Post-hoc calibration without distributional assumptions

Chirag Gupta

August 10, 2022

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA

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

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

Abstract

Most ML classifiers produce scores that indicate likelihood of class membership. These scores supplement class predictions and are often crucial for downstream decision making. However, no classifier guarantees that the produced scores are true probabilities in any sense. Scores can be interpreted as probabilities if they are calibrated. Informally, for binary classification (with labels 0 and 1), a score s in the interval [0,1] is said to be calibrated if the probability of the true label being 1 on the instances where s is predicted, is equal to s. A classifier is said to be calibrated if all scores it predicts are calibrated.

The primary goal of this thesis is to demonstrate that even if a given classifier is miscalibrated, additional data can be used to provably post-hoc calibrate it. Such calibration can be achieved in two different senses: (a) model calibration, which refers to calibration for a fixed data-generating distribution (given access to i.i.d. data from that distribution); and (b) forecast calibration, which refers to calibration against an online stream of data that is being generated adversarially. In this thesis, we touch upon both these forms of calibration.

The calibration algorithms we develop provably work under no assumptions about the data-generating procedure. In previous work, we showed such assumption-free guarantees in the binary calibration setting for binning methods—calibration algorithms that produce discretized (binned) scores. We also showed that the binary calibration algorithms and their guarantees can be extended to multiclass classification in a number of ways. We now propose two projects towards completion of this thesis. First, we suggest an approach for proving calibration guarantees for continuous (non-binned) calibration methods, also called scaling methods. Second, we propose ideas for specializing our multiclass calibration work for hierarchical classification, assuming that a taxonomy over the classes is known.

Contents

Overview and organization

Calibration has been studied in two different setups with historically disjoint literatures. Our primary focus in this thesis is *model calibration*, where we are interested in learning calibrated ML models [Platt et al., 1999, Zadrozny and Elkan, 2001, 2002]. No fixed model can be calibrated for arbitrarily generated data. Thus in order to define model calibration, one must 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 was first studied in a non-distributional, online learning style setup. Here, data is not assumed to be drawn from a fixed distribution, but can be arbitrary or even 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—actual data does not follow a distribution, but it is not being generated by an adversary either. 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 proposal is organized as follows. Chapters 1—3 are entirely in the binary classification setup. Multiclass calibration is discussed in Chapter 4.

1. In Chapter 1, we define model calibration and the post-hoc calibration setup.
2. In Chapter 2, we outline the goal of distribution-free post-hoc calibration, along with known results and open problems.
3. In Chapter 3, we define forecast calibration. In Section 3.1, we propose to use a forecast calibration technique called *calibeating* to derive a novel distribution-free post-hoc calibration algorithm.
4. In Chapter 4, we introduce multiclass model calibration. Our prior work on multiclass calibration does not leverage pre-existing relationships between the classes, such as a class hierarchy or taxonomy tree. In Section 4.3, we propose an approach to leverage this hierarchy.

All (sub)sections titled “Prior work: …” describe our work that has been completed and published as part of this thesis. All (sub)sections titled “Proposed work: …” describe proposed work towards completion of this thesis. The proposed timeline is at the end of the document, just before the bibliography.

Chapter 1

Model calibration (offline i.i.d. setting)

Let $g : \mathcal{X} \to [0, 1]$ be a 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 predictive 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 (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 formalized such a definition of approximate calibration in Gupta et al. [2020].

Definition 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 $\mathbb{1}\{g(\cdot) \geqslant t\}$ 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 1.1 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. We elaborate on this goal and the prior/proposed work in Chapter 2.

[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’.

Given a pre-learnt model g and calibration data $\mathcal{D} \sim P^n$, produce an estimate $m : [0,1] \to [0,1]$ of the mapping $g(X) \mapsto P(Y = 1 \mid g(X))$. If m is a good estimate, then $h := m \circ g \equiv m(g(\cdot))$ is better calibrated than g (for P).

Box 1: Post-hoc calibration of a pre-learnt model using held-out calibration data.

1.1 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 is easy to see (for instance, see Gupta et al. [2020, Proposition 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 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

$$\mathcal{D} := (X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n) \overset{i.i.d.}{\sim} P. \tag{1.3}$$

The paradigm of post-hoc calibration methods is summarized in Box 1. 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 2, we formalize Box 1 in a distribution-free setup.

Three methods for post-hoc calibration were proposed in close succession: Platt scaling [Platt et al., 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 1.1), a binary dataset with about 78% occurence of $Y = 0$ (no credit default).[2] This is() 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 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 3.1. 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.[3]

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

[3] 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.

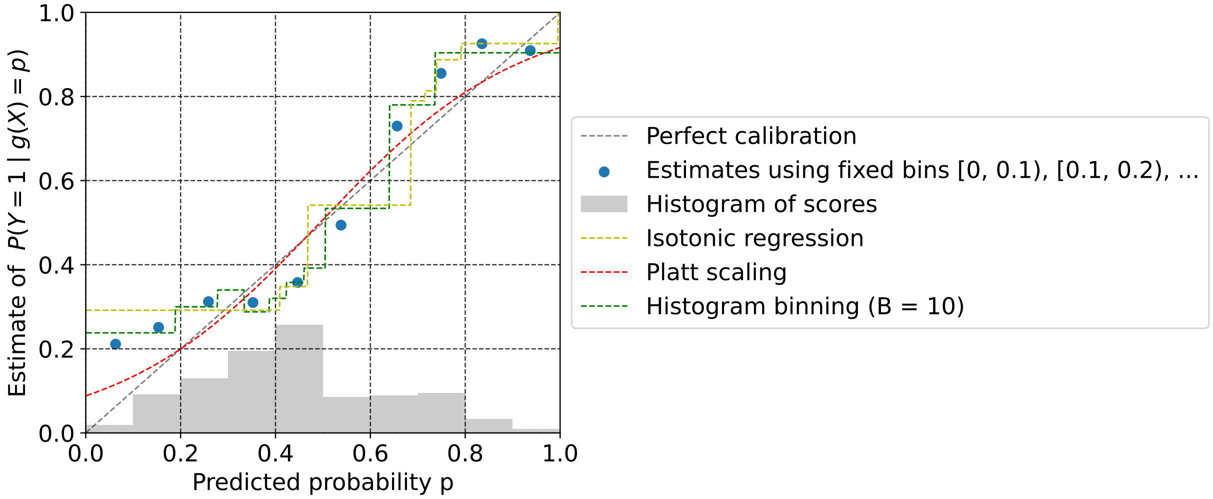


Figure 1.1: Post-hoc calibration of a logistic regression model $g : \mathcal{X} \to [0, 1]$. Plot is made for a fixed g on calibration data $\mathcal{D}$. The scores produced by g are miscalibrated as evidenced by the deviation of the blue scatter plot from the perfect calibration line. Post-hoc calibration 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) produces a smooth curve from a parametric family. Histogram binning (Section 1.3) and isotonic regression (Section 1.4) 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.

