OBTAINING ACCURATE PROBABILITIES USING CLASSIFIER CALIBRATION

by

Mahdi Pakdaman Naeini

B.Sc. in Software Engineering, Iran University of Science and Technology, 2001

M.Sc. in Artificial Intelligence and Robotics, University of Tehran, 2004

M.Sc. in Intelligent Systems Program, University of Pittsburgh, 2013

Submitted to the Graduate Faculty of

the Kenneth P. Dietrich School of

Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

KENNETH P. DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Mahdi Pakdaman Naeini

It was defended on

August 5th 2016

Dr. Gregory F. Cooper, Department of Biomedical Informatics, University of Pittsburgh

Dr. Milos Hauskrecht, Department of Computer Science, University of Pittsburgh

Dr. Shyam Visweswaran, Department of Biomedical Informatics, University of Pittsburgh

Dr. Jeff Schneider, The Robotics Institute, Carnegie Mellon University

Dissertation Director: Dr. Gregory F. Cooper, Department of Biomedical Informatics, University of Pittsburgh

OBTAINING ACCURATE PROBABILITIES USING CLASSIFIER CALIBRATION

Mahdi Pakdaman Naeini, PhD

University of Pittsburgh, 2016

Learning probabilistic classification and prediction models that generate accurate probabilities is essential in many prediction and decision-making tasks in machine learning and data mining. One way to achieve this goal is to post-process the output of classification models to obtain more accurate probabilities. These post-processing methods are often referred to as calibration methods in the machine learning literature.

This thesis describes a suite of parametric and non-parametric methods for calibrating the output of classification and prediction models. In order to evaluate the calibration performance of a classifier, we introduce two new calibration measures that are intuitive statistics of the calibration curves. We present extensive experimental results on both simulated and real datasets to evaluate the performance of the proposed methods compared with commonly used calibration methods in the literature. In particular, in terms of binary classifier calibration, our experimental results show that the proposed methods are able to improve the calibration power of classifiers while retaining their discrimination performance. Our theoretical findings show that by using a simple non-parametric calibration method, it is possible to improve the calibration performance of a classifier without sacrificing discrimination capability. The methods are also computationally tractable for large-scale datasets as they run in $O(N \log N)$ time, where N is the number of samples.

In this thesis we also introduce a novel framework to derive calibrated probabilities of causal relationships from observational data. The framework consists of three main components: (1) an approximate method for generating initial probability estimates of the edge types for each pair of variables, (2) the availability of a relatively small number of the causal relationships in the network for which the truth status is known, which we call a *calibration training set*, and (3) a

calibration method for using the approximate probability estimates and the calibration training set to generate calibrated probabilities for the many remaining pairs of variables. Our experiments on a range of simulated data support that the proposed approach improves the calibration of edge predictions. The results also support that the approach often improves the precision and recall of those predictions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

ACKNOWLEDGEMENT

First and for most, I would like to express my sincere gratitude to my advisor Greg Cooper for his consistent support, encouragement, and invaluable advice during my graduate studies at the University of Pittsburgh. I was very fortunate to work with such a considerate, wise, and knowledgeable professor in my graduate studies.

I am of course very grateful to my committee members Milos Hauskrecht, Shyam Visweswaran, and Jeff Schneider for their insightful comments on my dissertation along the way. Especially, I would like to thank Milos for his advice and fruitful discussions that we had during my Ph.D. studies.

I would like also to thank the directors of the Intelligent Systems Program (ISP), Jan Wiebe and Diane Litman for their selfless and consistent efforts in managing our program. I am also grateful to my ISP fellows including Fattaneh Jabbari, Jeya Balasubramanian, Cem Akkaya, Matthias Grabmair, Saeed Amizadeh, Gaurav Trivedi, Daniel Steinberg, Zahra Rahimi, Amin Tajgardoon. Also, I would like to express my gratitude to the considerate and kind administrators of ISP, Wendy Bergstein and Michele Thomas, for providing solutions for every problem that I brought to them and making administrivia easy.

I am deeply grateful to all of my teachers and friends a long the journey that leads me to coming to the University of Pittsburgh. I am thankful to all the professors that supported me at my graduate and undergraduate studies including Caro Lucas, Ruzbeh Tusserkani, Mahdi Sadeghi, Behzad Moshiri, and Mohsen Sharifi. Especially, I would like to express my gratitude to Ruzbeh Tusserkani for playing an inspiring and influential role in my life. I am thankful to all my teachers in the middle school and the high school including Davood Masoumi, Moharram Iradmousa, Ali Nour-Mohammadi, Heidar Zandiyeh, Ali Afkhami, Mr. Amirian, Mr. Darvish, Ali Nejati, and Mr. Ghaffari among the others. I am also thankful to my dear friends Ali Baradaran Hashemi, Alireza

Keshavarz-Haddad, and Kaveh Kavousi for their inspiration and advice.

I am wholeheartedly grateful to my parents Fatemeh and Nemat for their unconditional love and support. They have been always symbol of patience and strength all through my studies and life. I am also very grateful to my sisters Fereshteh and Maryam for their love and support and for all the joy they added to my life. I am thankful to Narges and Ahmed for welcoming me into their family and for believing in me ever since they first met me. I am thankful to FaceTime, Skype, Hangout, and ooVoo for letting me to be in contact with my beloved ones during these 6 years while I am far far away from them. They let me to feel the aging of my parents, the growth of my beloved nephew Saeed, and the beautiful smile of my little niece Melika after her birth.

Finally, and most importantly, I wish to express my deepest love and gratitude to Huma, the greatest blessing in my life. Thanks for your constant friendship and love, through all the stages of the journey that we have started together 12 years ago. For staying at the office together many weekends and evenings working together. For cheering me up at times of difficulty and anguish with your beautiful smile and peaceful comments on our life.

1.0 INTRODUCTION

Obtaining accurate probabilities is crucial in many real-world decision-making and data mining problems. Decision theory provides a rationale for intelligent agents to make decisions (Russell and Norvig, 2009) in which the utilities and probabilities are combined in determining the actions that maximize expected utility. In many of decision problems, the probabilities need to be well-calibrated in order to achieve this goal of finding the best action. The predicted probabilities of a forecaster are well-calibrated if they are close to the objective probabilities (i.e., the frequency of the events in the long run). More specifically, we say that a classification model is well-calibrated if events predicted to occur with probability p do occur about p fraction of the time for all p. This concept applies to binary as well as multi-class classification problems. Figure 1 illustrates the binary class calibration problem using a reliability curve (DeGroot and Fienberg, 1983; Niculescu-Mizil and Caruana, 2005b). The curve shows the probability predicted by the classification model versus the actual fraction of positive outcomes for a hypothetical binary classification problem, where Z is the binary event being predicted. The curve shows that when the model predicts $Z = 1$ to have probability 0.2, the outcome $Z = 1$ occurs in about 0.3 fraction of the time. The model is fairly well-calibrated, but it tends to underestimate the actual probabilities. In general, the straight dashed line connecting $(0, 0)$ to $(1, 1)$ represents a perfectly calibrated model. The closer a calibration curve is to this line, the better calibrated is the associated prediction model. Deviations from perfect calibration are very common in practice and may vary widely depending on the binary classification model that is used.

Producing well-calibrated probabilistic predictions is critical in many areas of science (e.g., determining which experiments to perform), medicine (e.g., deciding which therapy to give a patient), business (e.g., making investment decisions), and many others. These are only some examples of cost-sensitive decision-making data mining problems, where different instances have different mis-

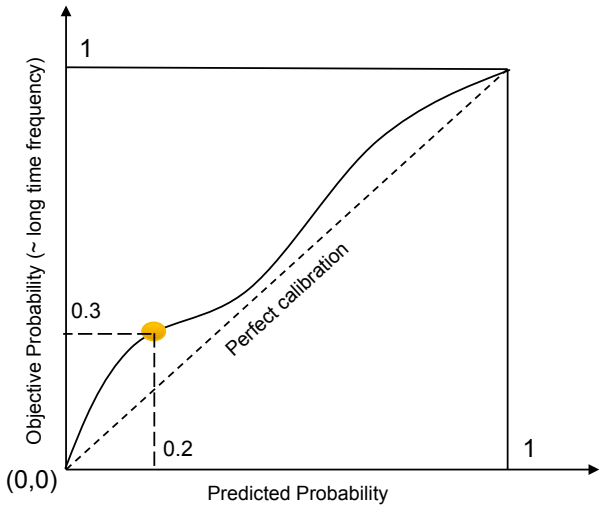


Figure 1: The solid line shows a calibration (reliability) curve for predicting $Z = 1$. The dotted line is the ideal calibration curve.

classification costs that might not be known at the training time. If in this type of problems we seek to maximize the expected utility instead of accuracy, then generating well-calibrated probabilities is critical (Korb and Nicholson, 2010; Fawcett and Niculescu-Mizil, 2007). For instance, using the German credit card UCI dataset (Feng et al., 1993), Zonneveldt et al. evaluated the performance of different types of classifiers in predicting whether a customer will default on a loan (Zonneveldt et al., 2010). Their experiments showed that in terms of predictive accuracy, any decent classifier (including NB, TAN, and augmented TANs) can perform better than the default prediction, i.e., simply granting every loan application. However, no classifier performed significantly better than the default model when evaluated in terms of expected value of the loans whether they use an original cost table or the refined version shown in Table 1. This can be as a result of calibration error of the trained classifiers [1].

In data mining problems, obtaining accurate probabilistic models is also important when comparing and combining the output of different classification models (Bella et al., 2013). For instance, a common way of addressing a multi-class classification problem is to reduce the problem to mul-

[1]Note that, however, the authors did not perform calibration analysis on the trained classifiers in their experiments

Table 1: German Credit Cost Table

	Original Cost		Refined Cost	
	Repay	Default	Repay	Default
Give loan	0	5	-1	10
Refuse loan	1	0	1	0

tiple binary classification problems using one-versus-one or one-versus-all method (Bishop, 2006). The one-versus-all method (a.k.a. one-versus-rest) involves training a single classifier per class, with the samples of that class as positive samples and all other samples as negatives. In the one-versus-one method trains $\frac{K(K-1)}{2}$ binary classifiers for a K-class multiclass problem; each receives the instances of a pair of classes from the original training set, and learn to discriminate the two classes. The scores generated by $\frac{K(K-1)}{2}$ binary classifiers are combined in a voting scheme to decide the final class of an instance. As noted in (Gebel and Weihs, 2007) performing multi-class classification using the one-versus-one or one-versus-all method would not be very accurate unless the scores generated by each binary classifier are comparable. Thus, calibrating the probability scores could potentially give a good basis for building multi-class classification models using binary classification models. Furthermore, building calibrated classification models is also useful when we aim to use the output of a classifier not only to discriminate the instances but also to rank them [2] (Zhang and Su, 2004; Jiang et al., 2005; Hashemi et al., 2010).

Research on learning well-calibrated models has not been explored in the machine learning and data mining literature as extensively as, for example, learning models that have high discrimination (i.e., high accuracy). Generally, there are two main approaches to obtaining well-calibrated classification models. The first approach is to build a classification model that is intrinsically well-calibrated *ab initio*. This approach will restrict the designer of the data mining model by requiring major changes in the objective function (e.g, using a different type of loss function) and could potentially increase the complexity and computational cost of the associated optimization program

[2] Note that, however, calibration is not necessary in order to rank the outputs of a classifier based on the class labels

to learn the model. The other approach is to rely on the existing discriminative data mining models and then calibrate their output using post-processing methods, which are referred to as classifier calibration methods in the literature (Bella et al., 2013). This approach is general, flexible, and it frees the designer of a data mining algorithm from modifying the learning procedure and the associated optimization method. However, this approach also has the potential to decrease discrimination when increasing calibration, if care is not taken.

This dissertation research focuses on post-processing methods to obtain well-calibrated classification models. The calibration methods we describe in this thesis are shown empirically to improve calibration of different types of classifiers (e.g., LR, SVM, and NB) while maintaining their discrimination performance well. Some commonly used post-processing binary classifier calibration methods include Platt scaling (Platt, 1999), histogram binning (Zadrozny and Elkan, 2001b), and isotonic regression (Zadrozny and Elkan, 2002). In all of these methods, the post-processing step can be seen as a function that maps the outputs of a prediction model to probabilities that are intended to be well-calibrated. Figure 1 shows an example of such a mapping.

In general, there are two main applications of post-processing calibration methods. First, they can be used to convert the outputs of discriminative classification methods with no apparent probabilistic interpretation to posterior class probabilities (Platt, 1999). An example is an SVM model that learns a discriminative model without a direct probabilistic interpretation. In this thesis, we show this use of calibration to map SVM outputs to well-calibrated probabilities. Second, calibration methods can be applied to improve the calibration of predictions of a probabilistic model that is miscalibrated. For example, an NB model is a probabilistic model, but its class posteriors are often miscalibrated due to unrealistic independence assumptions (Niculescu-Mizil and Caruana, 2005b). The proposed binary calibration methods we describe in this thesis are shown empirically to improve the calibration of NB models without reducing their discrimination. The proposed method can also work well on models that are less egregiously miscalibrated than are NB models.

1.1 HYPOTHESIS STATEMENT

Despite the importance of having calibrated prediction models, calibration should not be over-emphasized and become the only concern when learning a useful predictor, especially in the finite horizon decision-making problems, as noted first by DeGroot and Fienberg (DeGroot and Fienberg, 1983). To clarify, let us review the simple example presented in the decision theory book of Parmigiani and Inoue (Parmigiani and Inoue, 2009) (Chapter $10.4.1$). Assume a weather forecaster will be only evaluated based on how well-calibrated her predictions are at the end of the year. We take out all the days of the year that she announced the chance of the rain to be 10% and see how close is the empirical frequency to 10%. We repeat the procedure for 20% announced prediction and so on. If we only evaluate the forecaster based on how well-calibrated her predictions were over a finite horizon, then she will make announcements that are radically at odds with her beliefs. For instance, assume towards the end of the year she finds that the empirical frequency of her predictions for 10% days is much lower than 10%. Even if she is completely confident that there is going to be a heavy rain with flood tomorrow, it is still beneficial for her to announce the chance of rain is 10%.

The above example shows that calibration is not a substitute for other evaluation criteria for a classification model such as discrimination measures (e.g., AUC). However, it is beneficial in practice to build classification models that are well-calibrated in addition to being able to discriminate the instances. Our first hypothesis in this thesis is that *by using simple post-processing calibration methods, it is possible to improve the calibration capability of a classifier without sacrificing much discrimination capability*. This will justify the idea of building calibrated models using post-processing methods. The machine learner can focus on building models that discriminate the patterns well. Then he/ she can use post-processing calibration methods to calibrate the model without overfitting or losing too much the discrimination capability of the base classifier.

As we will describe in the Background section, the existing calibration methods use single mappings (either parametric or non-parametric) to obtain calibrated probabilities. As we will discuss in Chapter 2, the current methods are either biased about the output of the base classifier (e.g., isotonic regression-based calibration), or they are not using information that often the uncalibrated scores are generated by a well-performing classifier, in terms of a discrimination measure (e.g.

AUROC). Therefore, our second hypothesis states that *one can systematically produce probabilities that are more accurate than those obtained from existing calibration methods by using an ensemble of calibration models that are generated based on rational, realistic assumptions about the output of the classifier*.

