Non-parametric Bayesian Isotonic Calibration: Fighting Over-confidence in Binary Classification

Mari-Liis Allikivi(✉)[0000−0002−1019−3454] and Meelis Kull[0000−0001−9257−595X]

Institute of Computer Science
University of Tartu
Tartu, Estonia
{mari-liis.allikivi,meelis.kull}@ut.ee

Abstract. Classifiers can often output a score or a probability indicating how sure they are about the predicted class. Classifier calibration methods can map these into *calibrated* class probabilities, supporting cost-optimal decision making. Isotonic calibration is the standard non-parametric calibration method for binary classifiers, and it can be shown to yield the most likely monotonic calibration map on the given data, where monotonicity means that instances with higher predicted scores are more likely to be positive. Another non-parametric method is ENIR (ensemble of near-isotonic regression models) which allows for some non-monotonicity, but adds a penalty for it. We first demonstrate that these two methods tend to be over-confident and show that applying label smoothing improves calibration of both methods in more than 90% of studied cases. Unfortunately, label smoothing reduces confidence on the under-confident predictions also, and it does not reduce the raggedness of isotonic calibration. As the main contribution we propose a non-parametric Bayesian isotonic calibration method which has the flexibility of isotonic calibration to fit maps of all monotonic shapes but it adds smoothness and reduces over-confidence without requiring label smoothing. The method introduces a prior over piecewise linear monotonic calibration maps and uses a simple Monte Carlo sampling based approach to approximate the posterior mean calibration map. Our experiments demonstrate that on average the proposed method results in better calibrated probabilities than the state-of-the-art calibration methods, including isotonic calibration and ENIR.

Keywords: Binary classification · Classifier calibration · Non-parametric Bayesian.

1 Introduction

With the advances in artificial intelligence, classifiers are being incorporated into more and more decision-making processes. Sometimes it is enough to base the decisions only on the classifier's predicted labels. However, more often decision making benefits from knowing about how confident the classifier is in its prediction. For instance, in a medical diagnostic setting a high-confidence predicted positive

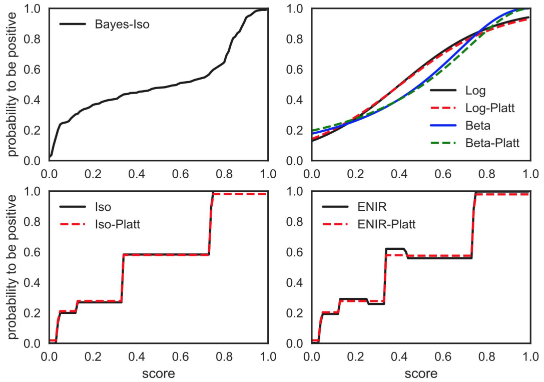

Fig. 1. Examples of calibration curves of the state-of-the-art calibration methods with and without Platt correction.

might be fully trusted by the doctor, whereas for low-confidence predicted positives the doctor might conduct additional tests. This usage requires the diagnostic classifier to be well-calibrated and not over-confident, since errors at high confidence levels are very costly. Most algorithms for learning binary classifiers can provide some kind of scores interpretable as confidence levels. For instance, in margin-based classifiers the distance from the decision boundary reflects confidence. For decision-maker's benefit it is useful if the confidence scores can be related to the expected probability of error. This is achieved, if the classifier outputs *calibrated class probabilities* [20]. The class probabilities in binary classification are calibrated, if among all instances predicted to be positive with probability p, the proportion of actual positives is also approximately p, for any $p \in [0, 1]$. Such interpretability of predicted probabilities combined with information about how much a false positive or a false negative would cost, allows decision-makers to estimate the expected cost for each possible decision and to follow the least costly option [5].

If the classifier outputs non-calibrated probabilities or confidence scores that are not probabilities at all, then one can apply classifier calibration methods to transform these outputs into the scale of calibrated probabilities. In case of binary classification this transformation can be represented as a mapping from real-valued output scores into probabilities to be positive, known as the *calibration map*, see examples in Figure 1. There are two approaches to finding these mappings: parametric and non-parametric. The best known parametric and non-parametric calibration methods are logistic calibration (also known as Platt scaling) [16] and isotonic calibration [21], respectively. Both methods model calibration maps as non-strictly monotonically increasing, also called *isotonic*. The reasoning behind this assumption is that if the classifier's confidence in the positive prediction increases then the probability to be positive should also increase.

Logistic calibration (also known as Platt scaling) fits a logistic sigmoid on the training data [16]. This method has two parameters, one determining its centre and another determining its slope at the centre. It can be implemented by applying univariate logistic regression to predict the binary label (1 for positive and 0 for negative) from the model output score. To reduce overfitting, Platt proposed a correction to the procedure and instead of 1 and 0 use labels $1 - \frac{1}{N_{+}+2}$ and $0 + \frac{1}{N_{-}+2}$ in fitting logistic regression [16], where N_{+} and N_{-} are the numbers of positives and negatives in the training data. This correction procedure is essentially label smoothing [6] but with a particular fixed amount of smoothing. We use notation 'Log' and 'Log-Platt' to refer to the uncorrected and corrected method, respectively.

Logistic calibration can be derived from first principles if assuming that the model output scores on the positives and negatives are both Gaussian distributed, with the same variance but different means. If the model outputs scores that are already probabilities but still require calibration, then it is more natural to use beta distributions instead of Gaussians, because beta distributions have support over the range $[0, 1]$. Following this reasoning, the paper [8] derived the Beta calibration method [8]. Beta calibration is a parametric family with 3 parameters, allowing a larger variety of shapes for the calibration map than logistic calibration. The family contains reverse sigmoidal functions and also the identity map, allowing the method to return the probabilities unchanged if the model is already calibrated, a property that logistic calibration does not have.