However, g appears miscalibrated. So, we look to estimate the mapping m on the same calibration data, as described in Box 1. The estimates produced by the aforementioned methods—Platt scaling, histogram binning, and isotonic regression—are plotted in Figure 1.1. In the following subsections, we describe the methods in further detail.

1.2 Platt scaling

Platt scaling learns the mapping from a parametric family:

$$\mathcal{M}_{\text{platt}} = \{m^{a,b} : a, b \in \mathbb{R}^2\}, \tag{1.4}$$

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

$$m^{a,b}(z) = \text{sigmoid}(az + b) = 1/(1 + e^{-(az+b)}). \tag{1.5}$$

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 1.1), the learnt parameters were $a \approx 4.7$ and $b \approx -2.3$. Thus the inflection point of the curve is roughly around $0.49 \approx 2.3/4.7$.

Algorithm 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 1.1 experiment).

1.3 Histogram binning

Histogram binning (Algorithm 1) learns a nonparametric mapping using a binning method. Nearby values of $g(x)$ are grouped together into a fixed number of bins, and a single estimate of the probability of $Y = 1$ is computed for each bin to define m. Algorithm 1 is directly borrowed from Gupta and Ramdas [2021]. We describe how it works in the following paragraph.

Histogram binning takes 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 1.1) 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 1.1 experiment).

1.4 Isotonic regression

The isotonic regression family corresponds to the nonparametric class of monotonically increasing mappings:

$$\mathcal{M}_{\text{isotonic}} = \{m : \text{for all } 0 \leqslant x \leqslant y \leqslant 1, m(x) \leqslant m(y)\}. \tag{1.6}$$

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

$$\begin{aligned} \underset{\widehat{\mu}_1,\widehat{\mu}_2,\ldots,\widehat{\mu}_n \in [0,1]}{\text{minimize}} \quad & \sum_{i=1}^{n} (Y_i - \widehat{\mu}_i)^2, \\ \text{such that} \quad & \forall i, j,\ \widehat{\mu}_i \leqslant \widehat{\mu}_j \iff Z_i \leqslant Z_j. \end{aligned} \tag{1.7}$$

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:

$$m(z) = \widehat{\mu}_j^\star, \text{where } Z_j = \max\{Z_i : Z_i \leqslant z\}. \tag{1.8}$$

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

$$m(z) = \max\{\widehat{\mu}_i^\star : Z_i \leqslant z\}, \tag{1.9}$$

because of the monotonicity constraint in the optimization problem (1.7).

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 1.1 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 $[0, 0.41)$.

Chapter 2

Distribution-free (DF) post-hoc calibration

We consider calibration guarantees for the post-hoc calibration methods discussed in Chapter 1, without making any assumptions about P, the underlying data-generating distribution. This is called the *distribution-free or DF* framework, a phrase which has recently been popular in the conformal prediction literature [Lei et al., 2018].

Let $\mathcal{A}$ be a post-hoc calibration algorithm that takes as input a given pre-learnt $g : \mathcal{X} \to [0, 1]$ along with calibration data $\mathcal{D} = \{(X_i, Y_i)\}_{i \in [n]}$, and outputs $\mathcal{A}(\mathcal{D}, g) = h : \mathcal{X} \to [0, 1]$, a predictor with presumably improved calibration properties compared to the original g. We formalize a DF calibration goal for $\mathcal{A}$ on the lines of approximate calibration (Definition 2). Recall that $\epsilon \in (0, 1)$ is a tolerance level of miscalibration and $\alpha \in (0, 1)$ is a tolerance level for probability of failure. The following formulation first appeared in Gupta et al. [2020] (equation (7) in arXiv version).

Definition 3 (Distribution-free (DF) calibration)**.** A post-hoc calibration method $\mathcal{A}$ is said to be DF calibrated if for every function g and every distribution P, $h = \mathcal{A}(\mathcal{D}, g)$ is approximately calibrated:

$$P^{n+1}(|P(Y = 1|h(X)) - h(X)| \geqslant \epsilon) \leqslant \alpha. \tag{2.1}$$

Above, P^{n+1} denotes the product distribution of the i.i.d. calibration data $\mathcal{D}$ and the test point (X, Y) (note that $h = \mathcal{A}(\mathcal{D}, g)$ is random over the calibration data $\mathcal{D}$). Additionally, P^{n+1} includes the internal randomization if $\mathcal{A}$ is a randomized algorithm; we accept this slight abuse of notation, writing P^{n+1} instead of a generic $\mathbb{P}$, in order to make it clear that the guarantee is also over the calibration data.

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 g or h).

Definition 4 (Distribution-free (DF) 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 DF 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, $h_n = \mathcal{A}(\mathcal{D}_n, g)$ satisfies condition (2.1) with parameters (ϵ_n, α).

Note that condition (2.1) requires approximate calibration not only over all P, but also over all g. Thus asymptotic calibration requires $\mathcal{A}$ to calibrate *any fixed* g over *all distributions* P, given i.i.d. data from P.

A primary goal of this thesis is to design calibration algorithms that are DF calibrated in the above senses. In Sections 2.1—2.3, we review prior work and open questions to this end. We briefly remark on additional related notions of DF calibration.

Remark 1. One can also consider alternative forms of DF calibration. In Gupta and Ramdas [2021], we outline the following:

1. A *conditional* DF guarantee:

$$P^n(\forall r \in \text{Range}(h),\ |P(Y = 1|h(X) = r) - r| \geqslant \epsilon) \leqslant \alpha. \tag{2.2}$$

We write the outer probability as P^n since (2.2) also has a PAC-style interpretation, which (2.1) does not. The condition (2.2) means that with probability $1 - \alpha$ over $\mathcal{D}$ (which is distributed as P^n), h satisfies the following deterministic property:

$$\forall r \in \text{Range}(h), |P(Y = 1|h(X) = r) - r| \leqslant \epsilon. \tag{2.3}$$

2. A bound on the expected calibration error (ECE). Define

$$\text{ECE}(h) := \mathbb{E}_P \left| P(Y = 1|h(X)) - h(X) \right|.$$

$\mathcal{A}$ is said to be DF calibrated with respect to the ECE if

$$P^n \left(\text{ECE}(h) \geqslant \epsilon \right) \leqslant \alpha. \tag{2.4}$$

Like (2.2), this bound also has a PAC-style interpretation since the outer probability is over the calibration data and the inner statement is a deterministic one about h.