The remainder of this thesis is organized as follows. First, we present background concepts on having calibrated probabilities and review related works and the existing calibration methods in Chapter [2]. Next, we present the statement of the contributions of this thesis on introducing new binary classifier calibration methods in Chapter [3]. In Chapter [4], we outline the results obtained in our experiments on binary classifier calibration problem. Chapter [5] presents our theoretical finding on binary classifier calibration. In Chapter [6] presents our proposed multi-class classifier calibration model as well as its application in a causal network discovery problem. Finally, Chapter [7] concludes the thesis and presents some potential future directions for our research.

2.0 BACKGROUND

In this section, we will present a review of previous methods related to classifier calibration, as well as a description of their theoretical properties. We will also discuss the concept of calibration versus refinement of classifiers and will present the evaluation measures used in this thesis for comparing the classifier calibration methods.

2.1 EXISTING CALIBRATION METHODS

Existing post-processing binary-classifier calibration models generally fall into one of two categories: parametric or non-parametric, and we discuss each in turn

2.1.1 Parametric Calibration Methods

2.1.1.1 Platt's Method Platt's method is a parametric binary classifier calibration method (Platt, 1999). While it was originally developed to transform the output of an SVM model into calibrated probabilities, it has also been used to calibrate other types of classifiers (Niculescu-Mizil and Caruana, 2005b; Gebel and Weihs, 2007; Niculescu-Mizil and Caruana, 2005a). It assumes that the posterior probability of the positive class given the (uncalibrated) classification scores has the form of a sigmoidal function as follows:

$$P(z|y) = \frac{1}{1 + \exp(z(ay + b))}, \tag{2.1}$$

where y is the predicted (uncalibrated) score generated by the classifier, z is the true class of the instance ($z \in \{+1, -1\}$), and a and b are the parameters of the model. This approach is equivalent

to assuming that the log odds of the calibrated scores are linear with respect to the (uncalibrated) classification score: $log\frac{P(Z=1|y)}{P(Z=0|y)} = ax + b$

In order to find the model parameters, Platt used a model-trust minimization algorithm (Gill et al., 1981) in a maximum likelihood framework. The method runs in $O(1)$ at test time, and thus, it is fast. Its key disadvantage is the restrictive shape of sigmoid function that rarely fits the true distribution of the predictions (Jiang et al., 2012).

2.1.1.2 Beta Distribution In order to estimate a calibrated score, one can use Bayes rule as:

$$P(Z = 1|y) = \frac{p(y|Z = 1)P(Z = 1)}{p(y|Z = 0)P(Z = 0) + p(y|Z = 1)P(Z = 1)}, \tag{2.2}$$

where $p(y|Z = 1)$ and $p(y|Z = 0)$ are conditional likelihoods, and $P(Z = 0)$ and $P(Z = 1)$ are the prior probabilities of the positive and negative classes, respectively. Garczarek proposed a calibration method using the inverted Beta distribution function to model the posterior class conditional probabilities and tune the parameters of the distribution using a moment matching method (Garczarek, 2002).

2.1.1.3 Asymmetric Laplace Method Bennett proposed a calibration method based on partitioning the space of calibrated probabilities into three different groups which was tailored for information retrieval applications (Bennett, 2003). Based on his observations in many document retrieval tasks, the score distributions behave quite differently in three regimes of "extremely irrelevant", "hard to discriminate" and "obviously relevant". He proposed the use of the asymmetric Laplace distribution to estimate the class conditional probabilities in Equation 2.2 as follows:

$$p(y|\theta, \beta, \gamma) = \begin{cases} \frac{\beta\gamma}{\beta+\gamma} \exp(-\beta(\theta - y)) & \text{if } y \leq \theta \\ \frac{\beta\gamma}{\beta+\gamma} \exp(-\gamma(y - \theta)) & \text{if } y > \theta \end{cases}$$

Bennett showed empirically that the asymmetric Laplace method can perform better than Platt's method on information retrieval-related tasks.

2.1.1.4 Piecewise Logistic Regression Zhang and Yang proposed an extension to Platt's calibration method by assuming that the log-odds of calibrated probabilities is a piecewise linear function. In their proposed method, the calibrated estimate is modeled as follows:

$$P(Z|y) = \frac{1}{1 + \exp(-Zf(y))}, \tag{2.3}$$

where y is the uncalibrated output of the classifier, Z is the true class of the instance (i.e., either -1 or 1), and $f(y)$ is a piecewise linear function. They defined $f(y)$ with $K+1$ knots $(\theta_0 < \theta_1 < \ldots < \theta_K)$ as $f(y) = \sum_{j=0}^{K} w_j l_j(y)$, where $l_j(y)$ is defined as:

$$l_j(y) = \begin{cases} \frac{y-\theta_{j-1}}{\theta_j - \theta_{j-1}} & \text{if } \theta_{j-1} \leq y < \theta_j \ (j = 1, 2, \ldots, K) \\ \frac{y-\theta_{j+1}}{\theta_j - \theta_{j+1}} & \text{if } \theta_j \leq y < \theta_{j+1} \ (j = 0, 1, \ldots, K-1) \\ 0 & \text{Otherwise} \end{cases}$$

The piecewise function $f(y)$ is continuous at all the $K+1$ knots, and its value will be equal to w_j at the j^{th} knot. They used a maximum likelihood method to estimate the parameters w_j. In order to define the number of the knots and the position of the knots, they used an ad hoc procedure. For the number of knots, they used a three-piece linear function based on an intuition similar to the asymmetric Laplace distribution-based calibration method, described above. The remaining task is to define the position of the knots $\theta_0, \theta_1, \theta_2, \theta_3$. They set $\theta_0 = miny \in D$ and $\theta_3 = \max\{y \in D\} + \epsilon$ where D is the training data and ϵ is a small positive number. For setting θ_1 and θ_2, they choose from $10\%, 20\%, \ldots, 90\%$ percentile of negative and positive instances, respectively, using the maximum likelihood method.

2.1.2 Non-Parametric Calibration Methods

2.1.2.1 Histogram Binning A popular non-parametric calibration method is the equal frequency histogram binning model which is also known as quantile binning (Zadrozny and Elkan, 2001b). In quantile binning, predictions are partitioned into B equal frequency bins. For each new prediction y that falls into a specific bin, the associated frequency of observed positive instances will be used as the calibrated estimate for $P(z = 1|y)$, where z is the true label of an instance that is either 0 or 1. Figure 2 shows a hypothetical example of using the equal frequency histogram

binning method to calibrate a hypothetical binary classifier. By using simple caching technique, histogram binning can be implemented in a way that allows it to be applied to large-scale data mining problems at test time. However, its limitations include (1) bins inherently "pigeonhole" calibrated probabilities into only B possibilities, (2) bin boundaries remain fixed over all predictions, and (3) there is uncertainty in the optimal number of the bins to use (Zadrozny and Elkan, 2002).

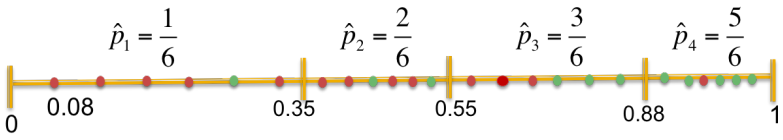


Figure 2: An example of histogram binning method for calibrating the output of a binary classification model. The green and red dots denote the instances that belong to positive and negative class, respectively. If the classification score is any where between 0.35 and 0.55, then the corresponding calibrated estimate will be $\frac{2}{6}$.

2.1.2.2 Isotonic Regression is the most commonly used non-parametric classifier calibration method in machine learning and data mining applications is the *isotonic regression-based calibration* (IsoReg) model (Zadrozny and Elkan, 2002). To build a mapping from the uncalibrated output of a classifier to the calibrated probability, IsoReg assumes it is an isotonic (monotonic) mapping following the ranking imposed by the base classifier. The commonly used algorithm for isotonic regression is the *Pool Adjacent Violators Algorithm* (PAVA), which is linear in the number of training data samples (Barlow et al., 1972). The IsoReg using PAVA works as follows: without loss of generality, we assume that the instances are sorted based on their (uncalibrated) classification scores y_i, so we have $y_1 \leq y_2 \leq \ldots \leq y_N$ (if the instances are not sorted, we can sort them in $O(N \log N)$). First, PAVA sets the calibrated estimate for each instance equal to the true class of the instance, so we have $\hat{p}_i^* = z_i$. Traversing the instances, for each pair of consecutive probabilities that violated the ordering if $\hat{p}_i^* > \hat{p}_{i+1}^*$ then they will be combined to form a new group of instances, and the calibrated estimate for the group will be equal to the their average $\hat{p}_i^*, \hat{p}_{i+1}^* = \frac{\hat{p}_i^* + \hat{p}_{i+1}^*}{2}$. The process will continue until the isotonic calibrated estimates are achieved

(i.e. until there is no pair of consecutive probabilities that violates the ordering).

An IsoReg model based on PAVA can be viewed as a histogram binning model (Zadrozny and Elkan, 2002) where the position of the boundaries are selected by fitting the best monotone approximation to the train data according to the ordering imposed by the classifier. While IsoReg can perform well on some real datasets, the monotonicity assumption makes can fail in real data mining applications. Specifically, this occurs in large scale data mining problems where we have to make simplifying assumptions in making computationally tractable classification models (e.g. naive Bayes classifiers) or learning algorithms (e.g. using variational methods in learning the parameters). Thus, one may need to relax the assumption depending on the application.

2.1.2.3 Similarity Binning Averaging Similarity binning averaging is an extension to histogram binning using the idea of cascading (Gama and Brazdil, 2000) in which the similarity of instances in the feature space will be taken into account when constructing the bins. The method has two phases as shown in Figure 3 [1]. In the first stage, the trained model M is used to generate uncalibrated probability estimates for the instances in the validation dataset. The estimated probability for the i^{th} instance will be added as a new feature to its corresponding feature set. This will yield a new dataset that is called validation data with probabilities (VDP). At the second stage (i.e. the test time), to calibrate a new instance I, it will be presented first to the model M to find the corresponding uncalibrated estimate p. The score will be added to the corresponding feature set of that instance. Next, the k most similar instances in the VDP dataset will be selected and the empirical class frequency of the neighboring instances will be used as the calibrated estimate for the instance I. SBA can be extended for multi-class classification problems. However, it is computationally intractable for large datasets and the correct choice of k is still a challenge in this method.

2.1.2.4 Adaptive Calibration of Predictions Adaptive calibration of predictions (ACP) is another extension to histogram binning (Jiang et al., 2012). ACP requires the derivation of a 95% statistical confidence interval around each individual prediction to build the bins. It then sets the calibrated estimate to the observed frequency of the instances with positive class, among all the

[1] The figure is taken from (Bella et al., 2009)

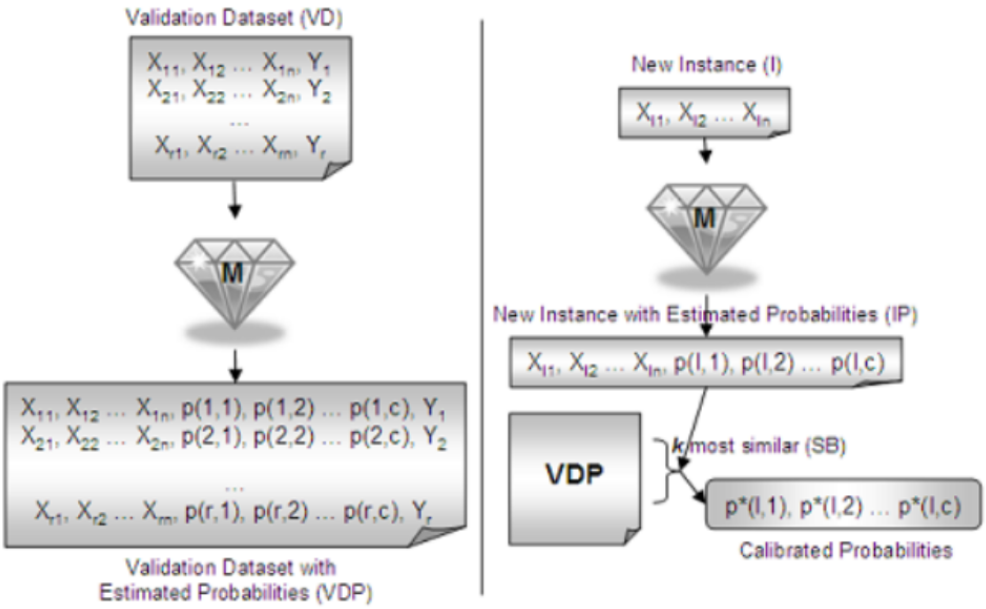


Figure 3: The left panel shows the first stage of the SBA, and the right panel shows the second stage of the SBA method

predictions in the training data, that fall within the bin. To date, ACP has been developed and evaluated using only logistic regression as the base classifier (Jiang et al., 2012).

2.2 OTHER RELATED METHODS

There are some less well-known calibration methods in the literature. For instance, Rüping (Rüping, 2006) proposed a method to render the calibration methods more robust against outliers in the dataset. The method trims the calibration datasets by removing outliers for which the generated estimation error is above a predefined threshold.

Bennett proposed a parametric method to calibrate the output of a naive Bayes classifier (Bennett, 2000). He modeled the probability of the positive class given the log-odds score generated by the classifier using a sigmoid function. This is equivalent to assuming that the log-odds is the

sufficient statistic to naive Bayes predictions. He proposed setting the parameters of the sigmoid function either heuristically or using an approach similar to Platt's method.

Rüping proposed a simple three-value (Rüping, 2004) scaling function to convert the classification scores generated by an SVM classifier into a probability space.

There is also a variation of the isotonic regression-based calibration method for predicting accurate probabilities with a ranking loss (Menon et al., 2012).

Another calibration method uses an optimization framework based on the alternating direction method of multipliers (ADMM) optimization method (Hestenes, 1969) to combine the output of multiple classifiers and obtain calibrated probabilities (Zhong and Kwok, 2013).

2.3 THEORETICAL WORKS

There exists a body of theoretical work on classifier calibration. For instance, Fawcett and Niculescu-Mizi (Fawcett and Niculescu-Mizil, 2007) showed that isotonic regression-based calibration using PAVA is equivalent to the receiver operating characteristics (ROC) convex hull method. As a result of this equivalence, for any given binary classification model and any given training data the isotonic regression based calibration using PAVA is identical to the convex-hull of the base classification model. Thus, the calibrated estimates generated by isotonic regression calibration using the PAVA algorithm on a given dataset will have discriminatin performance that is at least as good as the original (uncalibrated) classification scores in terms of area under the curve (AUC).

Cohen and Goldszmidt showed that an unbiased classifier is always well-calibrated; however, a well-calibrated classifier might not be unbiased. They also showed that the refinement error of a well-calibrated classifier provides an upper bound on the Bayes error. More specifically, they showed that $e_b < 2 * e_R$, where e_b is the Bayes error and e_R is the refinement error of the well-calibrated classifier. They also showed that using a threshold of 0.5 for the calibrated estimates is equivalent to finding the point of minimum error in a ROC curve (Fawcett, 2004; Lachiche and Flach, 2003). Finally, they also noted that calibrating a classifier is guaranteed not to decrease the classification accuracy.