Isotonic calibration is a non-parametric method, not constrained by the shapes within a particular parametric family. It uses PAV (pool adjacent violators) algorithm to learn a calibration map which is optimal on the training data, in the sense that no other monotonic calibration map yields a lower squared error between the resulting calibrated probabilities and actual binary labels [21]. As optimality is determined on the scores present in the training instances, the values of the calibration map on other scores are not determined: these gaps are filled in by linear interpolation or by extension into a piecewise constant function.

Ensemble of near isotonic regression (ENIR), is a calibration method that is based on and is shown to improve isotonic calibration [10]. It drops the monotonicity constraint, which makes sense in cases where the ROC curve of the classifier is non-convex. ENIR makes multiple calls to the near isotonic regression algorithm [18] which introduces a penalty for non-monotonicity into the loss measure. Each call is with a different value for penalty and the results are averaged with weights to obtain the final calibration function.

Finally, there are several non-parametric methods using binning, either by fixed width, fixed size, or more advanced methods, such as BBQ [12] and ABB [11]. However, these methods have been shown in [10] to be inferior to ENIR, so we will not consider them further in this paper.

It has been shown in [14] that logistic calibration outperforms isotonic calibration on smaller datasets and vice versa on larger datasets. This is because non-parametric methods overfit on smaller data whereas parametric methods have less tendency to overfit. At the same time, when enough data is provided for calibration, non-parametric methods can learn many different shapes while parametric methods are restricted to their parametric families. These statements will become one of the basis for constructing our experiments and interpreting the results.

In the following Section 2 we introduce proper losses as evaluation measures for calibration. In Section 3 we demonstrate that the existing non-parametric calibration methods are over-confident and propose to use Platt's correction, reducing log-loss and squared error in more than 90% of our studied cases. In Section 4 we propose our main contribution, a new non-parametric Bayesian isotonic calibration method. In Section 5 we perform experiments on synthetic and real data to demonstrate that on average, the new method performs either best or tied with best for all considered calibration set sizes and loss measures. Finally, Section 6 concludes and discusses future work.

2 Evaluation of Calibration

Following the definition of calibrated probabilities one needs to check whether among all instances with the same predicted probability p the actual proportion of positives is also close to p. However, for methods outputting a continuous scale of probabilities in $[0, 1]$ there is hardly any hope to find multiple instances with exactly the same predicted probability p. One way to evaluate calibration methods is to introduce bins around p and compare p to the empirical proportion of positives in the bins, as done by measures such as ECE (expected calibration error) [7]. Such methods ignore the differences of predictions within each bin, and therefore measure calibration to a limited granularity.

However, there is an alternative to this: *proper losses* (also called *proper scoring rules*). Proper losses are minimized if the calibration method achieves perfectly calibrated probabilities, due to the decomposition into *calibration loss* and *refinement loss* [3, 9]. Since refinement loss cannot decrease during calibration, any reduction in overall loss must be due to the reduction in calibration loss.

The best known proper losses are log-loss (a.k.a. cross-entropy) and Brier score (a.k.a. squared error), which are standard evaluation measures of class probability estimators [14]. If the instance is positive and the model predicts it to be positive with probability $\hat{p}$, then log-loss (LL) penalizes it with loss $-\ln \hat{p}$ and Brier score (BS) with loss $(1-\hat{p})^2$. If the instance is negative, then the losses are $-\ln(1-\hat{p})$ and $\hat{p}^2$, respectively. Both these losses are non-negative and minimized if the prediction is correct and with full confidence, i.e., $\hat{p} = 1$ for positives and $\hat{p} = 0$ for negatives. However, these measures behave differently with respect to over- and under-confidence. Brier score is symmetric in the sense that over- and under-estimating the calibrated probability to be positive by the same amount results in the same loss according to Brier score. In contrast, log-loss is highly sensitive to over-confidence, particularly at the high confidence cases. As an extreme case, full confidence in the wrong prediction yields infinite log-loss. Even if this happens only with one instance in the test set, the overall loss on the whole test set is still infinite due to averaging. Exactly this can happen often

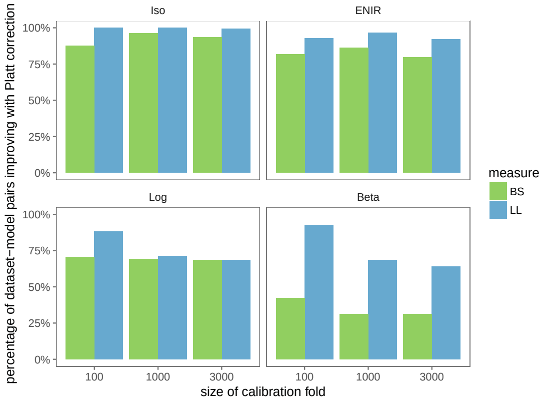

Fig. 2. Percentage of 153 dataset-model pairs where Platt correction improves over the uncorrected calibration method for Brier score (BS) and log-loss (LL).

with isotonic calibration and ENIR, whenever the lowest score in the training set has a negative class and/or highest score has positive class. While penalized infinitely by log-loss, any other proper loss would also penalize this.

3 Simple Improvement of Existing Methods

This motivates our first contribution: a simple improvement of isotonic calibration and ENIR. On these calibration methods we propose to use the same correction procedure as Platt used for logistic calibration. This means that isotonic calibration and ENIR should also be applied after replacing the class labels 1 and 0 by $1 - \frac{1}{N_{+}+2}$ and $0 + \frac{1}{N_{-}+2}$, respectively, where N_{+} and N_{-} are the numbers of positives and negatives in the training data.

We have evaluated this simple modification on $459 = 9 \times 17 \times 3$ calibration tasks, obtained by training 9 different models on 17 datasets and in each using either 100, 1000 or 3000 instances for learning the calibration map (see details about the experimental setup in Section 5.2). In 458 cases out of 459 log-loss was reduced when starting to use Platt's correction on isotonic calibration (Figure 2 top left, where the 459 cases are split between calibration sizes 100, 300, and 1000). The benefit is also obvious for Brier score, with improvement in 92% of the cases (424 out of 459). For reference, Figure 2 also shows the impact of Platt's correction on logistic and beta calibration methods. For logistic calibration the results confirm the benefit of Platt's correction, as expected. For beta calibration the correction turns out to be useful only for log-loss, and not for Brier score (improvement in ¡50% of cases).