Relationships between the proposed DF guarantees are discussed in Gupta and Ramdas [2021, Section 1].

2.1 Prior work: Platt scaling is not DF calibrated

Consider a slightly restricted class of Platt scaling mappings, those with $a \neq 0$.

$$\mathcal{M}_{\text{injective-platt}} = \{m^{a,b}(\cdot) : a, b \in \mathbb{R}^2, a \neq 0\}; \tag{2.5}$$

$m^{a,b}$ was defined in (1.5) as $m^{a,b}(z) = \text{sigmoid}(az + b) = 1/(1 + e^{-(az+b)})$. We call this class 'injective-platt' because each $m \in \mathcal{M}_{\text{injective-platt}}$ is injective:

$$\forall z_1 \neq z_2 \in [0, 1], m(z_1) \neq m(z_2).$$

In Gupta et al. [2020], we showed that satisfying (2.1) based on such an injective mapping class reduces to a problem of producing DF confidence intervals. However, producing DF confidence intervals with vanishing width (as the number of calibration points $n \to \infty$) is impossible without distributional assumptions [Barber, 2020]. This leads to the following theorem. .

Theorem 1 (Theorem 3 in arXiv version of Gupta et al. [2020])**.** *It is impossible for an injective post-hoc calibration algorithm to be distribution-free asymptotically calibrated.*

This lower bound also holds for other parametric methods that produce injective mappings, such as beta calibration [Kull et al., 2017], and to multiclass versions of Platt scaling called temperature scaling and vector scaling [Guo et al., 2017]. We expect the result to hold for any parametric class that produces smooth curves, even if the produced curves are not strictly injective. However, being a lower bound, the generality of Theorem 1 is perhaps less important than the methodological guidance we draw from it. Namely, we conclude from Theorem 1 that some binning-like non-injective mapping seems necessary for robust DF calibration guarantees. We can in fact show such guarantees for histogram binning, as discussed in the following subsection.

Nevertheless, Platt scaling often performs well in practice [Niculescu-Mizil and Caruana, 2005, Platt et al., 1999]. If data is less, Platt scaling performs better than binning methods [Gupta and Ramdas, 2021, Niculescu-Mizil and Caruana, 2005]. Some insight into the reason for this behavior can be gleaned from the proof of Theorem 1. In the proof, a distribution P is constructed for which the pre-learnt classifier g has no predictive power. It is shown that Platt scaling cannot satisfy calibration for this P, and thus cannot be DF calibrated. However, in practice, g itself is learnt on P-distributed data and has predictive utility. Platt scaling leverages this goodness of g.

Thus while DF guarantees cannot be shown for Platt scaling, we consider the possibility of showing other guarantees. In particular, we look to develop a robust version of Platt scaling that does as good as Platt scaling if a good Platt model exists, but reverts to binning if all Platt models are bad. In Section 3.1 we outline a proposed approach to this end, based on two tools: calibeating [Foster and Hart, 2021] and expert aggregation [Vovk, 1990].

2.2 Prior work: histogram binning is DF calibrated

In Gupta and Ramdas [2021], we showed that histogram binning is DF calibrated. The forthcoming result is for histogram binning with a small randomization term added to the scores—the randomization is only added to avoid a certain kind of degeneracy, and is not significant in practice.

Theorem 2 (Theorem 4, Gupta and Ramdas [2021]). *Suppose $n \geqslant 2B$ and let $\delta > 0$ be a small randomization parameter (arbitrarily small). Histogram binning is (ϵ, α)-calibrated for any $\alpha \in (0, 1)$ and*

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

Thus, as $n \to \infty$, histogram binning is DF asymptotically calibrated.

The main technical challenge in proving this result was the fact that histogram binning double dips the calibration data to compute both bin boundaries as well as the bin biases. In Algorithm 1 (line 15), m depends on the order statistics of the scores ($S_{(A_b)}$) as well as the estimated bin biases ($\widehat{\Pi}_b$). This double dipping makes it tricky to prove calibration guarantees for histogram binning.

We resolved this issue by exploiting a certain Markov property of order statistics (for one exposition, see Arnold et al. [2008, Chapter 2.4]). 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 b, the points in the bin are i.i.d. with distribution $Q_{Z|Z\in\text{Bin-}b}$. 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 bin b are also conditionally i.i.d. with distribution $Q_{Z|Z\in\text{Bin-}b}$. The Markov property allows us to double dip the data without sacrificing calibration properties, that is, the same data can be used for binning as well as estimating within-bin probabilities.

2.3 Open question: is isotonic regression DF calibrated?

The impossibility result of Theorem 1 does not apply for isotonic regression since it is a non-injective binning method. Showing DF calibration guarantees (or an impossibility result) for isotonic regression is currently open. Given the popularity and good performance of isotonic regression [Guo et al., 2017, Gupta and Ramdas, 2021, Niculescu-Mizil and Caruana, 2005, Zadrozny and Elkan, 2002], it would be interesting to resolve this question.

The techniques developed for histogram binning (described in the previous subsection), are not directly applicable for isotonic regression. In histogram binning, the bin boundaries are formed using *only* the scores $g(x)$; the labels play no part in this. Thus, we expect the labels to remain somewhat statistically independent even after *peeking* at the scores to form the bin boundaries. This intuition is elegantly captured by the Markov property of order statistics.

On the other hand, in isotonic regression, the labels are also used while forming bin boundaries. This is apparent from the theoretical formulation (1.7), as well as empirically (Figure 1.1, Table 1.2). In our experiment with the credit default dataset, we observed that when $g(x) \leqslant 0.4$, the true label values did not suggest a monotonic mapping. Consequently, the mapping learnt using isotonic regression merged all the $g(x) \leqslant 0.4$ points into a single bin. Showing calibration guarantees for isotonic regression will require arguing about the direct influence of label information in the bin formation, which we are currently unsure how to do. While proving DF guarantees for isotonic regression is quite relevant and interesting, it is not one of the problems we propose towards completion of this thesis.

Chapter 3

Forecast calibration (online adversarial setting)

Can we produce calibrated scores without knowing *anything* about the label-generating process—even if the labels are being produced adversarially? 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:

scores → forecasts, labels → 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.[1] At each time t, a forecast $p_t \in [0, 1]$ 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 Box 2 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.

[1] 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.

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.)

Box 2: Forecast-Calibration-Game-I. Nature knows p_t before revealing y_t. Parenthesized sentences instantiate the setup for the canonical example of rain prediction.

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

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

Box 3: Forecast-Calibration-Game-II. Nature knows the distribution of p_t before revealing y_t.

