*. 2015 (cit. on p. 40).

[192] Vladimir Vovk, Glenn Shafer, and Ilia Nouretdinov. "Self-calibrating probability forecasting". In: *Advances in Neural Information Processing Systems*. 2003 (cit. on pp. 40, 61, 206).

[193] Vladimir Vovk, Akimichi Takemura, and Glenn Shafer. "Defensive forecasting". In: *International Workshop on Artificial Intelligence and Statistics*. 2005 (cit. on pp. 203, 204, 267).

[194] Vladimir Vovk and Ruodu Wang. "Combining p-values via averaging". In: *Forthcoming, Biometrika* (2018) (cit. on pp. 261, 262).

[195] Peter Walley and Terrence L Fine. "Towards a frequentist theory of upper and lower probability". In: *The Annals of Statistics* 10.3 (1982), pp. 741–761 (cit. on p. 205).

[196] Lequn Wang, Thorsten Joachims, and Manuel Gomez Rodriguez. "Improving screening processes via calibrated subset selection". In: *International Conference on Machine Learning*. 2022 (cit. on p. 267).

[197] Antoine Wehenkel and Gilles Louppe. "Unconstrained monotonic neural networks". In: *Advances in Neural Information Processing Systems*. 2019 (cit. on p. 15).

[198] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. "Inequalities for the L1 deviation of the empirical distribution". In: *Hewlett-Packard Labs, Tech. Rep* (2003) (cit. on p. 125).

[199] Jonathan Wenger, Hedvig Kjellström, and Rudolph Triebel. "Non-Parametric Calibration for Classification". In: *International Conference on Artificial Intelligence and Statistics*. 2020 (cit. on pp. 15, 40).

[200] David Widmann, Fredrik Lindsten, and Dave Zachariah. “Calibration tests in multi-class classification: a unifying framework”. In: *Advances in Neural Information Processing Systems*. 2019 (cit. on pp. 17, 19, 40, 71, 128, 157).

[201] I-Cheng Yeh and Che-hui Lien. “The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients”. In: *Expert Systems with Applications* 36.2 (2009), pp. 2473–2480 (cit. on p. 71).

[202] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. “Understanding the effect of accuracy on trust in machine learning models”. In: *ACM CHI Conference on Human Factors in Computing Systems*. 2019 (cit. on p. 267).

[203] Bianca Zadrozny and Charles Elkan. “Obtaining Calibrated Probability Estimates from Decision Trees and Naive Bayesian Classifiers”. In: *International Conference on Machine Learning*. 2001 (cit. on pp. 3, 4, 8, 10, 21, 27, 31, 35, 40, 64, 66–68, 74, 76, 79, 94, 101, 103, 153, 158, 178).

[204] Bianca Zadrozny and Charles Elkan. “Transforming Classifier Scores into Accurate Multiclass Probability Estimates”. In: *International Conference on Knowledge Discovery and Data Mining*. 2002 (cit. on pp. 4, 8, 11, 14, 34, 35, 40, 67, 79, 101, 102, 115, 122, 139, 178).

[205] Sergey Zagoruyko and Nikos Komodakis. “Wide Residual Networks”. In: *British Machine Vision Conference*. 2016 (cit. on p. 104).

[206] Jize Zhang, Bhavya Kailkhura, and T Han. “Mix-n-Match: Ensemble and Compositional Methods for Uncertainty Calibration in Deep Learning”. In: *International Conference on Machine Learning*. 2020 (cit. on pp. 17, 18, 40, 41, 97, 104, 140, 155).

[207] Lijun Zhang, Tianbao Yang, and Zhi-Hua Zhou. “Dynamic regret of strongly adaptive methods”. In: *International Conference on Machine Learning*. 2018 (cit. on p. 155).

[208] Shengjia Zhao, Tengyu Ma, and Stefano Ermon. “Individual Calibration with Randomized Forecasting”. In: *International Conference on Machine Learning*. 2020 (cit. on p. 175).

[209] Martin Zinkevich. “Online convex programming and generalized infinitesimal gradient ascent”. In: *International Conference on Machine Learning*. 2003 (cit. on p. 146).