Isotonic calibration can suffer infinite log-loss due to over-confidence on instances at either end of the ranking by score. To understand the effect of Platt's correction on over- and under-confidence a bit better, we performed the following analysis. We considered the first and last 2.5% of the instances according to

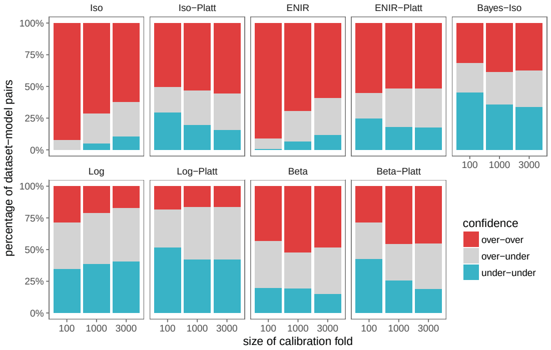

Fig. 3. Proportion of under- and over-confidence for 9 calibration methods on 153 data-model pairs over 3 different calibration data sizes.

the ranking by score. We say that a calibration method is over-confident on the last 2.5% instances, if the average calibrated probability on these instances is closer to 1 than the actual proportion of positives. Otherwise, we call it under-confident. Note that here we do not have a zone of being calibrated between over- and under-confidence, because we are interested in seeing the changes in over- and under-confidence after Platt correction. Similarly, we say that a calibration method is over-confident on the first 2.5% instances, if the average calibrated probability according to this method on these instances is closer to 0 than the actual proportion of positives (because here the model is over-confident in predicting the negative class).

Figure 3 shows the proportions of cases where the calibration method is over-confident at both ends (over-over), under-confident at both ends (under-under) or over-confident at one end and under-confident at the other (over-under). As expected, Platt correction reduces the proportion of over-over and increases the proportion of under-under. Overall, the balance between over- and under-confidence varies significantly across different methods. Interestingly, the most equal proportions of over- and under-confidence are shown by Bayes-Iso (non-parametric Bayesian isotonic calibration), which we will next motivate and present.

4 Proposed Method

Even though Platt correction helps to overcome some issues regarding over-confidence, there is no clear justification behind it. In case of fully separable training data where all negative instances have lower scores than positives it can be thought of as performing Laplace smoothing, which is a standard method to estimating class proportions, e.g. within a leaf of a decision tree. Laplace smoothing has a Bayesian interpretation, but this interpretation does not seem to apply to the Platt correction method. Our goal is to propose a fully Bayesian non-parametric calibration method which would perform well on both smaller and larger datasets, as opposed to current non-parametric methods which are outperformed by parametric methods on smaller datasets.

Suppose we have a fixed scoring classifier and we need to learn a calibration map $\widehat{cal}$ from N training instances, given the (uncalibrated) scores $\mathbf{s}^{tr} = (s_1, \dots, s_N)$ predicted by the classifier and the actual labels $\mathbf{y}^{tr} = (y_1, \dots, y_N)$. The calibration map would be evaluated by drawing a random test instance X, applying the classifier to obtain its score $\mathsf{S} = classifier(\mathsf{X})$, and then testing the calibrated probability $\mathsf{C} = \widehat{cal}(\mathsf{S})$ against the actual class Y with respect to a loss measure l by calculating $l(\mathsf{C}, \mathsf{Y})$. If the loss measure is a proper loss, then the expected loss would be minimized by the perfect calibration map cal defined as $cal(\mathsf{S}) = \mathbb{E}[\mathsf{Y}|\mathsf{S}]$. This result follows from the fact that Bregman divergences are minimized at the conditional expectation [1] and the proper losses are Bregman divergences where one of the inputs has been restricted to be binary [17]. Note that the perfectly calibrated probabilities $cal(S)$ are different from the Bayes-optimal probability estimator $\mathbb{E}[\mathsf{Y}|\mathsf{X}]$.

Isotonic calibration aims to find calibrated probability estimates $\hat{\mathbf{c}} = (\hat{c}_1, \dots, \hat{c}_N)$ at the sorted scores $s_1 \leq \dots \leq s_N$ present in the training data, where $\hat{\mathbf{c}}$ must belong to the space $\mathcal{I}^N$ of all real-valued vectors of length N constrained with isotonicity $0 \leq \hat{c}_1 \leq \dots \leq \hat{c}_N \leq 1$. This discrete calibration map can then be extended to $\widehat{cal}$ as a piecewise constant calibration map, or linear interpolation could be used to fill in the gaps between training scores. Since proper losses are minimized at the conditional expectation [1, 17], it is easy to show that due to pooling the isotonic calibration $\widehat{cal}_{iso}$ is minimizing any proper loss l on the training data. This means that $\hat{\mathbf{c}}^{iso} = \arg\min_{\hat{\mathbf{c}} \in \mathcal{I}^N} L(\hat{\mathbf{c}}, \mathbf{y}^{tr})$ where $L(\hat{\mathbf{c}}, \mathbf{y}^{tr}) = \sum_{i=1}^{N} l(\hat{c}_i, y_i^{tr})$.

4.1 Non-parametric Bayesian Isotonic Calibration

Inspired by isotonic calibration, we aim to estimate the calibration map on the predicted scores present in the training data, and elsewhere we would use linear interpolation. While standard isotonic finds the monotonic calibration map which minimises the loss on the training data (in the spirit of maximum likelihood), we aim to minimise the expected loss on future test data (in the spirit of maximum a posteriori). However, to avoid having to define a prior over all possible isotonic calibration maps from $\mathbb{R}$ to $[0, 1]$, we narrow the aim to minimise the expected loss on only those future test data which contain the same scores as our training data. Due to this we only need to define the prior over the N scores present in the training data. As actual test labels are not available during training, then the expected loss on future test data can never be known in practice, but can still be estimated based on the training data. To derive such an estimator, we will reason about the test labels and introduce notation for them. To avoid confusion with the actual test labels, we will be using the term *hypothetical labels* from now on. These hypothetical labels will only be used notationally, for deriving the methods, and these are not needed for running the proposed calibration algorithm.