Definition 5 (Forecast calibration)**.** Forecasts $(p_1, p_2, \ldots) \in [0, 1]^\infty$ are said to be calibrated if

$$\text{for all } x \text{ such that } \lim_{T\to\infty} N_x^T \to \infty, \text{ we have } \lim_{T\to\infty} p_x^T = x. \tag{3.1}$$

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 (3.1) no matter how nature behaves. Nature's goal is to make the forecaster appear miscalibrated.

Since nature sees p_t, it is easy to satisfy her goal: play $y_t = \mathbb{1}\{p_t \leqslant 0.5\}$.[2] 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 (Box 3). 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 (3.1) 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.

[2] 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, 1985, Oakes, 1985].

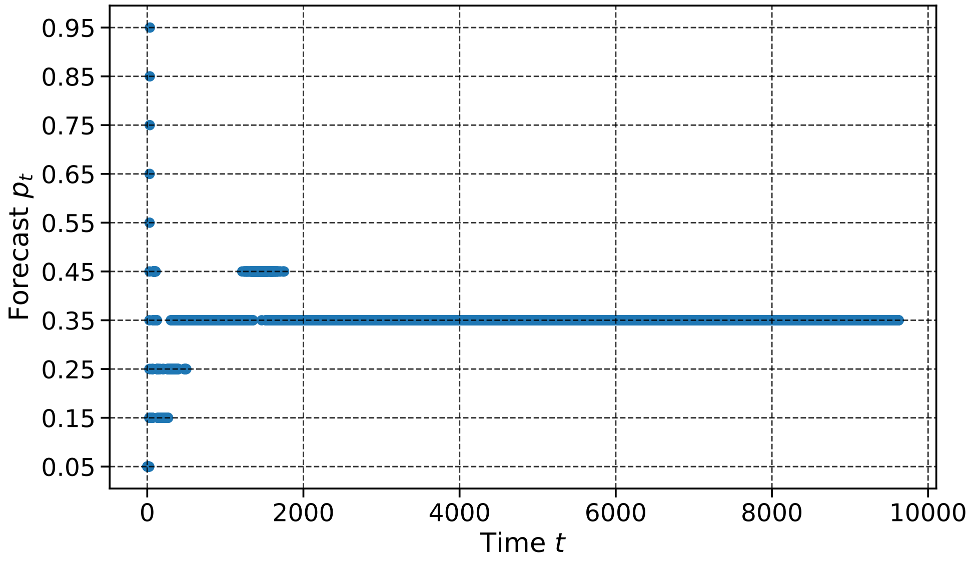


Figure 3.1: Foster [1999]'s ϵ-calibrated forecaster on Pittsburgh hourly rain data (2008-2012). The forecaster makes predictions on the grid $(0.05, 0.15, \ldots, 0.95)$. In the long run, the forecaster starts predicting 0.35 for every instance, closely matching the average number of instances on which it rained (≈ 0.37).

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 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 (3.1). We implement the algorithm for the grid $(0.05, 0.15, \ldots, 0.95)$. We observe (Figure 3.1) 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, predictive features are available for the label. These predictive features have predictive power even in complex scenarios such as if the data points are not independent, or if the relationship between features and labels is non-stationary. The simplest way to capture this predictive power 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 Chapters 1 and 2).

The proposed work in the following subsection is aimed at deriving an algorithm that is simultaneously robust to worst-case data and adaptive to the information offered by predictive features.

3.1 Proposed work: parametric post-hoc calibration mappings as expert forecasters

Suppose before making a prediction or forecast, we have access to the forecasts made by a number of expert forecasters. One of these expert forecasters may be perfectly calibrated, in which case we would like to identify and trust this expert. However, it is also possible that all of the experts provide misleading forecasts. In this case, we would like to make our own forecasts without relying on the experts.

Of particular interest to us is the infinite set of experts corresponding to parameterized post-hoc mappings (Section 1.1). Given a pre-defined scoring function $g : \mathcal{X} \to [0, 1]$, Platt scaling (equations (1.4) and (1.5)) learns a mapping from the output of g to another value. For parameters $(a, b) \in \mathbb{R}^2$, this mapping is given by $m^{a,b}(z \in [0, 1]) = \text{sigmoid}(az + b) \in [0, 1]$. We call the set of parameters as $\Theta_{\text{Platt}} = \mathbb{R}^2$. From now on, we refer to general parameter spaces as Θ and their elements as θ.

Generally, let Θ be an indexing over experts; $|\Theta|$ can be finite or infinite. In the case of post-hoc calibration, Θ would correspond to parametric or non-parametric recalibration maps based on the output of a pre-learnt g. For each $\theta \in \Theta$, suppose expert θ makes the prediction $m_t^\theta \in F$ at time t. We would like to use an aggregation algorithm $\mathcal{A}$ to aggregate these experts to produce another prediction in $[0, 1]$, $\mathcal{A} : \{m_t^\theta : \theta \in \Theta\} \mapsto [0, 1]$, such that the aggregated prediction is as well calibrated as the best expert in hindsight. Further, even if all experts are miscalibrated[3], we know that forecast calibration is nonetheless possible without experts (Chapter 3). Thus we would like our aggregated expert to still be calibrated.

For a decision-maker who plans to act on a score or forecast, it probably makes no difference if we forecast 0.301 or 0.3. Thus, let us simplify the setup and enforce that forecasts should be on some discrete grid, say an ϵ-grid $F = \{0, \epsilon, 2\epsilon, \ldots\}$. In the case of $m^\theta, \theta \in \Theta_{\text{Platt}}$ described above, we would have to discretize the prediction. Let χ be a discretization function that takes a prediction in $[0, 1]$ and outputs values in F. A natural example is the χ that rounds to the nearest multiple of ϵ: $\chi(x) = \epsilon \cdot \text{round}(x/\epsilon)$. Then $e^\theta := \chi \circ m^\theta \equiv \chi(m^\theta(\cdot))$ is a discretized Platt mapping.

$$\mathcal{M}_{\text{discretized-platt}} = \{e^\theta = \chi \circ m^\theta : m^\theta \in \mathcal{M}_{\text{platt}}\}. \tag{3.2}$$

Foster and Hart [2021] describe a procedure that 'calibeats' an expert forecaster, or is better calibrated than the expert forecaster. Their approach is to use the same bins or partitioning produced by the expert forecaster, but to produce a better calibrated forecast for every given bin. The calibeating algorithm is quite natural: maintain the running average of the observations for every bin and predict it as the calibrated score. The binning scheme (which is determined by the expert forecaster) governs the refinement of the forecast. The calibeating procedure does not change the refinement since the binning scheme does not change. On the other hand, the actual values we assign to the bins governs the calibration score. This latter entity can be pushed down to zero, using calibeating.