2.4 HOW TO EVALUATE CALIBRATION METHODS

This section discusses the evaluation measures that we will use for comparing various classifier calibration methods. We first describe the concept of scoring rules and proper scoring rules. Then, we describe the concept of refinement versus calibration of a forecaster. Finally, we present the actual measures used for evaluating calibration methods.

2.4.1 Proper Scoring Rules

In decision theory, it is common to use scoring rules in order to assess the quality of a probability estimation made by a forecaster (in our case, a classification model). A score can be thought of as a "cost function" or "loss function" in the pattern classification context (Friedman et al., 2001). In terms of eliciting probability distributions, the role of a scoring rule is to encourage a forecaster to be accurate and honest in assessing the probability of the events (i.e., the probability of target class z in the context of classification problems) (Gneiting and Raftery, 2007; Garthwaite et al., 2005). A scoring rule $L(y, z)$ is defined as a function of the predicted probability y and the true target class z (which is equal to 0 or 1 in binary classification problems). A scoring function is deemed proper if its expected value is minimized by using $P(Z = z)$, which is the objective probability distribution of z. A strictly proper loss function is uniquely minimized by the objective probability distribution. Several commonly used (strictly) proper scoring rules are the Brier (Brier, 1950), logarithmic (Good, 1952) and spherical scores, which are shown in Table 2. More detailed information about proper scoring rules and their applications can be found in (Bickel, 2007; Gneiting and Raftery, 2007).

Table 2: Proper Scoring Rules for Decision Problem with a Binary Outcome

	L(y,0)	L(y,1)
Brier scoring rule	$-y^2$	$-(1-y)^2$
Logarithmic scoring rule	$\log(1-y)$	$\log y$
Spherical scoring rule	$\frac{1-y}{\sqrt{y^2+(1-y)^2}}$	$\frac{y}{\sqrt{y^2+(1-y)^2}}$

2.4.2 Calibration vs. Refinement

As mentioned above, the Brier score, also known as the quadratic score, is one of the strictly proper loss functions that will be minimized if the probability scores match the true probability distribution of the target class. For a binary classification problem, the Brier score is defined as the average square error between the estimated probability of $Z = 1$ and the true class z:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(y_i - z_i)^2, \tag{2.4}$$

where N is the number of training data, y_i is the estimated probability of $P(Z = 1|x_i)$ and z_i is the true class of the i'th instance (i.e., $z_i \in \{0, 1\}$). The Brier score states that if the classifier makes a mistake in its decision over the two existing classes, then it will be penalized proportional to its confidence with which it asserts its decision. It can be decomposed into two separate terms that measures calibration and refinement (DeGroot and Fienberg, 1983; Cohen and Goldszmidt, 2004). To briefly review this concept in the context of binary classification problem, let $y \in [0, 1]$ be the estimated probability generated by a binary classification model M. Following (DeGroot and Fienberg, 1983; Cohen and Goldszmidt, 2004), we assume that y takes a finite number of values in the interval $[0, 1]$ (e.g., $y \in \{0, 0.01, 0.02, \ldots, 0.99, 1\}$). Assume $p(Z = 1|y)$ denotes the probability that the true class of an instance is positive given that estimated probability generated by the classifier is y. Degroot showed that the Brier score can be rewritten as follows (DeGroot and Fienberg, 1983):

$$BS = \underbrace{\sum_{y}\pi(y)(y - P(Z = 1|y))^2}_{L_2 \text{ Calibration error}} + \underbrace{\sum_{y}\pi(y)P(Z = 1|y)(1 - P(Z = 1|y))}_{\text{Refinement Error}}, \tag{2.5}$$

where $\pi(y)$ is the prior probability that the classifier will output y on a randomly chosen instance (it can be estimated using long time frequency of instances with classification score of y, as we will do in Section 2.4.3). The first term in Equation 2.5 is the L_2 calibration error that measures how close the estimated probability y is to the true probability $P(Z = 1|y)$ (i.e., the long run fraction of positive instances when the classifier outputs y). The second term accounts for refinement that is smaller if the value of $P(Z = 1|y)$ is concentrated near 0 or 1. Refinement accounts for the usefulness of each prediction. Informally, we dislike the predictions that are close to uncertainty

0.5; we are more interested in those predictions that are close to certainty 0 or 1. Assume the prior probability of a positive instance is p, then a default classifier that predicts p for all instances will always be calibrated. However, in general the predictions that it makes are not very useful in terms of making a decision about the class of the object.

2.4.3 Evaluation Measures

Although proper scoring rules can be used to evaluate the calibration performance of classification models, in practice, it is more common to separately assess the calibration refinement, or calibration and discrimination (Parmigiani and Inoue, 2009). In this thesis, we will use AUC and Accuracy in order to evaluate discrimination performance of a classifier.

For evaluating calibration performance, we use two simple statistics that measure calibration relative to the ideal reliability diagram (DeGroot and Fienberg, 1983; Niculescu-Mizil and Caruana, 2005b). Figure 1 shows an example of such a diagram. These measures are called *Expected Calibration Error* (ECE), and *Maximum Calibration Error* (MCE). In computing these measures, the predictions are sorted and partitioned into 10 bins. The predicted value of each test instance falls into one of the bins. The ECE calculates Expected Calibration Error of the bins, and MCE calculates the Maximum Calibration Error among the bins, using empirical estimates as follows:

$$ECE = \sum_{i=1}^{10} \pi(i) \cdot |o_i - e_i| \quad , \quad MCE = \max\left(|o_i - e_i|\right),$$

where o_i is the true fraction of positive instances in bin i, e_i is the mean of the post-calibrated probabilities for the instances in bin i, and $\pi(i)$ is the empirical probability (fraction) of all instances that fall into bin i. The lower the values of ECE and MCE, the better the calibration of a model.

Finally, we use *root mean square error* (RMSE) as a summary measure to evaluate the performance of a (binary) classifiers. This is due to the fact that RMSE is equal to the square root of the Brier score that evaluates the overall performance of a classifier in terms of calibration and refinement.

3.0 BINARY CLASSIFIER CALIBRATION METHODS

This chapter describes our methodological and algorithmic contributions that includes five new binary classifier calibration methods. These methods can be categorized as follows: (1) Extending histogram binning, also known as quantile binning, using non-parametric binary classification in which we used *kernel density estimation* (KDE) and *Dirichlet process mixtures* (DPM), (2) Extending histogram binning using Bayesian model averaging, (3) Extending histogram binning using selective Bayesian model averaging, (4) Extending isotonic regression using an ensemble of near isotonic regression models. (5) Extending all the binning based models using an ensemble of linear trend estimation method. The reminder of this chapter describes the above extensions in detail.

3.1 EXTENDING HISTOGRAM BINNING USING NON-PARAMETRIC BINARY CLASSIFICATION

In this section, we show that the histogram binning calibration method (Zadrozny and Elkan, 2001b) is a simple nonparametric plug-in classifier. In the calibration problem, given an uncalibrated probability estimate y, one way of finding the calibrated estimate $\hat{y} = \mathbb{P}(Z = 1|y)$ is to apply Bayes' rule as follows:

$$\mathbb{P}(Z = 1|y) = \frac{P(Z = 1) \cdot P(y|Z = 1)}{p(Z = 1) \cdot P(y|Z = 1) + P(z = 0) \cdot P(y|Z = 0)}, \tag{3.1}$$

where $P(z = 0)$ and $P(z = 1)$ are the priors of class 0 and 1 that are estimated from the training dataset. Also, $P(y|z = 1)$ and $P(y|z = 0)$ are predictive likelihood terms. If we use the histogram

density estimation method for estimating the predictive likelihood terms in the Bayes rule equation 3.1 we obtain the following: $\hat{P}(y|z=t) = \sum_{j=1}^{B} \frac{\hat{\theta}_j^t}{h_j} I(y \in B_j)$, where $I(.)$ is an indicator function, $t = \{0,1\}$, $\hat{\theta}_j^0 = \frac{1}{n}\sum_{i=1}^{N} I(y_i \in B_j, z_i = 0)$, and $\hat{\theta}_j^1 = \frac{1}{m}\sum_{i=1}^{N} I(y_i \in B_j, z_i = 1)$ are the empirical estimates of the probability of a prediction when class $z = t$ falls into bin B_j. Also, m and n define the total number of positive and negative instances in training data, respectively. Now, let us assume $y \in B_j$; By substituting the value of empirical estimates of $\hat{\theta}_j^0 = \frac{n_j}{n}$, $\hat{\theta}_j^1 = \frac{m_j}{m}$, $\hat{P}(z=0) = \frac{n}{N}$, $\hat{P}(z=1) = \frac{m}{N}$ from the training data and performing some basic algebra we obtain the following calibrated estimate: $\hat{y} = \frac{m_j}{m_j + n_j}$, where m_i and n_j are the number of positive and negative examples in bin B_j.

The above computations show that the histogram-binning calibration method is actually a simple plug-in classifier where we use the histogram-density method for estimating the predictive likelihood in terms of Bayes rule as given by 3.1. By casting histogram binning as a plug-in method for classification, it is possible to use more advanced frequentist methods for density estimation rather than using simple histogram-based density estimation. For example, if we use kernel density estimation (KDE) for estimating the predictive likelihood terms, the resulting calibrated probability $P(Z=1|X=x)$ is as follows:

$$\hat{P}(Z=1|Y=y) = \frac{nh_0 \sum_{i=1}^{N} K\left(\frac{|y-y_i|}{h_1}\right) I(z_i=1)}{nh_0 \sum_{i=1}^{N} K\left(\frac{|y-y_i|}{h_1}\right) I(z_i=1) + mh_1 \sum_{i=1}^{N} K\left(\frac{|y-y_i|}{h_0}\right) I(z_i=0)}, \quad (3.2)$$

where y_i are uncalibrated probability estimates generated by the base classifier in the training data, and m and n are respectively the number of positive and negative examples in training data. Also h_0 and h_1 are defined as the bandwidth of the predictive likelihood for class 0 and class 1 (Wasserman, 2006). The bandwidth parameters can be optimized using cross validation techniques. However, in this thesis we use Silverman's rule of thumb (Silverman, 1986) for setting the bandwidth to $h = 1.06\hat{\sigma}N^{-\frac{1}{5}}$, where $\hat{\sigma}$ is the empirical unbiased estimate of variance. It is possible to use the same bandwidth for both class 0 and class 1, which leads to the Nadaraya-Watson kernel estimator that we use in our experiments. However, we noticed that there are some cases for which KDE with different bandwidths performs better.

There are different types of smoothing kernel functions, such as the Gaussian, Boxcar, Epanechnikov, and Tricube functions. Due to the similarity of the results we obtained when using different type of kernels, we only report here the results of the simplest one, which is the Boxcar kernel.

$$
\begin{aligned}
&Boxcar : K(x) = \frac{1}{2} I(x > 0) \\
&Gaussian : K(x) = \frac{1}{\sqrt{2\pi} e^{\frac{-x^2}{2}}} \\
&Epanechnikov : K(x) = \frac{3}{4}(1 - x^2) I(x > 0) \\
&Tricube : K(x) = \frac{70}{81}(1 - |x|^3)^3 I(x > 0),
\end{aligned}
$$

It has been shown in Wasserman (2006) that kernel density estimators are mini-max rate estimators, and under the L_2 loss function the risk of the estimator converges to zero with the rate of $O_P(n^{\frac{-2\beta}{(2\beta+d)}})$, where β is a measure of smoothness of the target density, d is the dimensionality of the input data, and n is the number of training instances. From this convergence rate, we can infer that the application of kernel density estimation is likely to be practical when d is low. Fortunately, for the binary classifier calibration problem, the input space of the model is the space of uncalibrated predictions, which is a one-dimensional input space. This justifies the application of KDE to the classifier calibration problem.

The KDE approach presented above represents a non-parametric frequentist approach for estimating the likelihood terms of equation 3.1. Instead of using the frequentist approach, we can use Bayesian methods for modeling the density functions. The Dirichlet Process Mixture (DPM) method is a well-known Bayesian approach for density estimation (Antoniak, 1974; Ferguson, 1973; Escobar and West, 1995; MacEachern and Muller, 1998). For building a Bayesian calibration model, we model the predictive likelihood terms $P(X_i = x|Z_i = 1)$ and $P(X_i = x|Z_i = 0)$ in Equation 3.1 using the DPM method. Due to a lack of space, we do not present the details of the DPM model here, but instead refer the reader to (Antoniak, 1974; Ferguson, 1973; Escobar and West, 1995; MacEachern and Muller, 1998).

There are different ways of performing inference in a DPM model. One can choose to use either Gibbs sampling (non-collapsed or collapsed) or variational inference, for example. In implementing our calibration model, we use the variational inference method described in Kurihara et al. (2007). We chose it because it has fast convergence. We will refer to it as DPM.

3.2 BAYESIAN EXTENSION OF HISTOGRAM BINNING

In this section we describe the Bayesian extensions that we proposed for histogram binning calibration method. The two proposed methods are published at SDM 2015 (Pakdaman Naeini et al., 2015b), and AAAI 2015 (Pakdaman Naeini et al., 2015a). The main idea in both methods is to consider ensemble of histogram binning models and their weighted average to build a calibration model. In SDM 2015 we considered all possible binning models induced by the train data and we used $k2$ Bayesian scoring function (Cooper and Herskovits, 1992) for the weights; in the AAAI 2015 we used a selective Bayesian approach and the Bayesian $BDeu$ scoring function (Heckerman et al., 1995) for the weights. In the following we describe each calibration method in more detail[1].

3.2.1 Full Bayesian Approach

In this section we present two new Bayesian non-parametric methods for binary classifier calibration that generalize the histogram-binning calibration method (Zadrozny and Elkan, 2001b) by considering all possible binnings of the training data. The first proposed method, which is based on Bayesian model selection, is called *Selection over Bayesian* Binnings (SBB). We also generalize SBB by model averaging over all possible binnings; it is called *Averaging over Bayesian* Binnings (ABB). There are two main challenges here. One is how to score a binning model; we use a Bayesian score. The other is how to efficiently search over such a large space of binnings; we use dynamic programming to address this issue.

3.2.1.1 Bayesian Calibration Score Let Y and Z define respectively an uncalibrated classifier prediction and the associated true class of an instance. The i'th training sample is an instantiation of Y and Z, denoted by y_i and z_i. Also, let S be the *sorted* set of all uncalibrated classifier predictions $\{y_1, y_2, \ldots, y_N\}$ and $S_{l,u}$ be a list of the first elements of S, starting at l'th index and ending at u'th index. Let Pa denote a partitioning of S into a fixed number of bins $Pa = \{Bin_1, Bin_2, \ldots, Bin_B\}$ where, B is the total number of bins used to define Pa. A binning model

[1]Note that the K2 and BDeu scoring functions are two possible scoring functions among others that could be used (e.g., BIC, AIC, and AICc).