Following the Bayesian paradigm we assume that the perfect calibration map $\mathbf{C} = (\mathsf{C}_1, \ldots, \mathsf{C}_N)$ was drawn from $\mathcal{I}^N$ according to some prior distribution that we will specify in Section 4.2. We assume that both the training labels $\mathbf{Y}^{tr} = (\mathsf{Y}^{tr}_1, \ldots, \mathsf{Y}^{tr}_N)$ and hypothetical labels $\mathbf{Y}^{hyp} = (\mathsf{Y}^{hyp}_1, \ldots, \mathsf{Y}^{hyp}_N)$ were drawn independently according to the probabilities $\mathbf{C}$, that is $\mathsf{Y}^{tr}_i, \mathsf{Y}^{hyp}_i \sim Bernoulli(\mathsf{C}_i)$ for $i = 1, \ldots, N$. We define non-parametric Bayesian isotonic calibration as follows:

$$\hat{\mathbf{c}}^{Bayes-iso} = \underset{\hat{\mathbf{c}} \in \mathcal{I}^N}{\arg\min}\, \mathbb{E}\left[L(\hat{\mathbf{c}}, \mathbf{Y}^{hyp}) \mid \mathbf{Y}^{tr} = \mathbf{y}^{tr}\right] \tag{1}$$

where $L(\hat{\mathbf{c}}, \mathbf{y}) = \sum_{i=1}^{N} l(\hat{c}_i, y_i)$.

The following Theorem 1 will form the basis for calculating this conditional expectation numerically. It proves that the conditional expectation of Eq.(1) can be calculated as a ratio of two unconditional expectations involving the calibration map $\mathbf{C}$ and its likelihood under the observed training data, $\mathbb{P}(\mathbf{Y}^{tr} = \mathbf{y}^{tr} \mid \mathbf{C})$. This result can be thought of as Bayesian model averaging: models are sampled from the model prior and averaged weighting by their likelihoods.

Theorem 1. *Let $\mathbf{C}$, $\mathbf{Y}^{tr}$ and $\mathbf{Y}^{hyp}$ be random vectors of length N as defined above. Suppose we observe $\mathbf{Y}^{tr} = \mathbf{y}^{tr}$, then for $\hat{\mathbf{c}}^{Bayes-iso}$ as defined in Eq.(1) the following holds:*

$$\hat{\mathbf{c}}^{Bayes-iso} = \frac{\mathbb{E}\left[\mathbf{C} \cdot \mathbb{P}(\mathbf{Y}^{tr} = \mathbf{y}^{tr} \mid \mathbf{C})\right]}{\mathbb{E}\left[\mathbb{P}(\mathbf{Y}^{tr} = \mathbf{y}^{tr} \mid \mathbf{C})\right]} \tag{2}$$

$$\textit{where} \quad \mathbb{P}(\mathbf{Y}^{tr} = \mathbf{y}^{tr} \mid \mathbf{C}) = \prod_{\substack{i=1 \\ y_i^{tr}=1}}^{N} \mathsf{C}_i \prod_{\substack{i=1 \\ y_i^{tr}=0}}^{N} (1 - \mathsf{C}_i) \tag{3}$$

Proof. Since proper losses are minimized at the conditional expectation [1, 17], we have $\hat{\mathbf{c}}^{Bayes-iso} = \mathbb{E}\left[\mathbf{Y}^{hyp} \mid \mathbf{Y}^{tr} = \mathbf{y}^{tr}\right]$. According to the law of iterated expectations this is equal to $\mathbb{E}\left[\mathbb{E}\left[\mathbf{Y}^{hyp} \mid \mathbf{C}\right] \mid \mathbf{Y}^{tr} = \mathbf{y}^{tr}\right]$ which simplifies into $\mathbb{E}\left[\mathbf{C} \mid \mathbf{Y}^{tr} = \mathbf{y}^{tr}\right]$ as the components in random binary vector $\mathbf{Y}^{hyp}$ have been drawn according to probabilities in random vector $\mathbf{C}$. From the definition of conditional expectation and Bayes formula we get:

$$\mathbb{E}\left[\mathbf{C} \mid \mathbf{Y}^{tr} = \mathbf{y}^{tr}\right] = \int \mathbf{C}\, f_{\mathbf{C}|\mathbf{Y}^{tr}}(\mathbf{C}, \mathbf{y}^{tr})\, d\mathbf{C} =$$

$$\int \mathbf{C}\, \frac{\mathbb{P}(\mathbf{Y}^{tr} = \mathbf{y}^{tr} | \mathbf{C}) f_{\mathbf{C}}(\mathbf{C})}{\mathbb{P}(\mathbf{Y}^{tr} = \mathbf{y}^{tr})}\, d\mathbf{C} = \frac{\mathbb{E}\left[\mathbf{C}\; \mathbb{P}(\mathbf{Y}^{tr} = \mathbf{y}^{tr} | \mathbf{C})\right]}{\mathbb{P}(\mathbf{Y}^{tr} = \mathbf{y}^{tr})}$$

Eq.(2) follows from this using the law of iterated expectations and the fact that for binary variables the expectations are probabilities. Finally, the calculation of likelihood in Eq.(3) is straightforward, due to independence of the components within the binary vector.