If the number of experts forecasters is finite and given by N, then Foster and Hart [2021, Appendix A.7.2] show how to calibeat all of them simultaneously with an ϵ-calibration rate of

[3] in the absence of distributional assumptions, this is always a possibility (see Section 2.1)

$\sqrt{(N \log T)/T}$. To beat multiple experts simultaneously, we first construct a meta-expert whose Brier score is at least as good as that of the best expert in hindsight. Then, we calibeat the meta-expert.

The rate of $\sqrt{(N \log T)/T}$ may be unsatisfactory if N is large. For instance, in the case of discrete-Platt forecasters $\{e^\theta : \theta \in \Theta_{\text{Platt}}\}$ described above. It can be showed that at time T, the number of these forecasters is $\Omega(T^{|F|})$, where $|F|$ is the size of the ϵ-grid F. Nevertheless, by virtue of being monotonic mappings on top of the output of g, the Platt forecasters are very closely related to each other. We want to leverage this relationship to prove a better bound.

The problem boils down to producing a meta-expert that has a lower Brier score than the individual experts. For this, we propose to use Vovk's aggregated algorithm [Vovk, 1990]. Zhdanov and Vovk [2010] showed that the Brier score for *linear experts* is 1-mixable, implying that a $O(\log T)$ regret can be obtained for the Brier score. However, in the case of the Platt scaling class, the mapping is not just linear, but linear with a sigmoid transformation in the end. We foresee the following challenges/sub-questions that would need to be addressed:

1. To the best of our knowledge, efficient (say poly $\log(N)$) aggregation of sigmoid-linear experts is not a solved problem.

2. Instead of Platt scaling, we could consider a class of linear mappings, which as described above, is 1-mixable. However, linear mappings may not be the 'right' class for post-hoc calibration mappings.

3. On the other hand, instead of the Brier score, we could consider calibeating the log loss (or cross-entropy loss). The log loss for a sigmoid-linear mapping corresponds to the logistic loss over the linear mapping, which is 1-mixable [Foster et al., 2018, Shamir, 2020].[4] Thus with this approach, the aggregation of experts poses fewer technical problems, but the appropriate calibeating procedure is unclear.

4. The aggregated forecaster lies in the space $[0, 1]$. However, for implementing the calibeating algorithm, the output needs to be discretized to the ϵ-grid F. We hypothesize that discretization should not affect calibration significantly.

5. It is of practical importance that on real-world forecasting tasks, the forecaster we construct performs better than an expertise-agnostic ϵ-calibrated forecaster (like Foster [1999]). To perform experimental validation, we will identify appropriate time-series datasets where predictive features are available.

Finally, it would be interesting to see the performance of the forecaster on true i.i.d. data and compare it to other post-hoc calibration algorithms in terms of model calibration (Chapter 1). To this end, we briefly note that since Brier score is a proper scoring rule [Gneiting and Raftery, 2007], if the parametric class Θ is well-specified (meaning that the true data is drawn as per the distribution forecasted by one of the experts in the parametric class at each time step), then the asymptotically optimal forecaster in hindsight will be the data-generating forecaster.

We note a couple of interesting works on forecast aggregation that may be relevant for us, and will be considered while solving this problem [Neyman and Roughgarden, 2021, Turner et al., 2014].

[4] see also `https://www.dylanfoster.net/posters/logistic_colt2018.pdf`

Chapter 4

Multiclass model 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] and class-wise calibration [Kull et al., 2017].

In recent work [Gupta and Ramdas, 2022], we proposed a new notion of multiclass calibration, called *top-label calibration*, and provided a unified perspective on the various relaxations of canonical calibration, called *multiclass-to-binary reductions*. These are briefly discussed in the following subsections, to provide context for the proposed work on calibration for hierarchical classification.

4.1 Prior work: top-label calibration

Top-label calibration reduces multiclass calibration to a single binary calibration requirement corresponding to the predicted top class, called the top-label in this context. A classifier is said to be top-label calibrated if the reported probability for the top-label is calibrated, 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 6 (Top-label calibration)**.** The predictor (c, h) is said to be top-label calibrated (for the data-generating distribution P) if

$$P(Y = c(X) \mid c(X), h(X)) = h(X). \tag{4.1}$$

Calibration notion	Quantifier	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 4.1: 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.

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 can be provably achieved in a distribution-free framework (2) using a modification of binary histogram binning (Section 1.3). We described the algorithm, top-label histogram binning, and proved associated calibration guarantees in [Gupta and Ramdas, 2022, Appendix B]. 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 [Gupta and Ramdas, 2022, Table 2].

4.2 Prior work: multiclass-to-binary (M2B) reductions

A number of different notions of multiclass calibration have been proposed. In addition to top-label calibration described in the previous subsection, a few others are confidence calibration [Guo et al., 2017], top-K-confidence calibration [Gupta et al., 2021], and class-wise calibration [Kull et al., 2017]. We proposed a multiclass-to-binary (or M2B) reduction framework to unify these based on a simple observation: each of these notions reduce multiclass calibration to one or more binary calibration requirements. 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 4.1 illustrates the M2B framework by describing each of the calibration notions mentioned in the previous paragraph. While we have not defined class-wise calibration, the definition can be read off from the third line of Table 4.1: for every $l \in [L]$, the conditioning is on pred$(X) = h_l(X)$, and a single binary calibration statement is verified: $P(Y = l \mid \text{pred}(X)) = h_l(X)$. Thus, $\mathbf{h}$ is said to be class-wise calibrated (for P) if:

$$\forall l \in [L], P(Y = l \mid h_l(X)) = h_l(X).$$

The benefit of the framework is that for every M2B notion, a novel post-hoc calibration algorithm can be derived which is adapted to the given notion. This new algorithm first carves out a number of binary calibration problems from the overall multiclass calibration problem, based on the individual binary calibration claims in the M2B characterization. Then, each of the individual binary calibration problems can be solved separately using any binary calibration algorithm (such as Platt scaling, histogram binning, or isotonic regression, described in Section 1.1). The top-label histogram binning approach mentioned in the previous subsection is also derived from