M induced by the training set is defined as:

$$M \equiv \{B, S, Pa, \Theta\}, \tag{3.3}$$

where, Θ is the set of all the calibration model parameters $\Theta = \{\theta_1, \ldots, \theta_B\}$, which are defined as follows. For a bin Bin_b, which is determined by S_{l_b,u_b}, the distribution of the class variable $P(Z = 1|y \in Bin_b)$ is modeled as a binomial distribution with parameter θ_b. Thus, Θ specifies all the binomial distributions for all the existing bins in Pa. We note that our binning model is motivated by the model introduced in (Lustgarten et al., 2011) for variable discretization, which is here customized to perform classifier calibration. We score a binning model M as follows:

$$Score(M) = P(M) \cdot P(D|M), \tag{3.4}$$

where D defines the set of all pairs $\{(z_i, b_i)|i = 1, \ldots, N\}$, and b_i is the index of the bin in which i'th training instance is located. The marginal likelihood $P(D|M)$ in Equation 3.4 is derived using the marginalization of the joint probability of $P(D, \Theta)$ over all parameter space according to the following equation:

$$P(D|M) = \int_{\Theta} P(D|M, \Theta)P(\Theta|M)d_{\Theta} \tag{3.5}$$

Equation 3.5 has a closed form solution under the following assumptions: (1) All samples are i.i.d and the class distribution $P(Z|y \in Bin_b)$,which is the class distribution for instances located in Bin_b, is modeled using a binomial distribution with parameter θ_b, (2) the distribution of class variables over two different bins are independent of each other, and (3) the prior distribution over binning model parameters θs are modeled using a $Beta$ distribution. We also assume that the parameters of the $Beta$ distribution α and β are both equal to one, which corresponds to having a uniform distribution over each θ_b. Using $K2$ Bayesian model scoring, the closed form solution to the marginal likelihood given the above assumptions is as follows (Heckerman et al., 1995; Cooper and Herskovits, 1992):

$$P(D|M) = \prod_{b=1}^{B} \frac{n_b!\, m_b!}{(N_b + 1)!}, \tag{3.6}$$

where N_b is the total number of training instances located in the Bin_b. Also, n_b and m_b are respectively the number of class *zero* and class *one* instances among all N_b training instances inside the Bin_b.

The term $P(M)$ in Equation 3.4 specifies the prior probability of a binning of calibration model M. It can be interpreted as a structure prior, which we define as follows. Let $Prior(k)$ be the prior probability of there being a bin boundary between y_k and y_{k+1} in the binning given by model M, and model it using a $Poisson$ distribution with the mean parameter λ. For k from 1 to $N-1$, we define the ***Prior(k)*** function as:

$$Prior(k) = 1 - e^{-\lambda \frac{d(k,k+1)}{d(1,n)}} \tag{3.7}$$

where, $d(i,j) = y_j - y_i$ represents the distance between the two (uncalibrated) classifier outputs y_j and y_i, and y_j is greater than y_i. For the boundary cases where $k = 0$ and $k = N$, we define $Prior(0) = 1$ and $Prior(N) = 1$ which correspond to having a bin boundary at the lowest and the highest possible uncalibrated probabilities in S.

Consider the prior probability for the presence of bin Bin_b, which contains the sequence of training instances S_{l_b,u_b} according to model M. Assuming independence of the appearance of partitioning boundaries, we can calculate the prior of the boundaries defining bin Bin_b by using the $Prior$ function as follows:

$$Prior(u_b) \left(\prod_{k=l_b}^{u_b-1} (1 - Prior(k)) \right) \tag{3.8}$$

where the product is over all training instances from S_{l_b} to S_{u_b-1} , inclusive. Expression 3.8 gives the prior probability that no bin boundary is present between any consecutive pairs of values y_k in the sequence S_{l_b,u_b} and at least one bining boundary is between the values y_{u_b} and y_{u_b+1}. Combining Equations 3.8 and 3.6 into Equation 3.4, we obtain the following Bayesian score for calibration model M:

$$Score(M) = \prod_{b=1}^{B} \left[Prior(u_b) \left(\prod_{k=l_b}^{u_b-1} (1 - Prior(k)) \right) \frac{n_b!\, m_b!}{(N_b+1)!} \right] \tag{3.9}$$

3.2.1.2 The SBB and ABB models We can use the above Bayesian score to perform model selection or model averaging. Selection involves choosing the best partitioning model M_{opt} and calibrating a prediction x as $P(x) = P(x|M_{opt})$. As mentioned, we call this approach *Selection over Bayesian Binnings* (SBB). Model averaging involves calibrating predictions over all possible binnings. We call this approach *Averaging over Bayesian Binnings* (ABB) model. A calibrated prediction in ABB is derived as follows:

$$\begin{aligned} P(x) &= \sum_{i=1}^{2^{N-1}} P(M_i|D)P(x|M_i) \\ &\propto \sum_{i=1}^{2^{N-1}} Score(M_i)P(x|M_i), \end{aligned} \tag{3.10}$$

where N is the total number of training instances.

Both (SBB) and (ABB) consider all possible binnings of the N predictions in the training data, which is exponential in N. Thus, in general, a brute-force approach is not computationally tractable. Therefore, we apply dynamic programming, as described in the next two sections.

3.2.1.3 Dynamic Programming Search of SBB This section summarizes the dynamic programming method used in SBB. Recall that S is the *sorted* set of all uncalibrated classifier's outputs $\{y_1, y_2, \ldots, y_N\}$ in the training data set. Let $S_{1,u}$ define the prefix of set S including the set of the first u uncalibrated estimates $\{y_1, y_2, \ldots, y_u\}$. Consider finding the optimal binning models $M_{1,u}$ corresponding to the subsequence $S_{1,u}$ for $u \in 1, 2, \ldots, N$ of the set S. Assume we have already found the highest score binning of these models $M_{1,1}, M_{1,2}, \ldots, M_{1,u-1}$, corresponding to each of the subsequences $S_{1,1}, S_{1,2}, \ldots, S_{1,u-1}$. Let $v_1, v_2, \ldots, v_{u-1}$ denote the respective scores of the optimal binnings of these models. Let $Score_{i,u}$ be the score of subsequence $\{y_i, y_{i+1}, \ldots, y_u\}$ when it is considered as a single bin in the calibration model $M_{1,u}$. For all l from u to 1, SBB computes $v_{l-1} \times Score_{l,u}$, which is the score for the highest scoring binning $M_{1,u}$ of set $S_{1,u}$ for which subsequence $S_{l,u}$ is considered as a single bin. Since this binning score is derived from two other scores , we call it a *composite score* of the binning model $M_{1,u}$. The fact that this composite score is a product of two scores follows from the decomposition of Bayesian scoring measure we

are using, as given by Equation 3.9. In particular, both the prior and marginal likelihood terms of the score are decomposable.

In finding the best binning model $M_{1,u}$, SBB chooses the maximum composite score over all l, which corresponds to the optimal binning for the training data subset $S_{1,u}$; this score is stored in v_u. By repeating this process from 1 to N, SBB derives the optimal binning of set $S_{1,N}$, which is the best binning over all possible binnings. The pseudo code for the SBB dynamic programming is shown in Algorithm 1. The computational time complexity of the algorithm is $O(N^2)$.

input : $D = \{(y_1, z_1), \ldots, (y_N, z_N)\}$

output : Best binning models $M_{1,1}, \ldots, M_{1,N}$ and
corresponding Bayesian Scores $v_{1,1}, \ldots, v_{1,N}$

$v_{1,0} \leftarrow 1$;

$M_{1,0} \leftarrow \{\}$;

for $u \leftarrow 1$ **to** *N* **do**

  $p \leftarrow Prior(u)$;

  $v_{1,u} \leftarrow 0$;

  for $l \leftarrow u$ **to** *1* **do**

    $Bin_b \leftarrow [y_l, \ldots, y_u]$; % Defining the bin Bin_b

    $ML \leftarrow \frac{n_b! m_b!}{(N_b+1)!}$;

    $Score_{lu} \leftarrow p \times ML$;

    if $v_{1,l-1} \times Score_{lu} > v_{1,u}$ **then**

      $M_{1,l} \leftarrow M_{1,l-1} \cup \{Bin_b\}$;

      $v_{1,u} \leftarrow v_{1,l-1} \times Score_{l,u}$;

    end

    $p \leftarrow p \times (1 - Prior(l-1))$;

  end

end

Algorithm 1: The pseudo code for the dynamic programming search of SBB. It will use $M_{1,N}$, the highest scored binning model, to find the calibrated probability estimate of a new instance at the test time.

3.2.1.4 Dynamic Programming Search of ABB

The dynamic programming approach used in ABB is based on the above dynamic programming approach in SBB. It focuses on calibrating a particular instance $P(x)$. The ABB algorithm uses the $decomposability$ property of the Bayesian binning score in Equation 3.9. Assume we have already found in one forward run of the SBB method the highest score binning of the models $M_{1,1}, M_{1,2}, \ldots, M_{1,N}$, which correspond to each of the subsequences $S_{1,1}, S_{1,2}, \ldots, S_{1,N}$, respectively; let the values $V_1^f, V_2^f, \ldots, V_N^f$ denote the respective accumulative scores of the binning for these models [2], which we cache. We perform an analogous dynamic programming procedure to SBB in a backward manner (from highest to lowest prediction) and compute the highest score binning of these models $M_{N,N}, M_{N-1,N}, \ldots, M_{1,N}$, which correspond to each of the subsequences $S_{N,N}, S_{N-1,N}, \ldots, S_{1,N}$, respectively; let the values $V_N^b, V_{N-1}^b, \ldots, V_1^b$ denote the respective accumulative scores for these models, which we also cache. Using the decomposability property of the binning score given by 3.9, we can write the Bayesian model averaging estimate given by Equation 3.10 as follows:

$$P(x) \propto \sum_{1 \leq l \leq u \leq N} \left(V_{l-1}^f \times Score_{l,u} \times V_{u+1}^b \times \hat{p}_{l,u}(x) \right) \tag{3.11}$$

where $\hat{p}_{l,u}(x)$ is obtained using the frequency[3] of the training instances in the bin containing the predictions $S_{l,u}$. Remarkably, the dynamic programming implementation of ABB is also $O(N^2)$. However, since it is instance specific, this time complexity holds for each prediction that is to be calibrated (e.g., each prediction in a test set). To address this problem, we can partition the interval [0, 1] into R equally spaced bins and stored the ABB output for each of those bins. The training time is therefore $O(RN^2)$. During testing, a given p_{in} is mapped to one of the R bins and the stored calibrated probability is retrieved, which can all be done in $O(1)$ time.

3.2.2 Selective Bayesian Approach

In this section we present a new non-parametric calibration method called Bayesian Binning into Quantiles (BBQ) using a selective Bayesian approach. BBQ extends the simple histogram-binning calibration method (Zadrozny and Elkan, 2001b) by considering multiple binning models and their

[2] Instead of recording the optimal score, it defines the sum of the scores for all the corresponding binning models

[3] We actually use smoothing of these counts, which is consistent with the Bayesian priors in the scoring function

combination. The main challenge here is to decide on how to pick the models and how to combine them. BBQ considers multiple equal-frequency binning models that distribute the data-points in the training set equally across all bins. The different binning models differ in the number of bins they have. We combine them using a Bayesian score derived from the BDeu (Heckerman et al., 1995) score used for learning Bayesian network structures. Similar to what we did in SBB and ABB we define the score of a binning model in BBQ as follows:

$$Score(M) = P(M) \cdot P(D|M) \tag{3.12}$$

In order to derive the marginal likelihood $P(D|M)$ in Equation 3.12 we use the same notation and the assumptions that we used for SBB and ABB in Section 3.2.1.1. However, in BBQ we use BDeu Bayesian model scoring and we set the hyper-parameters of the Beta distribution [4] to be equal to $\alpha_b = \frac{N'}{B} p_b$ and $\beta_b = \frac{N'}{B}(1 - p_b)$, where N' is the equivalent sample size expressing the strength of our belief in the prior distribution and p_b is the midpoint of the interval defining the b'th bin in the binning model M. Given the above assumptions, the marginal likelihood can be expressed as (Heckerman et al., 1995):

$$P(D|M) = \prod_{b=1}^{B} \frac{\Gamma(\frac{N'}{B})}{\Gamma(N_b + \frac{N'}{B})} \frac{\Gamma(m_b + \alpha_b)}{\Gamma(\alpha_b)} \frac{\Gamma(n_b + \beta_b)}{\Gamma(\beta_b)},$$

where Γ is the gamma function and N_b is the total number of training instances located in the b'th bin. Also, n_b and m_b are respectively the number of class *zero* and class *one* instances among all N_b training instances in bin b. The term $P(M)$ in Equation 3.12 specifies the prior probability of the binning model M. In our experiments we use a uniform prior for modeling $P(M)$. BBQ uses the above Bayesian score to perform model averaging over the space of all possible equal frequency binnings. We could have also used the above Bayesian score to perform the model selection, which in our case would yield a single binning model. However, model averaging is typically superior to model selection (Hoeting et al., 1999). Hence a calibrated prediction in our BBQ framework is defined as:

$$P(z = 1|y) = \sum_{i=1}^{T} \frac{Score(M_i)}{\sum_{j=1}^{T} Score(M_j)} P(z = 1|y, M_i),$$

[4] the prior distribution over binning model parameters θ_bs

where T is the total number of binning models considered and $P(z = 1|y, M_i)$ is the probability estimate obtained using the binning model M_i, for the (uncalibrated) classifier output y. To choose models M_i we choose a restricted range of binning models, each defined by a different number of bins. We define the range of possible values of the number of bins as $B \in \{\frac{\sqrt[3]{N}}{C}, \ldots, C\sqrt[3]{N}\}$, where C is a constant that controls the number of binning models ($C = 10$ in our experiments). The choice of the above range is due to some previous results that show that the histogram binning classifier achieves the best convergence rate on the excess risk (i.e., the difference between the best empirical-risk-minimizer classifier and the Bayes classifier) for Lipschitz Bayes decision boundaries when we set number of bins to $\theta(\sqrt[3]{N})$ (Klemela, 2009; Scott and Nowak, 2003; singh, 2011; Nowak, 2009). Although the results are valid for histogram classifiers with fixed bin size, our experiments show that both fixed bin size and fixed frequency histogram classifiers behave quite similarly. We conjecture that a histogram classifier with equal frequency binning also achieves the best convergence rate on the excess risk by setting the number of bins to $\theta(\sqrt[3]{N})$. This is an interesting open problem for our future research.

Note that we can further restrict the number of binning models used in averaging in the application stage. That is, we may start by calculating the Bayesian score for all models in the above range, and select a subset of those that yield a higher Bayesian score afterwards. The number of resulting models can be determined by apriori fixing the number of models to be used in averaging or by checking for the sharp drops in the Bayesian scores over all such models. Assume $S_1, S_2, \ldots, S_N$ are the sorted Bayesian scores of histogram models in a decreasing order. We fix a small number $\alpha > 0$ ($\alpha = 0.001$ in our experiments) and pick the first k_α associated binning models as the refined set of models, where $k_\alpha = min\{k : \frac{S_k - S_{k+1}}{\sigma^2} \leq \alpha\}$ and σ^2 is the empirical variance of the Bayesian scores.

By using the above range the computational cost of BBQ will be equivalent to using a single equal frequency histogram binning model both in training and test. For the training time, the dominant computational cost is sorting uncalibrated probabilities, which is asymptotically $\theta(N \log N)$. The asymptotic computational cost of BBQ at test time is at the same order of time as single histogram binning since the number of binning models that will be selected after the post processing refinement procedure will be very small in practice. By the way, at the worst case the number of binning models in BBQ is equal to $\theta(\sqrt[3]{N})$, thus the worst case time complexity at the test time is

equal to $\theta(\sqrt[3]{N}\log N)$.