Our proposed non-parametric Bayesian isotonic calibration maps can be calculated by drawing many isotonic calibration maps from the prior distribution, calculating their likelihoods according to the training labels, and using these as weights in averaging all the sampled maps into one final result as is described in Algorithm 1. The algorithm returns a calibration map that is constructed from pairs of scores and calibrated probabilities, which are joined by linear interpolation as in isotonic calibration, to make predictions over all possible scores. Algorithm description mentions bounds which will be explained in Section 4.3. The time complexity of this algorithm is $\mathbb{O}(sn)$ where n is the size of calibration data and s is the number of candidate maps to be sampled from the prior.

```
Data: scores, labels, nrSamples
Result: calibration map
1. Calculate lower and upper bounds from labels
2. Generate nrSamples sample maps from prior with bounds
3. Evaluate the likelihood of each sample according to labels as shown in
   Eq.(3)
4. Calculate weighted average of sampled maps using likelihoods as
   weights
5. Compose the calibration map by joining the scores and the weighted
   average of the sample maps by linear interpolation
```

Algorithm 1: Bayes-Iso algorithm.

4.2 Selecting the Prior over Isotonic Maps

To fully specify our calibration method we must specify the prior distribution over the calibration maps in space $\mathcal{I}^N$. It is crucial to choose a prior which assigns a reasonably high probability density to all calibration maps that we deem reasonable, otherwise the method would never output such maps, even if made likely by the data.

One possible simple prior can be defined as sampling N independent values uniformly from $[0, 1]$ and sorting them to obtain an isotonic calibration map belonging to $\mathcal{I}^N$. However, this prior is highly concentrated around the calibration map where the values $\mathsf{C}_1, \dots, \mathsf{C}_N$ are equally spaced, represented as the diagonal in Figure 4A. Note that in this figure the X-axis represents relative ranks of scores rather than absolute scores coming out from the classifier. Concentration of probability mass around the diagonal implies that any calibration map that is not around the diagonal would be almost impossible to learn. However, in practice the true calibration map can be far from the diagonal, particularly if the classes are imbalanced.

Therefore, we need a prior that covers the space of all isotonic calibration maps more broadly. We have considered the existing priors on Bayesian isotonic regression (not restricted to output in the range $[0, 1]$) [13] but these do not adapt easily to our situation or do not provide broad coverage of the space of all isotonic calibration maps. Our proposed solution to achieve broad coverage is straightforward - while drawing a calibration map from our prior we first pick uniformly randomly a point in the 2-dimensional space of Figure 4 and then start to construct a map that goes through this chosen point. Note that we use the discrete uniform distribution for X-axis (because these are ranks $1, \ldots, N$) and continuous uniform distribution for Y-axis (because these are probabilities). In the next steps we apply the same procedure recursively, while ensuring isotonicity. This means that we next choose the second point uniformly randomly to the left and below from the first point and the third point uniformly randomly to the right and above from the first point. For example, if the first point is (x_1, y_1), then the second point (x_2, y_2) is chosen by sampling x_2 uniformly from $\{1, 2, \ldots, x_1 - 1\}$, and y_2 uniformly from $[0, y_1]$. Similarly, (x_3, y_3) is chosen by sampling x_3 uniformly from $\{x_1 + 1, x_1 + 2, \ldots, N\}$, and y_3 uniformly from $[y_1, 1]$. This procedure recursively delves into all ranges between existing points, until all points $1, \ldots, N$ on the X-axis have been chosen. Figure 4B shows a random sample of 200 calibration maps drawn from this prior for $N = 100$. Note that we have renormalised the X-axis to be from 0 to 1 instead of from 1 to 100.

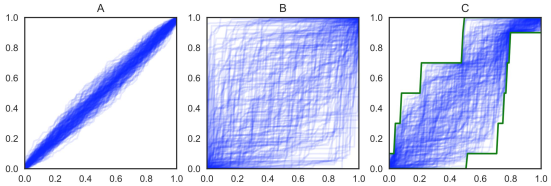

Fig. 4. Examples of 200 sampled curves of size 100. (A) Samples from a bad prior. (B) Samples from our defined prior. (C) Samples from our defined prior using bounds.

4.3 Practically Efficient Sampling from Prior

Having defined the prior we have fully specified our non-parametric Bayesian isotonic calibration method. However, straightforward implementation of this would result in poor performance. The reason is that the space of isotonic calibration maps $\mathcal{I}^N$ is vast and maps with the highest likelihoods are hardly ever found when randomly sampling from the prior. As a result, the estimation of $\mathbb{E}\left[\mathbf{C} \cdot \mathbb{P}(\mathbf{Y}^{tr} = \mathbf{y}^{tr} \mid \mathbf{C})\right]$ would mostly be based on maps C with low likelihood and numerically dominated by very few maps with higher likelihood, resulting in a high variance estimate that would not be precise enough. If we can avoid sampling maps that have near-zero likelihoods, then the estimate stabilises, while still being a good approximation of the true posterior mean map. Therefore, we propose a method to use training data to obtain a lower and upper bound and to sample only those calibration maps that are fully between them. This does change our prior and in this sense is not purely Bayesian, but in practice it provides a reasonably good estimate of the posterior mean with the original prior.

Our algorithm is inspired by calibration methods that use binning. Let us consider a bin of B consecutive instances with labels $y_{j+1}, y_{j+2}, \ldots, y_{j+B}$ within a full ranked list of N training instances. If the proportion of positives in this bin is p, then this can be used as an estimate for the average calibrated probability within this bin, that is $\overline{C} = \frac{1}{B}\sum_{i=1}^{B} C_{j+i} \approx p$. However, since the calibration is isotonic, we know that $C_{j+1} \leq \overline{C} \leq C_{j+B}$. Hence, we can use p as an approximate upper bound for C_{j+1} and an approximate lower bound for C_{j+B}. Taking into account that the estimation of the proportion of positives has variance in the order of $1/\sqrt{B}$, we use in practice the bounds $C_{j+1} \leq p+1/\sqrt{B}$ and $C_{j+B} \geq p-1/\sqrt{B}$.