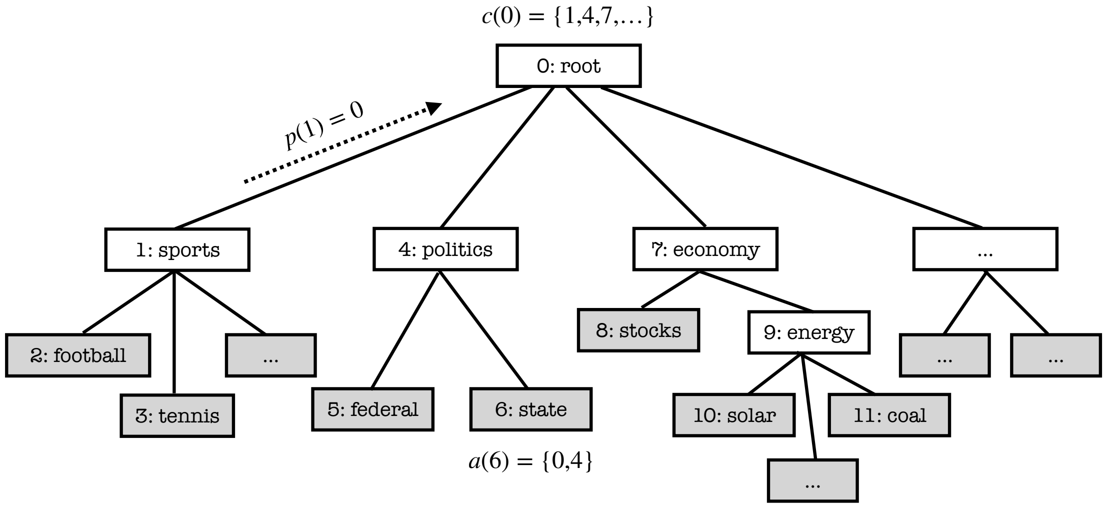


Figure 4.1: An illustrative taxonomy over the set of topics (classes) for a news article. Broader topics such as `sports` contain narrower topics such as `football` and `tennis`. The hypothetical `root` class is numbered 0. The actual L classes are numbered $[L] = \{1, \dots, L\}$. Leaf classes are the shaded boxes. The figure also provides one example each of the 'child-of', 'parent-of', and 'ancestor-of' functions (c, p, a respectively).

this framework. Like top-label histogram binning, any M2B + histogram binning algorithm satisfies theoretical guarantees and exhibits good empirical performance; see our paper [Gupta and Ramdas, 2022] for more details.

4.3 Proposed work: calibration for hierarchical classification

Suppose the L classes of interest are related to each other via a known taxonomy tree, such as in Figure 4.1 which represents an illustrative taxonomy tree over topics (the classes) for news articles. We use this taxonomy as a running example. Each node of a taxonomy tree represents a class, with parent nodes representing broader classes that subsume the narrower classes corresponding to their children. The root node subsumes all L classes. We label each non-root node with the index of the class that the node corresponds to, and we label the root node as 0. The complete set of nodes is denoted as $\mathcal{N} := \{0, 1, \dots, L\}$. The set of non-root nodes which is also the set of classes is denoted, as usual, as $[L]$.

Further, we use $c : \mathcal{N} \to 2^{[L]}$ to denote the 'child-of' function (mapping every node to its children) and $p : \mathcal{N}\backslash\{0\} \to \mathcal{N}$ be to denote its inverse, the 'parent-of' function (mapping every node to its parent). Thus we have that for every class $l \in [L]$, $p(l) \in \mathcal{N}$ is the unique node such that $l \in c(p(l))$. Also, let $a : \mathcal{N} \to 2^{\mathcal{N}}$ be the 'ancestor-of' function, mapping a node to the set of its ancestors. Every node is considered its own ancestor, that is, for all $l \in \mathcal{N}$, $l \in a(l)$. We denote the set of all leaf nodes as $\mathcal{L} := \{l : c(l) = \varnothing\}$. All notation introduced so far is summarized in Table 4.2

A given data-point x can belong to multiple classes; let the assigned label $y \in [L]$ to corre-

$\mathcal{N}$	set of all nodes including the root node labeled 0
$[L]$	set of non-root nodes, equivalently the L classes
$\mathcal{L}$	set of leaf nodes, equivalently the narrowest classes
$c : \mathcal{N} \to 2^{[L]}$	child-of function mapping a node to the set of its children
$p : \mathcal{N}\backslash\{0\} \to \mathcal{N}$	parent-of function mapping a node to its parent
$a : \mathcal{N} \to 2^{\mathcal{N}}$	ancestor-of function mapping a node to the set of its ancestors

Table 4.2: Notation used to describe the L-class taxonomy.

spond to the narrowest class to which x belongs. Thus x also belongs to each of the ancestors of y, $a(y)$ (recall that the root node 0 is an ancestor as well, although it is not one of the L classes). As such, we allow x to be labeled as a non-leaf node y, in which case x does not belong to any of the sub-classes of y.

In hierarchical classification, we want to produce a classifier $c : \mathcal{X} \to \mathcal{N}$ that maps a feature vector in $\mathcal{X}$ to a node representing a class prediction. Due to the hierarchical setup, $c(x)$ is understood as a prediction that x belongs to $c(x)$ as well as its ancestors. We define the accuracy of c as the probability that the true class is an ancestor of the predicted class:

$$\text{Accuracy of } c := P(Y \in a(c(X))). \tag{4.2}$$

To produce an accurate c in this sense, one can just predict the class 0 each time. Thus a secondary requirement of narrowness is needed, for example, the narrowness of c can be defined as the expected *depth* of $c(X)$ in the taxonomy tree. We leave the informal definitions of depth and narrowness vague and move on to the main object of our study.

4.3.1 Path calibration: a proposed notion of hierarchical calibration

Suppose one wants to supplement c with calibrated scores. To start with, if one cared only about the predicted probability of the narrowest predicted class, $c(x)$ (and not the ancestors of $c(x)$), then the problem reduces to top-label calibration (Section 4.1). Given the hierarchical nature of the labels however, one would expect to receive scores not just for $c(x)$ but for every ancestor of $c(x)$. Path calibration is calibration for all these labels.