3.3 CALIBRATION USING NEAR ISOTONIC REGRESSION

In this section we introduce the *ensemble of near isotonic regression* (ENIR) calibration method. The essential idea in ENIR is to use prior knowledge that the scores to be calibrated are in fact generated by a well-performing classifier in terms of discrimination. Isotonic Regression-based calibration (IsoReg) also uses such prior knowledge; however, it is biased by constraining the calibrated scores to obey the ranking imposed by the classifier. In the limit, this is equivalent to presuming the classifier has AUC equal to 1, which rarely happens in real world applications. In contrast, BBQ does not make any assumptions about the correctness of classifier rankings. ENIR provides a balanced approach that spans between IsoReg and BBQ. In particular, ENIR assumes that the mapping from uncalibrated scores to calibrated probabilities is a near isotonic (monotonic) mapping; it allows violations of the ordering imposed by the classifier and then penalizes them through the use of a regularization term. Figure 4 shows the calibration curve of a logistic regression binary classifier trained on the liver-disorder UCI dataset. The dataset consists of 345 total instances and the final AUC is equal to 0.73. The figure shows that the isotonicity assumption made by IsoReg is violated when comparing the frequency of observations in the first and the second bins.

ENIR utilizes the near isotonic regression method (Tibshirani et al., 2011) that seeks a nearly monotone approximation for a sequence of data $y_1, \ldots, y_n$. The proposed calibration model extends the commonly used isotonic regression-based calibration by a (approximate) selective Bayesian averaging of a set of nearly isotonic regression models. The set includes the isotonic regression model as an extreme member. From another viewpoint, ENIR can be considered as an extension to a recently proposed calibration model BBQ (Pakdaman Naeini et al., 2015a) by relaxing the assumption that probability estimates are independent inside the bins and finding the boundary of the bins automatically through an optimization algorithm.

Before getting into the details of the method, we define some notation. Let y_i and z_i define respectively an uncalibrated classifier prediction and the true class of the i'th instance. Since we

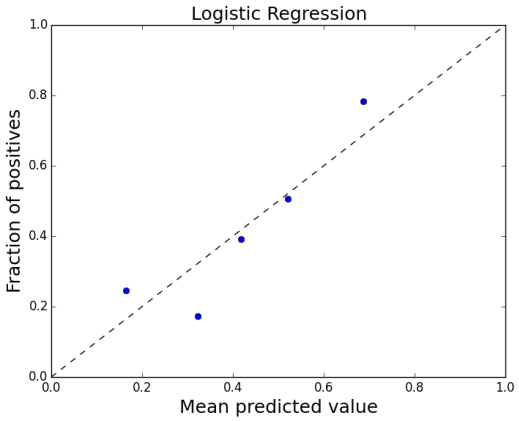


Figure 4: Calibration curve based on the use of 5 equal frequency bins when we use a logistic regression model for the binary classification task in the liver-disorder UCI dataset. Considering the frequency of observations in the first and the second bin, we notice the violation of the isotonicity assumption that was made by IsoReg.

aim to calibrate a binary classifier's output[5], thus, $z_i \in \{0, 1\}$ and $y_i \in [0, 1]$. Let $\mathcal{D}$ define the set of all training instances (y_i, z_i). Without loss of generality, we can assume that the instances are sorted based on the classifier scores y_i, so we have $y_1 \leq y_2 \leq \ldots \leq y_N$, where N is the total number of samples in the training data.

The standard isotonic regression based calibration model finds the calibrated probability estimates by solving the following optimization problem:

$$\begin{aligned} \hat{\boldsymbol{p}}_{iso} = \underset{\boldsymbol{p} \in R^N}{\text{argmin}} \quad & \frac{1}{2}\sum_{i=1}^{N}(p_i - z_i)^2 \\ \text{s.t.} \quad & p_1 \leq \ldots \leq p_N \\ & 0 \leq p_i \leq 1 \; \forall i \in \{1, \ldots, N\}, \end{aligned} \tag{3.13}$$

where $\hat{\boldsymbol{p}}_{iso}$ is the vector of calibrated probability estimates. The rationale behind the model is to assume that the base classifier ranks the instances correctly. To find the calibrated probability

[5]For classifiers that output scores that are not in the unit interval (e.g. SVM), we use a simple sigmoid transformation $f(x) = \frac{1}{1+\exp(-x)}$ to transform the scores into the unit interval.

estimates, it seeks the best fit of the data that are consistent with the classifier's ranking. A unique solution to the above convex optimization program exists and can be obtained by an inductive iterative algorithm called *pool adjacent violator algorithm* (PAVA) that runs in $O(N)$. Note, however, that isotonic regression calibration still needs $O(N \log N)$ computations, due to the fact that instances are required to be sorted based on the classifier scores y_i. PAVA iteratively groups the consecutive instances that violate the ranking constraint and uses their average over z (frequency of positive instances) as the calibrated estimate for all the instances within the group. We define the set of these consecutive instances that are located in the same group and attain the same predicted calibrated estimate as a bin. Therefore, an isotonic regression-based calibration can be viewed as a histogram binning method (Zadrozny and Elkan, 2002) where the position of boundaries are selected by fitting the best monotone approximation to the training data according to the ranking imposed by the classifier.

One can show that the second constraint in the optimization given by Equation 3.13 is redundant, and it is possible to rewrite the equation in the following equivalent form:

$$\begin{aligned} \hat{\boldsymbol{p}}_{iso} = \underset{\boldsymbol{p} \in R^N}{\text{argmin}} \quad & \frac{1}{2} \sum_{i=1}^{N} (p_i - z_i)^2 + \lambda \sum_{i=1}^{N-1} (p_i - p_{i+1}) \nu_i \\ \text{s.t.} \quad & \lambda = +\infty, \end{aligned} \tag{3.14}$$

where $\nu_i = \mathbb{1}(p_i > p_{i+1})$ is the indicator function of ranking violation. Relaxing the equality constraint in the above optimization program leads to a new optimization problem called nearly isotonic regression (Tibshirani et al., 2011).

$$\hat{\boldsymbol{p}}_{\lambda} = \underset{\boldsymbol{p} \in R^N}{\text{argmin}} \quad \frac{1}{2} \sum_{i=1}^{N} (p_i - z_i)^2 + \lambda \sum_{i=1}^{N-1} (p_i - p_{i+1}) \nu_i, \tag{3.15}$$

where λ is a positive real number that regulates the tradeoff between the monotonicity of the calibrated estimates with the goodness of fit by penalizing adjacent pairs that violate the ordering imposed by the base classifier. The above optimization problem is convex having a unique solution $\hat{\boldsymbol{p}_{\lambda}}$, where the use of the subscript λ emphasizes the dependency of the final solution to the value of λ.

The entire path of solutions for any value of λ of the near isotonic regression problem can be found using a similar algorithm to PAVA which is called *modified pool adjacent violator algorithm*

(mPAVA) (Tibshirani et al., 2011). mPAVA finds the whole solution path in $O(N \log N)$, and needs $O(N)$ memory space. Briefly, the algorithm works as follows: It starts by constructing N bins, each bin containing a single instance of the train data. Next, it finds the solution path by starting from the saturated fit $p_i = z_i$, that corresponds to setting $\lambda = 0$, and then increasing λ iteratively. As the λ increases the calibrated probability estimates $\hat{\boldsymbol{p}}_{\lambda,i}$, for each bin, will change linearly with respect to λ until the calibrated probability estimates of two consecutive bins attain equal value. At this stage, mPAVA merges the two bins that have the same calibrated estimate to build a larger bin, and it updates their corresponding estimate to a common value. The process continues until there is no change in the solution for a large enough value of λ that corresponds to finding the standard isotonic regression solution. The essential idea of mPAVA is based on a theorem stating that if two adjacent bins are merged on some value of λ to construct a larger bin, then the new bin will never split for all larger values of λ (Tibshirani et al., 2011).

mPAVA yields a collection of nearly isotonic calibration models, with the over fitted calibration model at one end ($\hat{\boldsymbol{p}}_{\lambda=0} = \boldsymbol{z}$) and the isotonic regression solution at the other ($\hat{\boldsymbol{p}}_{\lambda=\lambda_\infty} = \hat{\boldsymbol{p}}_{iso}$), where λ_∞ is a large positive real number. Each of these models can be considered as a histogram binning model where the position of boundaries and the size of bins are selected according to how well the model trades off the goodness of fit with the preservation of the ranking generated by the classifier, which is governed by the value of λ, (As λ increases the model is more concerned to preserving the original ranking of the classifier, while for the small λ it prioritizes the goodness of fit.)

ENIR employs the approach just described to generate a collection of models (one for each value of λ). It then uses the Bayesian Information Criterion (BIC) to score each of the models [6]. Assume mPAVA yields the binning models $M_1, M_2, \ldots, M_T$, where T is the total number of models generated by mPAVA. For any new classifier output y, the calibrated prediction in the *ENIR* model is defined using selective Bayesian model averaging (Hoeting et al., 1999):

$$P(z = 1|y) = \sum_{i=1}^{T} \frac{Score(M_i)}{\sum_{j=1}^{T} Score(M_j)} P(z = 1|y, M_i),$$

where $P(z = 1|y, M_i)$ is the probability estimate obtained using the binning model M_i for the uncalibrated classifier output y. Also, $Score(M_i)$ is defined using the BIC scoring function (Schwarz

[6] Note that we exclude the highly overfitted model that corresponds to $\lambda = 0$ from the set of models in ENIR

et al., 1978). Next, for the sake of the completeness, we briefly describe the mPAVA algorithm; more detailed information about the algorithm and the derivations can be found in (Tibshirani et al., 2011).

3.3.1 The Modified PAV Algorithm

Suppose at a value of λ we have N_λ bins, $B_1, B_2, \ldots, B_{N_\lambda}$. We can represent the unconstrained optimization program given by Equation 3.15 as the following loss function that we seek to minimize :

$$\begin{aligned}\mathcal{L}_{B,\lambda}(\boldsymbol{z}, \boldsymbol{p}) = \quad & \frac{1}{2}\sum_{i=1}^{N_\lambda}\sum_{j\in B_i}(p_{B_i} - z_i)^2 + \\ & \lambda\sum_{i=1}^{N_\lambda - 1}(p_{B_i} - p_{B_{i+1}})\nu_i,\end{aligned} \tag{3.16}$$

where p_{B_i} defines the common estimated value for all the instances located at the bin B_i. The loss function $\mathcal{L}_{B,\lambda}$ is always differentiable with respect to p_{B_i} unless two calibrated probabilities are just being joined (which only happens if $p_{B_i} = p_{B_{i+1}}$ for some i). Assuming that $\hat{p}_{B_i}(\lambda)$ is optimal, the partial derivative of $\mathcal{L}_{B,\lambda}$ has to be 0 at $\hat{p}_{B_i}(\lambda)$, which implies:

$$|B_i|\hat{p}_{B_i}(\lambda) - \sum_{j\in B_i} z_j + \lambda(\nu_i - \nu_{i-1}) = 0 \text{ for } i = 1, \ldots, N_\lambda \tag{3.17}$$

Rewriting the above equation, the optimum predicted value for each bin can be calculated as:

$$\hat{p}_{B_i}(\lambda) = \frac{\sum_{j\in B_i} z_j - \lambda\nu_i + \lambda\nu_{i-1}}{|B_i|} \text{ for } i = 1, \ldots, N_\lambda \tag{3.18}$$

While PAVA uses the frequency of instances in each bin as the calibrated estimate, Equation 3.18 shows that mPAVA uses a shrunken version of the frequencies by considering the estimates that are not following the ranking imposed by the base classifier. In Equation 3.17, taking derivatives with respect to λ yields:

$$\frac{\partial\hat{p}_{B_i}}{\partial\lambda} = \frac{\nu_{i-1} - \nu_i}{|B_i|}, \text{ for } i = 1, \ldots, N_\lambda, \tag{3.19}$$

where we set $\nu_0 = \nu_N = 0$ for notational convenience. As we noted above, it has been proven that the optimal values of the instances located in the same bin are tied together and the only way that they can change is to merge two bins as they can never split apart as the λ increases (Tibshirani et al., 2011). Therefore, as we make changes in λ, the bins B_i, and hence the values ν_i remain constant. This implies the term $\frac{\partial \hat{p}_{B_i}}{\partial \lambda}$ is a constant in Equation 3.19. Consequently, the solution path remains piecewise linear as λ increases, and the breakpoints happen when two bins merge together. Now, using the piecewise linearity of the solution path and assuming that the two bins B_i and B_{i+1} are the first two bins to merge by increasing λ, the value of $\lambda_{i,i+1}$ at which the two bins B_i and B_{i+1} will merge is calculated as:

$$\lambda_{i,i+1} = \frac{\hat{p}_{B_i}(\lambda) - \hat{p}_{B_{i+1}}(\lambda)}{a_{i+1} - a_i} + \lambda \text{ for } i = 1, \dots N_\lambda - 1, \tag{3.20}$$

where $a_i = \frac{\partial \hat{p}_{B_i}}{\partial \lambda}$ is the slope of the changes of $\hat{p}_{B_i}$ with respect to λ according to Equation 3.19. Using the above identity, the λ at which the next breakpoint occurs is obtained using the following equation:

$$\begin{aligned} \lambda^* &= \min_i \lambda_{i,i+1} \\ \mathbb{I}^* &= \{i | \lambda_{i,i+1} = \lambda^*\}, \end{aligned} \tag{3.21}$$

where $\mathbb{I}^*$ indicates the set of the indexes of the bins that will be merged by their consecutive bins changing the λ[7]. If $\lambda^* < \lambda$ then the algorithm will terminate since it has obtained the standard isotonic regression solution, and by increasing λ none of the existing bins will ever merge. Having the solutions of the near isotonic regression problem in Equation 3.15 at the breakpoints, and using the piecewise linearity property of the solution path, it is possible to recover the solution for any value of λ through interpolation. However, the current implementation of ENIR only uses the near isotonic regression based calibration models that corresponds to the value of λ at the breakpoints. The sketch of the algorithm is shown as Algorithm 2.

[7]Note that there could be more than one bin achieving the minimum in Equation 3.21, so they should be all merged with the bins that are located next to them.

3.4 CALIBRATION USING LINEAR TREND FILTERING

In all the classifier calibration methods, the post-processing step can be seen as a mapping function that transforms the outputs of a classification model to probabilities that are intended to be well-calibrated. In all of the histogram binning-based calibration models —including quantile binning (Zadrozny and Elkan, 2001b), isotonic-regression-based calibration (IsoReg) (Zadrozny and Elkan, 2002), and our previous Bayesian extensions to the histogram binning, ABB and BBQ, (Pakdaman Naeini et al., 2015c,a) — the generated mapping function will be a piecewise constant function. In this section we introduce the *ensemble of linear trend estimation* (ELiTE) calibration method that has the following three main advantages relative to all the above histogram binning-based calibration methods: (1) ELiTE assumes that the calibration mapping function is piecewise linear while the mapping found by quantile binning, IsoReg, ABB, and BBQ are always piecewise constant, (2) ELiTE removes the restrictive assumption that probability estimates are independent between the neighboring bins, and (3) ELiTE automatically finds the boundary of the bins through an optimization algorithm by trading off the best fit of the training instances for the tendency to follow the same trend in probability estimates. This trade-off will be controlled by a regularization parameter.