The above shows how a bin can be used to set bounds for the lower and upper end of the bin. In order to obtain bounds for the calibrated probability at a given test instance we apply the above reasoning on the bins of size B to the left and to the right of this instance within the ranking. If the considered instance is close to one end of the full ranking, then of course the size of the bin towards that end would necessarily be smaller. In the experiments we used the bin size $B = N/10$. The advantage of a larger bin is that p can be approximated more precisely, but at the same time the average is taken over a region where the calibrated probability within the ranking is varying more, so there is a tradeoff in selecting the size of B.

This method results in non-monotonic bounds: for $s_i < s_{i+1}$ the lower bound at s_i could be higher than at s_{i+1}. In such cases we extended the lower bound to monotonicity, that is s_i would adopt the lower bound from s_{i+1}. Symmetrically, the same can happen with upper bounds: for $s_i < s_{i+1}$ the upper bound at s_i could be higher than at s_{i+1}. In this case we raise the upper bound of s_{i+1} to match the upper bound of s_i. By ensuring monotonicity this way the bounds can only become wider, lower bounds can only be lowered and upper bounds raised.

One possibility to apply the bounds on the sampling is to perform rejection sampling - the drawn calibration maps which fall out of bounds would be discarded. However, this can make the sampling very slow, as with tight bounds most of the maps would be discarded. Fortunately, it is easy to modify our prior slightly to be easily directly sampled from between the bounds. After drawing the X-axis value from the discrete uniform distribution we draw the Y-axis value from the uniform distribution between the bounds, rather than between 0 and 1. Similarly, we can at each step sample along the X-axis first, and then sample along the Y-axis uniformly, constrained between the bounds. An example of sampling from between bounds can be seen in Figure 4C. Note that the bounds shown are learned from the actual training labels in an example dataset, which is why they are not symmetric. They are shown to illustrate the idea, in reality the bounds will be always different for different datasets.

5 Experiments

We start the experiments with a case study on a synthetic dataset, in order to demonstrate empirically how our proposed Bayesian isotonic calibration converges to the true perfect calibration map as the dataset size increases, outperforming all state-of-the-art calibration methods. More precisely, we will demonstrate how Bayes-Iso works in the setting that it is designed for. This is followed by a large-scale study on real datasets, illustrating which calibration methods work well when calibration data size is changed. We will see that based on average ranks over all dataset-model pairs Bayes-Iso performs either best or tied with best for all considered training set sizes and loss measures.

5.1 Experiments on Synthetic Data

Bayes-Iso is designed to be better whenever the true calibration function is not in the families of parametric methods. In such cases parametric methods perform poorly due to model mismatch and the existing non-parametric methods due to over-confidence. We will demonstrate this effect on a synthetic dataset. We have generated a dataset where the calibration map does not belong to the logistic and beta calibration map families, because in case of parametric shapes it would be clear that parametric methods would be the best choice. According to our generative model the classes are balanced, and a hypothetical scoring classifier is generating scores that are on actual negatives distributed as $Beta(1,3)$, and on positives as a balanced mixture of $Beta(1.5,3)$ and $Beta(30,3)$. The perfect calibration map is shown in Figure 5 with a red dashed line and on our generated test data with 100000 instances results with ideal log-loss of 0.1620 and Brier score of 0.4741. Table 1 shows how close to the ideal each of the calibration methods reaches on training set sizes 100 and 3000 (on size 1000 methods ranked identically to 3000, not shown). Results were averaged over 10 replicate experiments. Note that according to the results in Section 3, we applied Platt correction on all reference methods, except for beta calibration with Brier score. Bayes-Iso algorithm used 10000 samples to estimate the calibration map.

Results in Table 1 show that Bayes-Iso gets very close to the ideal, winning over all other methods. Even though the true calibration map is not in the parametric family, Beta calibration gets close enough shape to be the second best on the smallest dataset. This example demonstrates that existing parametric methods

Table 1. Average Brier score and log-loss on synthetic datasets of sizes 100 and 3000. Beta calibration is used for Brier score and Beta-Platt for log-loss. Numbers in subscript show the ranking of the scores.

Method	BS100	LL100	BS3000	LL3000
Ideal	.1621	.4741	.1621	.4741
Bayes-Iso	$.1655_1$	$.4878_1$	$.1625_1$	$.4753_1$
ENIR-Platt	$.1683_3$	$.5029_3$	$.1627_2$	$.4778_2$
Iso-Platt	$.1685_4$	$.5036_4$	$.1627_3$	$.4779_3$
Beta(-Platt)	$.1672_2$	$.4895_2$	$.1660_4$	$.4862_4$
Log-Platt	$.1720_5$	$.5112_5$	$.1721_5$	$.5097_5$

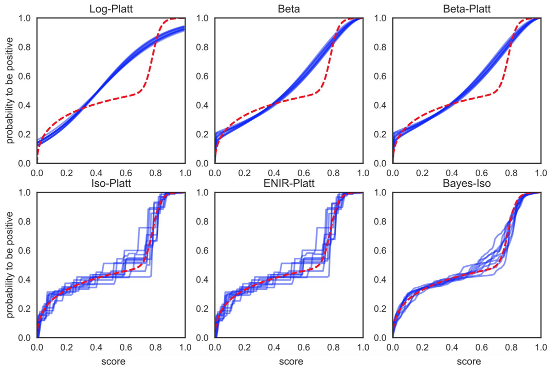

Fig. 5. 10 calibration maps learned on 10 replicate synthetic datasets of size 1000 for six different calibration methods (blue). True underlying calibration map (red).

are often better than non-parametric ones on smaller datasets, because they don't overfit to small data as easily. Bayes-Iso on the other hand is less-confident than other non-parametric methods and works well also on small datasets. On bigger datasets non-parametric methods dominate over parametric ones as expected, and Bayes-Iso shows the best results. Figure 5 demonstrates the variance of all considered calibration methods across the 10 replicate experiments on training set size 1000. We can see that for size 1000 parametric methods clearly cannot learn the true calibration function whereas non-parametric methods can.