Definition 7 (Path calibration)**.** The scoring function $\mathbf{h} : \mathcal{X} \to ([0, 1] \cup \bot)^L$ is said to be path calibrated (for c and P) if it produces calibrated scores for ancestors of $c(X)$ in the following sense:

$$\text{for every } l \in a(c(X)),\ P(l \in a(Y) \mid c(X), \mathbf{h}(X)) = h_l(X), \tag{4.3}$$

and abstains ($\bot$) for non-ancestors of $c(X)$:

$$\text{for every } l \notin a(c(X)),\ h_l(X) = \bot.$$

The event $l \in a(Y)$ may appear cryptic at first, so we illustrate with an example. Table 4.3 shows what $l \in a(Y)$ means for the Figure 4.1 taxonomy, with l being the ancestors of the `solar` class. Path calibration is the hierarchical classification version of top-label calibration due to the conditioning on the predicted class $c(X)$. While in top-label calibration, one only cares about

Class l	Event $l \in a(Y)$ translates to
10 (`solar`)	Y = 10 (since `solar` is a leaf class)
9 (`energy`)	$Y \in \{9, 10, 11\}$ ({`energy, solar, coal`})
7 (`economy`)	$Y \in \{7, 8, 9, 10, 11\}$ ({`economy, stocks, energy, solar, coal`})

Table 4.3: Translation of the event $l \in a(Y)$ from the definition of path calibration (4.3), for ancestors of the leaf class `solar`, based on the Figure 4.1 taxonomy.

the predicted probability for the predicted class, in path calibration, one must also consider the predicted probabilities for ancestors of the predicted class.

For any reasonable $\mathbf{h}$, path calibrated or not, we expect that classes should receive smaller scores than their ancestors. We call this property *path monotonicity*.

Definition 8 (Path monotonicity)**.** A scoring function $\mathbf{h}$ is said to be path monotonic if for all $x \in \mathcal{X}$, every class $l \in a(c(x))$, and $l^A \in a(l)$, $h_{l^A}(x) \geqslant h_l(x)$.

It can be shown that if $\mathbf{h}$ is perfectly path calibrated, then it is path monotonic (formal result suppressed for brevity). However, path monotonicity may not be satisfied by an approximately path calibrated $\mathbf{h}$ (for which the $=$ of (4.3) holds approximately and with high probability). This sets up the following question.

Question: How can we achieve distribution-free approximate path calibration while satisfying the path monotonity requirement?

The M2B framework for multiclass calibration (Section 4.2) works if calibration claims being made are akin to binary calibration. In path calibration, the conditioning is on a vector-valued function $\mathbf{h}$, not a scalar-valued function. Thus the M2B framework does not solve the path calibration problem.

4.3.2 A preliminary approach for post-hoc path calibration

The goal here is to post-hoc path calibrate an existing hierarchical classifier given access to a calibration dataset $\mathcal{D} \sim P^n$. Let us assume that the classifier predicts some leaf class as the most likely class (instead of sometimes predicting an internal node). That is, $c : \mathcal{X} \to \mathcal{L}$, can be inferred from the hierarchical classifier. Let $g : \mathcal{X} \to [0, 1]$ be the confidence score for the predicted leaf class.

In top-label histogram binning [Gupta and Ramdas, 2022], bins are created separately for each class l based on the confidence function g. Then, for each bin, the probability of the label being l is estimated. For path calibration, we create bins on the leaf nodes exactly as we would do in top-label calibration. Then, we estimate the bin biases separately for all ancestors of the leaf node. For every leaf class $l \in \mathcal{L}$, we do the following:

1. **Create a calibration sub-dataset corresponding to l.** Namely,

$$\mathcal{D}_l := \{(g(X_i), Y_i) : c(X_i) = l\}.$$

   $\mathcal{D}_l$ represents the set of calibration points that are predicted to belong to leaf class l, with the features replaced by the confidence scores for l.

2. **Define binning scheme.** A binning scheme throws data-points (in feature space $\mathcal{X}$) into one of many finite bins. Let $B_l \in \mathbb{N}$ be a hyperparameter specifying the number of bins. Based on the dataset $\mathcal{D}_l$, we define a binning scheme $\mathcal{B}_l : \mathcal{X} \to [B_l]$. One could using any binning scheme; as a first pass we will use histogram binning (see Sections 1.3 and 2.2).
3. **Learn bin biases for all ancestors of l.** Consider some point $x \in \mathcal{X}$ such that $c(x) = l$. Let the bin for point x be $\mathcal{B}_l(x) = b$. For every ancestor $l' \in a(l)$, we define $\widehat{\pi}_{l',b,l}$ based on the calibration data $\mathcal{D}$ as follows:

$$\begin{aligned}\widehat{\pi}_{l',b,l} &:= \frac{\text{\# points predicted as class } l \text{ and bin } b\text{, that belong to class } l'}{\text{\# points predicted as class } l \text{ and bin } b} \\ &= \frac{|\{c(X_i) = l,\ \mathcal{B}_l(X_i) = b,\ l' \in a(Y_i) : (X_i, Y_i) \in \mathcal{D}\}|}{|\{c(X_i) = l,\ \mathcal{B}_l(X_i) = b : (X_i, Y_i) \in \mathcal{D}\}|}.\end{aligned} \tag{4.4}$$

   The requirement 'points that belong to class l'' is capture by the condition $l' \in a(Y_i)$ due to the hierarchical nature of the problem.
4. **Aggregate the bin biases into a single predictor.** Finally, $h_{l'}(x)$ is defined piecewise based on the predicted class and bin identities. If $c(x) = l$, and $\mathcal{B}_l(x) = b$, then

$$h_{l'}(x) = \begin{cases} \widehat{\pi}_{l',b,l} & \text{if } l' \in a(l), \\ \perp & \text{if } l' \notin a(l). \end{cases} \tag{4.5}$$

It can be verified that the predictor $\mathbf{h}$ produced using the above approach will be path monotonic (Definition 8). Further, $\mathbf{h}$ exactly reduces to the top-label predictor for the leaf classes.

We expect that calibration guarantees similar to those shown for top-label histogram binning [Gupta and Ramdas, 2022] would also hold in the path calibration sense for the procedure described above. However, the proposed approach has the following drawbacks, which we will be primarily interested in resolving:

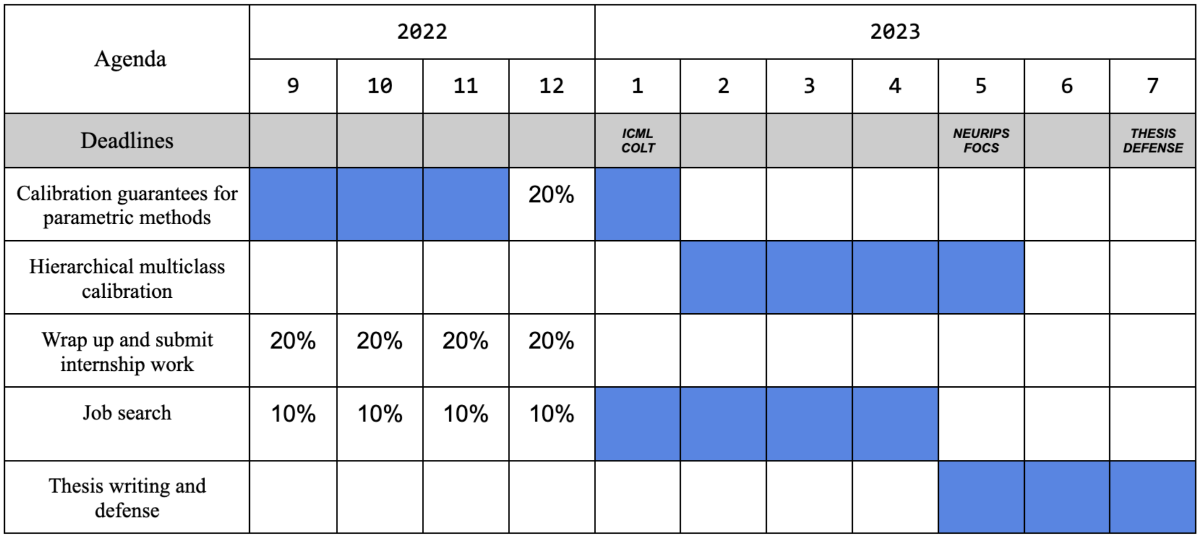
1. Using only the scores for the predicted leaf class for binning ignores the scores that the hierarchical classifier produces for ancestors of the leaf class. Binning based on the full vector-valued hierarchical prediction is also tricky (see Gupta and Ramdas [2022, Appendix G]). We will attempt to identify a practical middle ground.
2. Classes that are ancestors of the leaf class are broader, and perhaps easier to make calibrated predictions for, since more data is available for them. However, the current approach of estimating the bin biases $\widehat{\pi}_{l',b,l}$ (4.4) relies only on the calibration points for which $c(X_i) = l$.

Both drawbacks mentioned are regarding statistical efficiency, and not validity. In this thesis, we will investigate the practical properties of the proposed path calibration algorithm. Our first line of investigation will be to evaluate the extent of statistical inefficiency in practice by implementing the preliminary approach on actual hierarchical classification datasets. Based on the empirical study, we will propose improvements to the current procedure.

We identified some sources for relevant datasets: `https://sites.google.com/site/hrsvmproject/datasets-hier`, `https://www.imageclef.org/2010/ICPR/`, `https://press.liacs.nl/mirflickr/#sec_introduction`, `http://kt.ijs.si/DragiKocev/PhD/resources/doku.php?id=hmc_classification`. On looking at some of these datasets more carefully, we found that class labels are often not on a single branch of the taxonomy. For instance, a news document classified into the Figure 4.1 taxonomy can belong to both `coal` and `politics` simultaneously. Such multi-branch classification is beyond the scope of the problem as described currently, but we will consider expanding our investigation to multi-branch hierarchical classification.

Timeline for proposed work

The expected graduation date is July 31, 2023. I am aiming to defend in the first half of July, 2023.



Agenda	2022				2023						
	9	10	11	12	1	2	3	4	5	6	7
Deadlines					ICML COLT				NEURIPS FOCS		THESIS DEFENSE
Calibration guarantees for parametric methods	🟦	🟦	🟦	20%	🟦						
Hierarchical multiclass calibration						🟦	🟦	🟦	🟦		
Wrap up and submit internship work	20%	20%	20%	20%							
Job search	10%	10%	10%	10%	🟦	🟦	🟦	🟦			
Thesis writing and defense									🟦	🟦	🟦

Bibliography

Barry C Arnold, Narayanaswamy Balakrishnan, and Haikady Navada Nagaraja. *A first course in order statistics*. SIAM, 2008. 2.2

Miriam Ayer, H Daniel Brunk, George M Ewing, William T Reid, and Edward Silverman. An empirical distribution function for sampling with incomplete information. *The Annals of Mathematical Statistics*, pages 641–647, 1955. 1.4

Rina Foygel Barber. Is distribution-free inference possible for binary regression? *Electronic Journal of Statistics*, 14(2):3487–3524, 2020. 2.1

Richard E Barlow. Statistical inference under order restrictions; the theory and application of isotonic regression. Technical report, 1972. 1.4

A Philip Dawid. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982. (document), 2

A Philip Dawid. Comment: The impossibility of inductive inference. *Journal of the American Statistical Association*, 80(390):340–341, 1985. 2

Dean P Foster. A proof of calibration via Blackwell's approachability theorem. *Games and Economic Behavior*, 29(1-2):73–78, 1999. 3.1, 3, 5

Dean P Foster and Sergiu Hart. "Calibeating": Beating forecasters at their own game. 2021. 2.1, 3.1

Dean P Foster and Rakesh V Vohra. Asymptotic calibration. *Biometrika*, 85(2):379–390, 1998. (document), 3

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. 3

Drew Fudenberg and David K Levine. An easier way to calibrate. *Games and economic behavior*, 29(1-2):131–137, 1999. (document)

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007. 3.1

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017. 2.1, 2.3, 4, 4.2

Chirag Gupta and Aaditya Ramdas. Distribution-free calibration guarantees for histogram binning without sample splitting. In *International Conference on Machine Learning*, 2021. 1.3, 1, 1, 2.1, 2.2, 2, 2.3

Chirag Gupta and Aaditya Ramdas. Top-label calibration and multiclass-to-binary reductions. In *International Conference on Learning Representations*, 2022. 4, 4.1, 4.2, 4.3.2, 4.3.2, 1

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. 1, 1.1, 2, 2.1, 1

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. 4.2

Meelis Kull, Telmo M Silva Filho, and Peter Flach. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 11(2):5052–5080, 2017. 2.1, 4, 4.2

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018. 2

Eric Neyman and Tim Roughgarden. From proper scoring rules to max-min optimal forecast aggregation. *arXiv preprint arXiv:2102.07081*, 2021. 3.1

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *International Conference on Machine Learning*, 2005. 2.1, 2.3

David Oakes. Self-calibrating priors do not exist. *Journal of the American Statistical Association*, 80(390):339–339, 1985. 2

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. (document), 1.1, 2.1

Gil I Shamir. Logistic regression regret: What's the catch? In *Conference on Learning Theory*, 2020. 3

Brandon M Turner, Mark Steyvers, Edgar C Merkle, David V Budescu, and Thomas S Wallsten. Forecast aggregation via recalibration. *Machine learning*, 95(3):261–289, 2014. 3.1

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. 4

Volodimir G Vovk. Aggregating strategies. In *Computational Learning Theory*, 1990. 2.1, 3.1

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *International Conference on Machine Learning*, 2001. (document), 1.1

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002. (document), 1.1, 2.3

Fedor Zhdanov and Vladimir Vovk. Competitive online generalized linear regression under square loss. In *Joint European Conference on Machine Learning and Knowledge Discovery* *in Databases*, 2010. 3.1