In order to describe the details of the method, we use similar notation used in describing $ENIR$: Let y_i and z_i define respectively an uncalibrated classifier prediction and the true class of the i'th instance. In this chapter, we focus on calibrating a binary classifier's output[8], and thus, $z_i \in \{0, 1\}$ and $y_i \in [0, 1]$. Without loss of generality, we can assume that the instances are sorted based on the classifier scores y_i, so we have $y_1 < y_2 < \ldots < y_N$, where N is the total number of samples in the training data. Borrowing the term "bin" from the histogram binning literature, we define each bin as the largest interval over the training data with a uniform slope of change. The problem of finding an optimum piecewise linear calibration mapping can be formulated as the

[8]For classifiers that output scores that are not in the unit interval (e.g., SVM), we use a simple sigmoid transformation $f(x) = \frac{1}{1+\exp(-x)}$ to transform the scores into the unit interval.

following optimization program:

$$\begin{aligned} \hat{\boldsymbol{p}} = \underset{\boldsymbol{p} \in R^N}{\text{argmin}} \quad & \frac{1}{2}\sum_{i=1}^{N}(p_i - z_i)^2 \\ \text{s.t.} \quad & ||\mathbf{v}||_0 \leq B - 1 \end{aligned} \tag{3.22}$$

where $||\mathbf{v}||_0 = \sum_i 1(v_i \neq 0)$ is ℓ_0 norm defined as the number of nonzero elements of the vector $\mathbf{v}$. Also, the vector $\mathbf{v} \in R^{N-2}$ is defined as the second order finite difference vector associated with the training data [9] $\mathbf{v}_i = \frac{p_{i+2}-p_{i+1}}{y_{i+2}-y_{i+1}} - \frac{p_{i+1}-p_i}{y_{i+1}-y_i}$, and B is an optimization parameter that is defined as the maximum number of bins that we could have over all the training data (Thus, $B-1$ shows the number of change points or kinks in the calibration mapping function). The above optimization program tries to keep the estimated probability p_i close to z_i, the true class of the corresponding training instance, while the program constrains the number of kinks or change points in the slope of the calibration mapping [10]. Solving the above optimization program is intractable and requires combinatorial optimization methods (Kim et al., 2009). A natural convex relaxation of this problem can be obtained by substituting the ℓ_0 norm with the ℓ_1 norm using the sparsity property of the ℓ_1 norm. After relaxing the ℓ_0 norm, it is possible to rewrite the resulting constrained optimization program in the following equivalent Lagrangian form:

$$\hat{\boldsymbol{p}} = \underset{\boldsymbol{p} \in R^N}{\text{argmin}} \quad \frac{1}{2}\sum_{i=1}^{N}(p_i - z_i)^2 + \lambda||\mathbf{v}||_1 \tag{3.23}$$

where $\hat{\boldsymbol{p}} = (\hat{p}_1, \ldots, \hat{p}_n)$ is the vector of calibrated estimates and $||\mathbf{v}||_1 = \sum_{i=1}^{N} |\frac{p_{i+2}-p_{i+1}}{y_{i+2}-y_{i+1}} - \frac{p_{i+1}-p_i}{y_{i+1}-y_i}|$. Also, λ is a positive real number that regulates the trade-off between the complexity of the model and the goodness of fit by penalizing the total variation over the slope of the resulting calibration mapping function. The above optimization program is equivalent to the ℓ_1 (linear) trend filtering signal approximation model (Kim et al., 2009). The linear trend filtering itself is a special case of the recently introduced adaptive piecewise polynomial trend filtering model (Tibshirani, 2014) [11].

[9] Note that an element of $\mathbf{v}$ is zero when the slope remains the same between two successively predicted points.

[10] If some of the training instances obtain equal classification scores, they will be replaced by an instance with the target value z that is equal to the average of their corresponding z_i. In this case, we form a weighted objective in the optimization program in Equation 3.22

[11] Note that the adaptive piecewise polynomial trend filtering model is itself a special case of generalized lasso problem (Tibshirani and Taylor, 2011)

The piecewise linear trend filtering estimation has the following properties that make it an attractive choice for estimating the calibration mapping function: (1) The final solution to the optimization program $\hat{\mathbf{p}}$ will be a continuous piecewise linear function with the change points occurring on the training data (Kim et al., 2009), so the final calibration mapping function will be a continuous function of uncalibrated scores y_i, and the estimated probabilities will not have any abrupt changes at the boundary of the bins, (2) Due to shrinkage property of the lasso-based penalties, the final probability estimates in two neighboring bins will shrink toward each other (Tibshirani and Taylor, 2011), as a result it will relax the restrictive independence assumption made in histogram binning-based calibration models, (3) The solution path to the optimization program in Equation 3.23 is piecewise linear with respect to the regularization parameter λ. This will make it computationally efficient to find the entire path of the solutions to the trend filtering problem for small sample sizes (Tibshirani and Taylor, 2011).

There are a few different methods to solve the trend filtering optimization problem: It is possible to convert the trend filtering problem into the standard lasso problem and then use the LARS algorithm to find all the solution paths with respect to λ (Efron et al., 2004; Tibshirani and Taylor, 2011). It is also possible to cast the problem as a special case of the generalized lasso signal approximation (Tibshirani and Taylor, 2011), and then derive the dual program and utilize the piecewise linearity property of the solution path in the dual program to find the entire path of the solutions (Tibshirani and Taylor, 2011). However, these two methods do not scale well for large N (Tibshirani, 2014). Another approach is to use coordinate descent methods to solve the dual program of the generalized lasso problem (Tibshirani, 2014). There are two specialized optimization methods designed to solve the trend filtering optimization problem. The first method is based on the specialized interior-point method optimization that was proposed by Kim et al. (Kim et al., 2009). The method requires $O(N)$ computations to solve a banded linear system of equations in each iteration of the interior-point optimization algorithm; in the worst-case it will solve the trend filtering for a single value of λ in $O(N^{1.5})$. However, the authors claim that in practice the interior-point method converges in tens of iterations, in which case the general running time for solving the optimization problem will still be $O(N)$.

The other specialized optimization method for trend filtering problem is recently proposed by Ramdas and Tibshirani (2014). They introduced a specialized *alternating direction method of mul-*

tipliers (ADMM), and they showed that their method has better scalability and faster convergence rate for large scale problems compared to the interior-point based method, while on the small sample sizes they have similar performance to the interior-point based method proposed by Kim et al. (2009). In our implementation of ELiTE, we use the specialized ADMM optimization method [12]. For the sake of completeness, we briefly describe the method; more detailed information about the algorithm and the derivations can be found in Ramdas and Tibshirani (2014).

In order to solve the trend filtering problem in Equation 3.23, the specialized ADMM method introduces a new auxiliary parameter $\boldsymbol{\alpha}$ to rewrite the unconstrained optimization program in Equation 3.23 as the following constrained optimization program:

$$\begin{aligned} \hat{\boldsymbol{p}} = \underset{\boldsymbol{p}\in R^N, \alpha\in R^{N-1}}{\text{argmin}} \quad & \frac{1}{2}||\boldsymbol{p}-\boldsymbol{z}||_2^2 + \lambda||D_{N-1}\boldsymbol{\alpha}||_1 \\ \text{s.t.} \quad & \alpha = A\mathbf{p}, \end{aligned} \tag{3.24}$$

where $D_k \in R^{k-1\times k}$ is defined as follows:

$$D_k = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix},$$

and, $A = diag(\frac{1}{y_2-y_1}, \frac{1}{y_3-y_2}, \dots, \frac{1}{y_N-y_{N-1}})D_N$. The corresponding augmented Lagrangian of the optimization program in Equation 3.24 will be as $L(\mathbf{p}, \boldsymbol{\alpha}, \mathbf{u}) = \frac{1}{2}||\boldsymbol{p}-\boldsymbol{z}||_2^2 + \lambda||D_{N-1}\boldsymbol{\alpha}||_1 + \frac{\rho}{2}||\boldsymbol{\alpha} - A\mathbf{p} + \mathbf{u}||_2^2 - \frac{\rho}{2}||\mathbf{u}||_2^2$. Using the augmented Lagrangian and performing some calculations (Boyd et al., 2011), the ADMM iterations will be as follows:

$$\begin{aligned} \mathbf{p} &\leftarrow (I + \rho A^T A)^{-1}(\mathbf{z} + \rho A^T(\boldsymbol{\alpha} + \mathbf{u})) \\ \boldsymbol{\alpha} &\leftarrow \underset{\alpha\in R^{N-1}}{\text{argmin}} \frac{1}{2}||A\mathbf{p} - \mathbf{u} - \boldsymbol{\alpha}||_2^2 + \frac{\lambda}{\rho}||D_{N-1}\boldsymbol{\alpha}||_1 \\ \mathbf{u} &\leftarrow \mathbf{u} + \boldsymbol{\alpha} - A\mathbf{p} \end{aligned}$$

The tricky part in the above sequential updates is the second equation related to updating the value of $\boldsymbol{\alpha}$. It requires solving an optimization problem that is equivalent to the fused lasso signal

[12] The specialized ADMM code is publicly available at https://github.com/statsmaths/glmgen

approximation (Tibshirani et al., 2005). In implementing the specialized ADMM method, Ramdas et al. used a computationally efficient dynamic programming method proposed by Johnson (2013) that finds the fused lasso solution in $O(N)$. They reported that ADMM iterations converge in a constant number of iterations. As a result, the ultimate time for finding the trend filtering solution will still be $O(N)$ (Ramdas and Tibshirani, 2014).

ELiTE employs the specialized ADMM optimization method just described to generate a collection of trend filtering models (one for each value of λ ranging equally in the log space from λ_{max} to $\lambda_{max} * 10^{-4}$, where λ_{max} is the corresponding value of λ that gives the best affine approximation of the calibration mapping that is $\lambda_{max} = ||(D_{N-1}AA^TD_{N-1}^T)^{-1}D_{N-1}A\mathbf{z}||_\infty$ (Kim et al., 2009)). It then uses the Akaike information criterion with a correction for finite sample sizes (AICc) (Cavanaugh, 1997) to score each of the models [13]. We use the unbiased estimate of the degree of freedom for each linear trend filtering model as the effective number of parameters in computing the scores (Tibshirani, 2014). Assume ELiTE yields the piecewise linear calibration models $M_1, M_2, \ldots, M_T$, where T is the total number of generated models by changing λ (in our experiments $T = 50$). For any new classifier output y, the calibrated prediction in the *ELiTE* model is defined using the following weighted averaging (Hoeting et al., 1999):

$$P(z = 1|y) = \sum_{i=1}^{T} \frac{Score(M_i)}{\sum_{j=1}^{T} Score(M_j)} P(z = 1|y, M_i),$$

where $P(z = 1|y, M_i)$ is the probability estimate obtained using the trend filter model M_i for the uncalibrated classifier output y. Also, $Score(M_i)$ is obtained using the AICc scoring function (Schwarz et al., 1978). A nice property of using ELiTE in calibrating binary classification models, is its ability to include other type of prior knowledge to the optimization program through new constraints. For instance, it is possible to add the near-isotonicity constraints to the linear trend filtering optimization program to find a linear trend calibration mapping which is nearly isotonic. This can be done by simply adding the near-isotonic constraint to our formulation (Ramdas and Tibshirani, 2014).

[13] We also tried BIC and AIC model scoring functions. The AIC scoring shows extreme overfitting to the training data, while BIC results were comparable to AICc scoring. We finally chose AICc since it performed slightly better than BIC in general.

input : $D = \{(y_1, z_1), \ldots, (y_N, z_N)\}$
output : (1) a set of binning models $M_1, \ldots, M_T$,
(2) their corresponding scoring $S_1, \ldots, S_T$
Invariant: Pairs are sorted based on y_i
$\lambda \leftarrow 0$;
$\lambda^* \leftarrow 0$;
$t \leftarrow 1$;
$N_\lambda = N$;
for $i \leftarrow 1$ **to** N **do**
  $B_i = \{i\}$;
  $p_i = z_i$;
end
while $\lambda^* = \lambda$ **do**
  Update the slopes a_i using Equation 3.19;
  Update merging values $\lambda_{i,i+1}$ using Equation 3.20;
  Compute λ^* and $\mathbb{I}^*$ using Equation 3.21;
  if $\lambda^* \leq \lambda$ **then**
    terminate ;
  end
  for $i \leftarrow 1$ **to** N_λ **do**
    //update corresponding probability estimate as:
    $\hat{p}_{B_i}(\lambda^*) = \hat{p}_{B_i}(\lambda) + a_i \times (\lambda^* - \lambda)$;
  end
  Merge appropriate bins as indicated in the set $\mathbb{I}^*$;
  Update number of bins N_λ;
  Store the corresponding calibration model in M_t;
  Store the score of the calibration model in S_t;
  $\lambda \leftarrow \lambda^*$;
  $t \leftarrow t + 1$;
end

Algorithm 2: The *modified pool adjacent violators algorithm* (mPAVA) that yields a set of near-isotonic-regression-based calibration models $M_1, \ldots, M_T$

4.0 EMPIRICAL RESULTS ON BINARY CLASSIFIER CALIBRATION METHODS

This chapter presents the results and findings of our newly introduced binary classifier calibration methods compared with several commonly used calibration methods including Platt's method, histogram binning, and isotonic regression. We will present the results in seven main sections. Section 4.1 presents the experimental results for our proposed calibration methods based on KDE and DPM. Section 4.2.1 shows the results of our experiments on SBB and ABB. The results are published in the SIAM Data Mining (SDM) 2015 (Pakdaman Naeini et al., 2015b). Section 4.2.2 presents our results and findings on utilizing BBQ for calibrating binary classifiers, these set of the results have been published at AAAI 2015 (Pakdaman Naeini et al., 2015a). Section 4.3 presents the results of our proposed binary classifier calibration method ENIR, these set of the results have been submitted to ICDM 2016. Section 4.4 presents the experimental result of our newly introduced binary classifier calibration method ELiTE, these set of the results have been published at SDM 2016 (Pakdaman Naeini et al., 2015a). Section 4.2.3 present the set of experiments in comparing the two Bayesian extensions that we introduced to the histogram binning calibration method. Finally, Section 4.5 concludes this chapter by presenting the set of experiments we performed to compare our newly introduced binary classifier calibration methods including: KDE based calibration method, BBQ, ENIR, and ELiTE.

4.1 EXPERIMENTAL RESULTS ON KDE-DPM

This section describes the set of experiments that we performed to evaluate the performance of calibration methods based KDE and DPM described in section 3.1. To evaluate the calibration performance of each method, we ran experiments on both simulated and on real data. For the

Table 3: Experimental Results on Simulated dataset 5(b)

(a) SVM Linear

	SVM	Hist	Platt	IsoReg	KDE	DPM
RMSE	0.50	0.39	0.50	0.46	0.38	0.39
AUC	0.50	0.84	0.50	0.65	0.85	0.85
ACC	0.48	0.78	0.52	0.64	0.78	0.78
MCE	0.52	0.19	0.54	0.58	0.09	0.16
ECE	0.28	0.07	0.28	0.35	0.03	0.07

(b) SVM Quadratic Kernel

	SVM	Hist	Platt	IsoReg	KDE	DPM
RMSE	0.21	0.09	0.19	0.08	0.09	0.08
AUC	1.00	1.00	1.00	1.00	1.00	1.00
ACC	0.99	0.99	0.99	0.99	0.99	0.99
MCE	0.35	0.04	0.32	0.03	0.07	0.03
ECE	0.14	0.01	0.15	0.00	0.01	0.00

evaluation of the calibration methods, we used 5 different measures. The first two measures are Accuracy (ACC) and the Area Under the ROC Curve (AUC), which measure discrimination. The three other measures are the Root Mean Square Error (RMSE), Expected Calibration Error (ECE), and Maximum Calibration Error (MCE), which measure calibration.