Since Bayes-Iso is a non-deterministic method its results can vary on the same dataset across different runs. Figure 6 shows results on 10 runs on exactly the same dataset on each of the 3 data sizes, complemented with bounds as learned within the Bayes-Iso method. The figure demonstrates that each of the runs results in a high-quality calibration map with very low variance across runs. But we can also notice that the larger the calibration data, the more differences the learned maps start to have. This is expected as we need more and more sampling to converge with Bayes-Iso in case of larger data.

5.2 Experimental Setup on Real Data

The methods are evaluated on the following 17 datasets from OpenML [19]: SEA(50), BNG(breast-w), BNG(sonar), BNG(heart-statlog), 2dplanes, house_16H, cal_housing, houses, house_8L, fried, letter, BNG(spectf_test), BNG(Australian), BNG(SPECTF), skin-segmentation, creditcard, numerai28.6. These were selected as datasets with a binary target variable, no missing values, at most 100 numerical features, and the number of instances between 20000 and 1 million.

Performance of calibration methods is known to vary with dataset size [14]. We decided that we can see size-related effects best if we fix particular sizes (100, 1000 and 3000) for the fold on which we apply the calibration method. To make the losses on different sizes directly comparable we further chose to keep

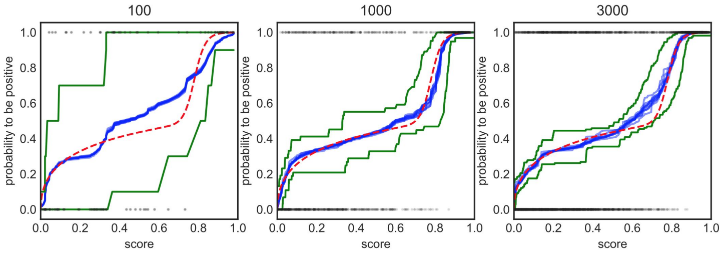

Fig. 6. 10 calibration maps for different sized data learned with Bayes-Iso on the same dataset (blue). Green lines show the lower and upper bounds for sampling, red line shows the true underlying calibration map.

the classifier and the test set constant. We achieved all this by first randomly downsampling all datasets to the same size of 20000 instances, and then running 5-fold nested cross validation. In the internal 5-fold cross-validation we use 4 folds to train the model and 1 fold to calibrate. This 1 internal fold was big enough (3200) to allow randomly sampling calibration datasets of required 3 sizes. The goodness of the calibration maps are evaluated on the outer fold, that we call the test fold, which is of size 4000. To make experiments run faster we have trained the classification models on 3000 out of the 12800 instances of the 4 internal folds. This choice certainly makes the models weaker but still allows to achieve our objective of comparing calibration methods. The classification models were trained with 9 different learning algorithms, selected from among the same as used in the large-scale comparisons in [14] and [2]: decision tree (DT), naive bayes (NB), support vector machine (SVM), random forest (RF), logistic regression (LR), K-nearest neighbors (KNN), boosted trees (ADA), bagged trees (BAG-DT) and artificial neural networks (ANN). The implementations for these algorithms were taken from the scikit-learn package [15] using the default parameters, except for the decision tree, for which we used minimum leaf size of 10.

Overall, we trained a classifier for each of the $17 \times 9 \times 5 \times 5 = 3825$ combinations of 17 datasets, 9 classifier learning algorithms, 5 external and 5 internal cross-validation folds. For each trained classifier we learned $3 \times 9 = 27$ calibration maps resulting from 3 dataset sizes and 9 calibration algorithms (logistic, beta, isotonic calibration and ENIR with and without Platt correction, and Bayes-Iso).

We used existing packages for Beta calibration and ENIR, and modified scikit-learn implementation for logistic calibration (to switch off Platt correction). Other methods were implemented from scratch. [1]

5.3 Experiment Results on Real Data

First, we evaluated Bayes-Iso against other non-parametric methods (that are Platt corrected). Table 2 shows the percentage of dataset-model pairs where

[1] Code with implementations of the algorithms and experiments on real data is available at `https://github.com/mlkruup/bayesiso`.

Bayes-Iso outperformed both Iso-Platt and ENIR-Platt, across different sizes of calibration datasets. Bayes-Iso was the best non-parametric method on the majority of cases, in particular on smaller sizes. This is expected as isotonic calibration and ENIR are known to be overfitting on smaller datasets but more suitable on larger ones, where they become more competitive to Bayes-Iso.

Increase in dataset size leads to Bayes-Iso sampling the space of isotonic maps more sparsely, and more often a single map dominates all others within the sample, in the sense that its likelihood is higher than all others summed up. This can be used as an indicator flag of potential poor performance. The column 3000 LH in Table 2 shows results where the flagged cases (27% of all cases) have been eliminated. The improvement from 56% and 59% in column 3000 to 71% and 73% in column 3000 LH means that there is a big potential in improving our method further by more efficient bounds and more sampling. It is also comforting that Bayes-Iso can *itself* flag cases of potential instability.

Secondly, we wanted to compare all state-of-the-art calibration methods, including the parametric ones, to Bayes-Iso. We have an initial hypothesis that Bayes-Iso should perform well both on larger and smaller datasets whereas parametric methods work better on smaller and other non-parametric methods on larger datasets. We demonstrate this in a large-scale comparison against all considered calibration methods. We performed Friedman test with post hoc analysis on average ranks [4] of models ordered by log-loss and Brier score. The results are illustrated as critical difference diagrams in Figure 7. We can see that Bayes-Iso performs either best or tied with the best, based on the average ranks across all dataset-model pairs. This holds true for all sizes of the calibration set (100, 1000, 3000) and both loss measures (BS, LL). This supports our hypothesis about the behaviour of the methods with respect to the calibration set sizes.

It is not easy to give recommendations for the most suitable calibration method for different models since good performance for a calibration method is more dependent on the dataset size and how a particular model is performing on a dataset. Factors like calibration data size, goodness of the model, distribution of scores in the classes, class distribution, shape of the true calibration map are probably more important factors and most likely have joint effects when deciding on the best method to use. We have found some examples about how these factors affect the performance of Bayes-Iso. One discovered case is when we have a small dataset and a model with very high accuracy. In this case Bayes-Iso is too

Table 2. Percentage of improved dataset-model pairs where Bayes-Iso improved on other non-parametric methods.

Size	100	1000	3000	3000 LH
BS	86%	79%	56%	71%
LL	92%	84%	59%	73%

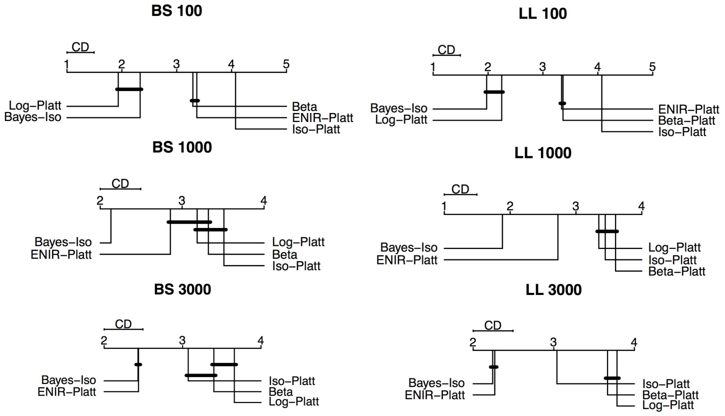

Fig. 7. Critical difference diagrams based on ranks of methods over 153 dataset-model pairs over different calibration dataset sizes and losses.

under-confident when compared to ENIR-Platt and Iso-Platt. The reason could be that since the model is very good then even with small dataset for calibration it is beneficial to have high confidence predictions. Because of the joint effects of the formerly mentioned factors, these patterns are difficult to identify and interpret. Extensive experiments left for future work could give us more insight into these effects and help us identify situations where one or another calibration method is the most suitable.

6 Conclusions

For decision-making purposes it is important that the classifiers were well-calibrated. Parametric calibration methods work well on small datasets, but on bigger datasets the parametric assumption often does not hold and non-parametric methods perform better. In this work we have first demonstrated that existing non-parametric calibration methods produce over-confident predictions. We have discovered that the same correction method that was used in logistic calibration by Platt can be used for reducing over-confidence in isotonic calibration and ENIR, reducing log-loss and Brier score in more than 90% of our studied cases. Our main contribution is a novel non-parametric Bayesian isotonic calibration (Bayes-Iso). Bayes-Iso has the flexibility of isotonic calibration to fit maps of all monotonic shapes but it additionally provides smoothness and reduces over-confidence without requiring a separate correction procedure. When comparing against the state-of-the-art methods on 153 calibration tasks Bayes-Iso works either best or tied with the best depending on the size of the calibration dataset. The current version of Bayes-Iso experiences instability when scaling up to learn a calibration map from many more than 3000 instances. As future work we envision ways to make Bayes-Iso scale up to much larger sizes, as the calibration map could easily be learned in bins of 1000 consecutively ranked instances and later merged into a single calibration map.

Acknowledgments

This work was supported by the Estonian Research Council under grant PUT1458.

References

1. Banerjee, A., Guo, X., Wang, H.: On the optimality of conditional expectation as a Bregman predictor. IEEE Transactions on Information Theory **51**(7), 2664–2669 (2005)
2. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 161–168. ACM (2006)
3. DeGroot, M.H., Fienberg, S.E.: The comparison and evaluation of forecasters. The Statistician pp. 12–22 (1983)
4. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7**(Jan), 1–30 (2006)
5. Elkan, C.: The foundations of cost-sensitive learning. In: International Joint Conference on Artificial Intelligence. vol. 17, pp. 973–978. Lawrence Erlbaum Associates Ltd (2001)
6. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
7. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning. vol. 70, pp. 1321–1330. JMLR (2017)
8. Kull, M., De Menezes E Silva Filho, T., Flach, P.: Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers, pp. 623–631. JMLR (4 2017)
9. Kull, M., Flach, P.: Novel decompositions of proper scoring rules for classification: score adjustment as precursor to calibration. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 68–85. Springer (2015)
10. Naeini, M.P., Cooper, G.F.: Binary classifier calibration using an ensemble of near isotonic regression models. In: IEEE 16th International Conference on Data Mining. pp. 360–369. IEEE (2016)
11. Naeini, M.P., Cooper, G.F., Hauskrecht, M.: Binary classifier calibration using a Bayesian non-parametric approach. In: Proceedings of the 2015 SIAM International Conference on Data Mining. pp. 208–216. SIAM (2015)
12. Naeini, M.P., Cooper, G.F., Hauskrecht, M.: Obtaining well calibrated probabilities using Bayesian binning. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence. pp. 2901–2907. AAAI Press (2015)
13. Neelon, B., Dunson, D.B.: Bayesian isotonic regression and trend analysis. Biometrics **60**(2), 398–406 (2004)
14. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: Proceedings of the 22nd International Conference on Machine Learning. pp. 625–632. ICML '05, ACM (2005)

15. Pedregosa, et al.: Scikit-learn: Machine learning in Python. JMLR **12**, 2825–2830 (2011)
16. Platt, J., et al.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in Large Margin Classifiers **10**(3), 61–74 (1999)
17. Reid, M.D., Williamson, R.C.: Information, divergence and risk for binary experiments. Journal of Machine Learning Research **12**(Mar), 731–817 (2011)
18. Tibshirani, R.J., Hoefling, H., Tibshirani, R.: Nearly-isotonic regression. Technometrics **53**(1), 54–61 (2011)
19. Vanschoren, J., Van Rijn, J.N., Bischl, B., Torgo, L.: OpenML: networked science in machine learning. ACM SIGKDD Explorations Newsletter **15**(2), 49–60 (2014)
20. Zadrozny, B., Elkan, C.: Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In: ICML. vol. 1, pp. 609–616. Citeseer (2001)
21. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 694–699. ACM (2002)