Simulated data. For the simulated data experiments, we used a binary classification dataset in which the outcomes were not linearly separable. The scatter plot of the simulated dataset is shown in Figure 5(b). The data were divided into 1000 instances for training and calibrating the prediction model, and 1000 instances for testing the models.

To conduct the experiments on simulated datasets, we used two extreme classifiers: support vector machines (SVM) with linear and quadratic kernels. The choice of SVM with a linear kernel allows us to see how the calibration methods perform when the classification model makes over simplifying (linear) assumptions. Also, to achieve good discrimination on the data in figure 5(b), SVM with quadratic kernel is intuitively an ideal choice. So, the experiment using quadratic kernel SVM allows us to see how well different calibration methods perform when we use an ideal learner for the classification problem, in terms of discrimination.

As seen in Table 3, KDE and DPM based calibration methods performed better than Platt and isotonic regression in the simulation datasets, especially when the linear SVM method is used as the base learner. The poor performance of Platt is not surprising given its simplicity, which consists of a parametric model with only two parameters. However, isotonic regression is a nonparametric

model that only makes a monotonicity assumption over the output of base classifier. When we use a linear kernel SVM, this assumption is violated because of the non-linearity of data. As a result, isotonic regression performs relatively poorly, in terms of improving the discrimination and calibration capability of a base classifier. The violation of this assumption can happen in real data as well. In order to mitigate this pitfall, Menon et al. (2012) proposed using a combination of optimizing AUC as a ranking loss measure, plus isotonic regression for building a ranking model. However, this is counter to our goal of developing post-processing methods that can be used with any existing classification models. As shown in Table 3(b), even if we use an ideal SVM classifier for these linearly non-separable datasets, our proposed methods perform better or comparable isotonic regression based calibration.

As can be seen in Table 3(b), although the SVM base learner performs very well in the sense of discrimination based on AUC and ACC measures, it performs poorly in terms of calibration, as measured by RMSE, MCE, and ECE. Moreover, all of the calibration methods retain the same discrimination performance that was obtained prior of post-processing, while improving calibration.

Real data. In terms of real data, we used a KDD-98 dataset , which is available from the UCI KDD repository. The dataset contains information about people who donated to a particular charity. Here the decision making task is to decide whether a solicitation letter should be mailed to a person or not. The letter costs $\$0.68$. The training set includes $95,412$ instances in which it is known whether a person made a donation, and if so, how much the person donated. Among all these training cases, $4,843$ were responders. The validation set includes $96,367$ instances from the same donation campaign of which $4,873$ where responders.

Following the procedure in (Zadrozny and Elkan, 2001b, 2002), we built two models: a *response model* $r(x)$ for predicting the probability of responding to a solicitation, and an *amount model* $a(x)$ for predicting the amount of donation of person x. The optimal mailing policy is to send a letter to those people for whom the expected donation return $r(x)a(x)$ is greater than the cost of mailing the letter. Since in this research we are not concerned with feature selection, our choice of attributes are based on (Mayer and Sarkissian, 2003) for building the response and amount prediction models. Following the approach in (Zadrozny and Elkan, 2001a), we built the amount model on the positive cases in the training data, removing the cases with more than $\$50$ as outliers. Following their construction, we also provided the output of the response model $r(x)$ as

an augmented feature to the amount model $a(x)$.

In our experiments, in order to build the response model, we used three different classifiers: SVM, $LogisticRegression$ and $naiveBayes$. For building the amount model, we also used a support vector regression model. For implementing these models, we used the liblinear package (Fan et al., 2008). The results of the experiment are shown in Table 4. In addition to previous measures of comparison, we also show the amount of profit obtained when using different methods. As seen in these tables, the application of calibration methods results in at least $\$3000$ more in expected net gain from sending solicitations. This result also supports the general point made in Chapter 1 about the importance of using calibrated probabilities in solving decision analysis problems.

4.2 EXPERIMENTAL RESULTS FOR BAYESIAN BINNING METHODS

This section presents the results of our experiments on evaluating the performance of our Bayesian extensions to histogram binning calibration method. Section 4.2.1 presents the empirical results for evaluating the performance of the selective Bayesian binning (SBB) and averaging Bayesian binning (ABB) models that presented in Section 3.2.1. Section 4.2.1 presents the results of experiments on the Bayesian Binning into Quantile (BBQ) method. Section 4.2.3 presents the results of experiments on comparing the performance of ABB versus BBQ. Finally, Section 4.2.4 presents the results of experiments on comparing performance of the two scoring functions K2 and BDeu in building our Bayesian binning calibration methods.

4.2.1 Experimental Results for ABB-SBB

This section describes the set of experiments that we performed to evaluate the calibration methods described in Section 3.2.1. To evaluate the calibration performance of each method, we ran experiments using both simulated data and real data. In our experiments on simulated data, we used logistic regression (LR) as the base classifier, whose predictions are to be calibrated. The choice of logistic regression was made to let us compare our results with the state-of-the-art method ACP,

Table 4: Experimental Results on KDD 98 dataset. The first column of each table shows the result of a model without post-processing calibration

(a) Logistic Regression

	LR	Hist	Plat	IsoReg	KDE	DPM
RMSE	0.500	0.218	0.218	0.218	0.218	0.219
AUC	0.613	0.610	0.613	0.612	0.611	0.613
ACC	0.56	0.95	0.95	0.95	0.95	0.95
MCE	0.454	0.020	0.013	0.030	0.004	0.017
ECE	0.449	0.007	0.004	0.013	0.002	0.003
Profit	10560	13183	13444	13690	12998	13696

(b) Naïve Bayes

	NB	Hist	Plat	IsoReg	KDE	DPM
RMSE	0.514	0.218	0.218	0.218	0.218	0.218
AUC	0.603	0.600	0.603	0.602	0.602	0.603
ACC	0.622	0.949	0.949	0.949	0.949	0.949
MCE	0.850	0.008	0.008	0.046	0.005	0.010
ECE	0.390	0.004	0.004	0.023	0.002	0.003
Profit	7885	11631	10259	10816	12037	12631

(c) SVM Linear

	SVM	Hist	Plat	IsoReg	KDE	DPM
RMSE	0.696	0.218	0.218	0.219	0.218	0.218
AUC	0.615	0.614	0.615	0.500	0.614	0.615
ACC	0.95	0.95	0.95	0.95	0.95	0.95
MCE	0.694	0.011	0.013	0.454	0.003	0.019
ECE	0.660	0.004	0.004	0.091	0.002	0.004
Profit	10560	13480	13080	11771	13118	13544

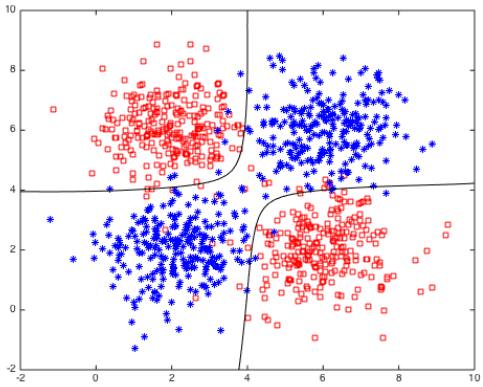
which as published is tailored for LR. For the simulated data, we used two synthetic datasets in which the outcomes were not linearly separable. The scatter plots of the two simulated datasets are shown in Figures 5(a), 5(b). These extreme choices allow us to see how well the calibration methods perform when the classification model makes over simplifying (linear) assumptions in learning non-linear concepts. In the simulation data we used 600 randomly generated instances for training the LR model, 600 random instances for learning calibration-models, and 600 random instances for testing the models [1].

We also performed experiments on three different sets of real binary classification data. The first set is the UCI Adult dataset. The prediction task is a binary classification problem to predict whether a person makes over $50K a year using his or her demographic information. From the original Adult dataset, which includes 48842 total instances with 14 real and categorical features, we used randomly 2000 instances for training classifiers, 600 for calibration-model learning, and 600 instances for testing.

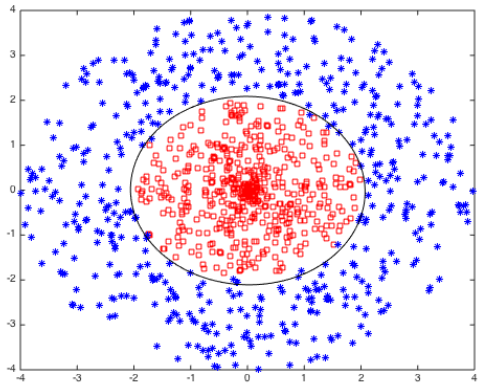
We also used the UCI SPECT dataset, which is a small biomedical binary classification dataset. SPECT allows us to examine how well each calibration method performs when the calibration dataset is small in a real application. The dataset involves the diagnosis of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified into two categories: normal or abnormal. This dataset consists of 80 training instances, with an equal number of positive and negative instances, and 187 test instances with only 15 positive instances. The SPECT dataset includes 22 binary features. Due to the small number of instances, we used the original training data as both our training and calibration datasets, and we used the original test data as our test dataset.

For the experiments on the Adult and SPECT datasets, we used three different classifiers: LR, naïve Bayes, and SVM with a polynomial kernels. The choice of the LR model allows us to include the ACP method in the comparison, because as mentioned it is tailored to LR. Naïve Bayes is a well-known, simple, and practical classifier that often achieves good discrimination performance, although it is usually not well calibrated. We included SVM because it is a relatively modern

[1] Based on our experiments, the separation between training set and calibration set is not necessary. However, Zadrozny and Elkan (2001b) states that for the histogram model it is better to use another set of instances for calibrating the output of classifier in order to prevent overfitting; thus, we do so in our experiments.



(a) Scatter plot of XOR configuration



(b) Scatter plot of Circular configuration

Figure 5: Scatter plots of the simulated data: The top panel shows the XOR configuration dataset in which the black curves indicate the decision boundaries found by using SVM with RBF kernel. The bottom panel shows the circular configuration dataset in which the black oval indicates the decision boundary found using SVM with a quadratic kernel.

classifier that is being frequently applied [2].

The other real dataset that we used for evaluation contains clinical findings (e.g., symptoms, signs, laboratory results) and outcomes for patients with community acquired pneumonia (CAP) (Fine et al., 1997). The classification task we examined involves using patient findings to predict dire patient outcomes, such as mortality or serious medical complications. The CAP dataset includes a total of 2287 patient cases (instances) that we divided into 1087 instances for training of classifiers, 600 instances for learning calibration models, and 600 instances for testing the calibration models. The data includes 172 discrete and 43 continuous features. For our experiments on the naïve Bayes model, we just used the discrete features of data, and for the experiments on SVM we used all 215 discrete and continuous features. Also, for applying the LR model to this dataset, we first used the PCA feature transformation because of the high dimensionality of data and the existing correlations among some features, which produced unstable results due to singularity issues.

The Tables $5(a), 5(b), \ldots, 5(j)$ show the comparisons of different methods with respect to evaluation measures on the simulated and real datasets. In these tables in each row we show in bold the two methods that achieved the best performance with respect to a specified measure.

As can be seen, there is no superior method that outperforms all the others on all dataset s on all measures. However, SBB and ABB are superior to Platt and isotonic regression in all the simulation datasets. We discuss the reason why below. Also, SBB and ABB perform as well or better than isotonic regression and the Platt method on the real dataset s.

In all of the experiments, both on simulated datasets and real dataset s, both SBB and ABB generally retain or improve the discrimination performance of the base classifier, as measured by ACC and AUC. In addition, they often improve the calibration performance of the base classifier in terms of the $RMSE$, ECE and MCE measures.

4.2.1.1 Discussion Having a well-calibrated classifier can be important in practical machine learning problems. There are different calibration methods in the literature and each one has its own pros and cons. The Platt's method uses a sigmoid as a mapping function. The parameters of the sigmoidal transformation function are learned using a maximum likelihood estimation framework.

[2] The output of the SVM model is mapped to the interval $[0, 1]$ using a simple sigmoid function

Table 5: Experimental Results on Simulated and Real datasets

(a) Non-Linear XOR configuration results

	LR	ACP	IsoReg	Platt	Hist	SBB	ABB
AUC	0.497	**0.950**	0.704	0.497	0.931	0.914	**0.941**
ACC	0.510	**0.887**	0.690	0.510	0.855	**0.887**	**0.888**
RMSE	0.500	**0.286**	0.447	0.500	0.307	0.307	**0.295**
MCE	0.521	**0.090**	0.642	0.521	0.152	0.268	**0.083**
ECE	0.190	**0.056**	0.173	0.190	0.072	0.104	**0.062**

(b) Non-Linear Circular configuration results

	LR	ACP	IsoReg	Platt	Hist	SBB	ABB
AUC	0.489	**0.852**	0.635	0.489	0.827	0.816	**0.838**
ACC	0.500	0.780	0.655	0.500	**0.795**	**0.790**	0.773
RMSE	0.501	**0.387**	0.459	0.501	0.394	0.393	**0.390**
MCE	0.540	0.172	0.608	0.539	**0.121**	0.790	**0.146**
ECE	0.171	0.098	0.186	0.171	**0.074**	0.138	**0.091**

(c) Adult Naïve Bayes

	NB	IsoReg	Platt	Hist	SBB	ABB
AUC	**0.879**	0.876	**0.879**	0.877	0.849	**0.879**
ACC	0.803	0.822	**0.840**	0.818	**0.838**	0.835
RMSE	0.352	**0.343**	**0.343**	**0.341**	0.345	**0.343**
MCE	0.223	0.302	**0.092**	0.236	0.373	**0.136**
ECE	0.081	0.075	**0.071**	0.078	0.114	**0.062**

(d) Adult Linear SVM

	SVM	IsoReg	Platt	Hist	SBB	ABB
AUC	**0.864**	0.856	**0.864**	**0.864**	0.821	**0.864**
ACC	0.248	**0.805**	0.748	**0.815**	0.803	**0.805**
RMSE	0.587	0.360	0.434	**0.355**	0.362	**0.357**
MCE	0.644	0.194	0.506	**0.144**	0.396	**0.110**
ECE	0.205	0.085	0.150	**0.077**	0.108	**0.061**

(e) Adult Logistic Regression

	LR	ACP	IsoReg	Platt	Hist	SBB	ABB
AUC	0.730	0.727	**0.732**	0.730	0.743	0.699	0.731
ACC	0.755	**0.783**	0.753	0.755	0.753	0.762	**0.762**
RMSE	0.403	0.402	0.403	0.405	0.400	**0.401**	**0.401**
MCE	**0.126**	0.182	0.491	0.127	0.274	0.649	**0.126**
ECE	**0.075**	**0.071**	0.118	0.079	0.092	0.169	0.076

(f) SPECT Naïve Bayes

	NB	IsoReg	Platt	Hist	SBB	ABB
AUC	**0.836**	0.815	**0.836**	0.832	0.733	0.835
ACC	0.759	**0.845**	0.770	0.824	**0.845**	**0.845**
RMSE	0.435	**0.366**	0.378	0.379	**0.368**	0.374
MCE	0.719	0.608	0.563	0.712	**0.347**	**0.557**
ECE	0.150	**0.141**	0.148	**0.145**	0.149	0.157

(g) SPECT SVM Quadratic kernel

	SVM	IsoReg	Platt	Hist	SBB	ABB
AUC	**0.816**	0.786	**0.816**	0.766	0.746	0.810
ACC	0.257	**0.834**	0.684	**0.845**	0.813	0.813
RMSE	0.617	0.442	0.460	0.463	**0.398**	**0.386**
MCE	**0.705**	**0.647**	0.754	0.934	0.907	0.769
ECE	0.235	0.148	0.162	0.180	**0.128**	**0.131**

(h) SPECT Logistic Regression

	LR	ACP	IsoReg	Platt	Hist	SBB	ABB
AUC	**0.744**	0.742	0.733	**0.744**	0.738	0.733	0.741
ACC	**0.658**	0.561	0.626	**0.668**	0.620	0.620	0.626
RMSE	0.546	0.562	0.558	0.524	0.565	**0.507**	**0.496**
MCE	0.947	1.000	1.000	0.884	0.997	**0.813**	**0.812**
ECE	0.181	0.187	0.177	0.180	0.183	**0.171**	**0.173**

(i) CAP Naïve Bayes

	NB	IsoReg	Platt	Hist	SBB	ABB
AUC	**0.848**	0.845	**0.848**	0.831	0.775	0.838
ACC	0.730	**0.865**	0.847	0.853	0.832	**0.865**
RMSE	0.504	**0.292**	0.324	0.307	0.315	**0.304**
MCE	0.798	0.188	0.303	**0.087**	0.150	**0.128**
ECE	0.161	0.071	0.097	**0.056**	**0.067**	**0.067**

(j) CAP Linear SVM

	SVM	IsoReg	Platt	Hist	SBB	ABB
AUC	**0.858**	**0.858**	**0.858**	0.847	0.813	**0.863**
ACC	**0.907**	0.900	0.882	0.887	0.902	**0.908**
RMSE	0.329	**0.277**	0.294	0.287	0.285	**0.274**
MCE	0.273	**0.114**	0.206	**0.110**	0.240	0.121
ECE	0.132	0.058	0.093	**0.057**	0.083	**0.050**

(k) CAP Logistic Regression

	LR	ACP	IsoReg	Platt	Hist	SBB	ABB
AUC	**0.920**	0.910	0.917	**0.920**	0.901	0.856	**0.921**
ACC	0.925	0.932	**0.935**	0.928	0.897	**0.935**	0.932
RMSE	**0.240**	**0.240**	**0.234**	0.242	0.259	**0.240**	**0.240**
MCE	0.199	**0.122**	0.286	**0.154**	0.279	0.391	0.168
ECE	**0.066**	**0.062**	0.078	0.082	0.079	0.103	0.069

The main advantage of the Platt scaling method is its fast recall time. However, the shape of the sigmoid function can be restrictive, and it often cannot produce well calibrated probabilities when the instances are distributed in feature space in a biased fashion (e.g. at the extremes, or all near separating hyper plane) (Jiang et al., 2012).

Table 6: Time complexity of calibration methods in training on N samples and application on 1 test sample

	Platt	**Hist**	**IsoReg**	**ACP**	**SBB**	**ABB**
Time Complexity (Training/Application)	$O(NT)/O(1)$	$O(NlogN)/O(b)$	$O(N)/O(b)$	$O(NlogN)/O(N)$	$O(N^2)/O(b)$	$O(RN^2)/O(1)$

Note that N and b are the size of training sets and the number of bins found by the method respectively. T is the number of iteration required for convergence in Platt's method and R reflects the number of bins being used by cached ABB.

Histogram binning is a non-parametric method which makes no special assumptions about the shape of mapping function. However, it has several limitations, including the need to define the number of bins and the fact that the bins remain fixed over all predictions (Zadrozny and Elkan, 2002). ABB alleviates these problems by performing Bayesian averaging over all set of possible binning models on the training data.

Isotonic regression-based calibration is another non-parametric calibration method which has been shown to perform very well in comparison to other calibration methods in real datasets (Niculescu-Mizil and Caruana, 2005b; Caruana and Niculescu-Mizil, 2006; Zadrozny and Elkan, 2002). However, isotonic regression has some weaknesses. The most significant limitation of the isotonic regression is its isotonicity (monotonicity) assumption. As seen in Tables 5(a), 5(b) in the simulation data, when the isotonicity assumption is violated through the choice of classifier and the nonlinearity of data, isotonic regression performs relatively poorly, in terms of improving the discrimination and calibration capability of a base classifier. The violation of this assumption can happen in real data secondary to the choice of learning models and algorithms, specifically when we encounter large scale classification problems in which we have to make simplifying assumptions to build the learning models. In order to mitigate this pitfall, Menon et al. (2012) proposed a new isotonic based calibration method using a combination of optimizing AUC as a ranking loss measure, plus isotonic regression for building an accurate ranking model. However, this is counter to our goal of developing post-processing methods that can be used with any existing classification

models. There is also another interesting extension of isotonic regression for calibrating the output of multiple classifiers (Zhong and Kwok, 2013), but it is not included in our experiments, since we focus on calibrating the output of a single binary classifier in these experiments.

A classifier calibration method called *adaptive calibration of predictions* (ACP) was recently introduced (Jiang et al., 2012). A given application of ACP is tied to a particular model M, such as a logistic regression model, that predicts a binary outcome Z. ACP requires a 95% confidence interval (CI) around a particular prediction p_{in} of M. ACP adjusts the CI and uses it to define a bin. It sets p_{out} to be the fraction of positive outcomes $(Z = 1)$ among all the predictions that fall within the bin. On both real and synthetic datasets, ACP achieved better calibration performance than a variety of other calibration methods, including simple histogram binning, Platt scaling, and isotonic regression (Jiang et al., 2012). The ACP post-calibration probabilities also achieved among the best levels of discrimination, according to the AUC. ACP has several limitations, however. First, it requires not only probabilistic predictions, but also a statistical confidence interval (CI) around each of those predictions, which makes it tailored to specific classifiers, such as logistic regression (Jiang et al., 2012). Second, based on a CI around a given prediction p_{in}, it commits to a single binning of the data around that prediction; it does not consider alternative binnings that might yield a better calibrated p_{out}. Third, the bin it selects is symmetric around p_{in} by construction, which may not optimize calibration. Finally, it does not use all of the training data, but rather only uses those predictions within the confidence interval around p_{in}. The proposed ABB method mitigates these problems by performing a Bayesian averaging over all set of possible binning models on the training data. As one can see from the tables, ACP performed well when logistic regression is the base classifier, both in simulated and real datasets. Also, as can be seen in the results of our experiments, SBB and ABB performed comparable to ACP in both simulation and real dataset s.

In general, the SBB and ABB algorithms appear promising, especially ABB, which overall outperformed SBB. Neither algorithm makes restrictive (and potentially unrealistic) assumptions, as does Platt scaling and isotonic regression. They also are not restricted in the type of classifier with which they can apply, unlike ACP.

The main disadvantage of SBB and ABB is their running time. If N is the number of training instances, then SBB has a training time of $O(N^2)$, due to its dynamic programming algorithm that searches over every possible binning, whereas the time complexity of ACP, histogram binning,

and isotonic regression are $O(NlogN)$ (Jiang et al., 2012). Also, the cached version of ABB has a training time of $O(RN^2)$, where R reflects the number of bins being used. Nonetheless, it remains practical to use these algorithms to perform calibration on a desktop computer when using training datasets that contain thousands of instances. Note that, the amount of data that is needed to calibrate classification models is much less than the amount needed to train them, because the calibration feature space has only a single dimension [3]. In addition, the testing time is only $O(b)$ for SBB where b is the number of binnings found by the algorithm and $O(1)$ for the cached version of ABB. Table 6 shows the time complexity of different methods in learning for N training instances and recall for only one instance.

4.2.2 Experimental Results for BBQ

This section describes the set of experiments that we performed to evaluate the performance of the proposed calibration method in section 3.2.2 in comparison to other commonly used calibration methods: histogram binning, Platt's method, and isotonic regression. To evaluate the calibration performance of each method, we ran experiments on both simulated and on real data.

4.2.2.1 Simulated Data For the simulated data experiments, we used the simulated binary classification dataset is shown in Figure 5(b). The data were divided into 1000 instances for training and calibrating the prediction model, and 1000 instances for testing the models.

As seen in Tables 7, BBQ outperforms Platt's method and isotonic regression on the simulation dataset, especially when the linear SVM method is used as the base learner. The poor performance of Platt's method is not surprising given its simplicity, which consists of a parametric model with only two parameters. However, isotonic regression is a non-parametric model that only makes a monotonicity assumption over the output of the base classifier. When we use a linear kernel SVM, this assumption is violated because of the non-linearity of data. As a result, isotonic regression performs relatively poorly, in terms of improving the discrimination and calibration capability of the base classifier. As shown in Table 7(b), even if we use an ideal SVM classifier for our linearly non-separable dataset, the proposed method performs as well as an isotonic regression-

[3] It is actually the space of (uncalibrated) classifier's outputs, which is the interval $[0, 1]$

Table 7: Experimental Results on th circular configuration simulated dataset shown in Figure 5(b)

(a) SVM Linear

	SVM	Hist	Platt	IsoReg	BBQ
AUC	0.50	0.84	0.50	0.65	0.85
ACC	0.48	0.78	0.52	0.64	0.78
RMSE	0.50	0.39	0.50	0.46	0.38
ECE	0.28	0.07	0.28	0.35	0.03
MCE	0.52	0.19	0.54	0.58	0.09

(b) SVM Quadratic Kernel

	SVM	Hist	Platt	IsoReg	BBQ
AUC	1.00	1.00	1.00	1.00	1.00
ACC	0.99	0.99	0.99	0.99	0.99
RMSE	0.21	0.09	0.19	0.08	0.08
ECE	0.14	0.01	0.15	0.00	0.00
MCE	0.35	0.04	0.32	0.03	0.03

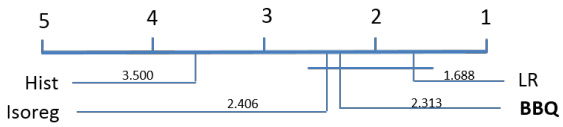
based calibration.

As can be seen in Table 7(b), although the SVM based learner performs very well in terms of discrimination based on AUC and ACC measures, it performs poorly in terms of calibration, as measured by RMSE, MCE, and ECE. Moreover, while improving calibration, all of the calibration methods retain the same discrimination performance that was obtained prior to post-processing.

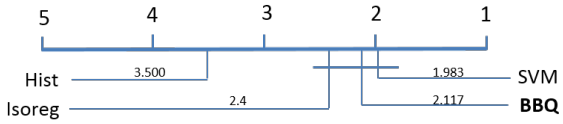
4.2.2.2 Real Data In terms of real data, we used 30 different real world binary classification dataset s from the UCI and LibSVM repository [4] (Bache and Lichman, 2013; Chang and Lin, 2011). We used three common classifiers, namely, Logistic Regression (LR), Support Vector Machines (SVM), and Naive Bayes (NB) to evaluate the performance of the proposed calibration method. To evaluate the performance of calibration models, we used the recommended statistical test procedure by Janez Demsar (Demšar, 2006). More specifically, we used the (modified) Friedman nonparametric hypothesis testing method (Friedman, 1937; Iman and Davenport, 1980)

[4]The datasets used were as follows: spect, breast, adult, pageblocks, pendigits, ad, mamography, satimage, australian, code rna, colon cancer, covtype, letter unbalanced, letter balanced, diabetes, duke, fourclass, german numer, gisette scale, heart, ijcnn1, ionosphere scale, liver disorders, mushrooms, sonar scale, splice, svmguide1, svmguide3, coil2000, balance.

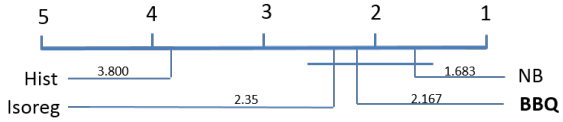
followed by Holm's step-down procedure (Holm, 1979) to evaluate the performance of BBQ in comparison to the other calibration methods across the 30 real dataset s. Next, we briefly describe the test procedure; more detailed information can be found in (Demšar, 2006).



(a) AUC results with LR as classifier



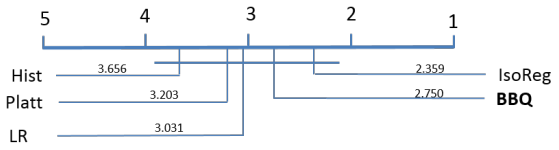
(b) AUC results with SVM as classifier



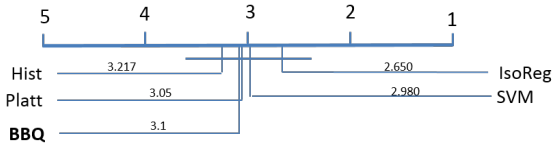
(c) AUC results with NB as classifier

Figure 6: Performance of each method in terms of average rank of AUC on the real datasets. All the methods which are not connected to BBQ by the horizontal bar are significantly different from BBQ (using F_F test followed by Holm's step-down procedure at a 0.05 significance level).

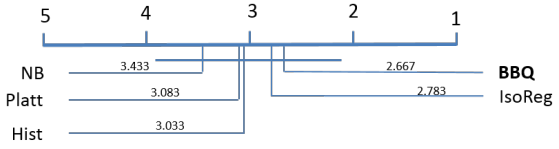
The Friedman test (Friedman, 1937) is a non-parametric version of the ANOVA test. For more concrete description of how the test performs, assume we aim to compare the performance of the calibration methods in terms of $RMSE$ and our base classifier is LR. The Friedman test ranks the RMSE of LR in addition to the RMSE of the calibration methods (Hist, Platt, IsoReg, BBQ) for each dataset separately, with the best performing method getting the rank of 1, the second best the rank of 2, and so on. In case of ties, average ranks are assigned to the corresponding methods. Let $r_{i,j}$ be the rank of i'th of the 5 methods (LR, Hist, Platt, Isoreg, BBQ) at the j'th of the 30 datasets. The Friedman test computes the average rank of each method $R_i = \frac{1}{30}\sum_{j=1}^{30} r_{i,j}$. The null hypothesis states that all the methods are statistically equivalent and so their associated rank



(a) ACC results with LR as classifier



(b) ACC results with SVM as classifier



(c) ACC results with NB as classifier

Figure 7: Performance of each method in terms of average rank of ACC on the real datasets. There is no statistically significant difference between the performance of the methods in terms of ACC (using the F_F test at a 0.05 significance level).

R_i should be equal. Under the null-hypothesis, the Friedman statistic

$$\chi_F^2 = \frac{12N}{k(k+1)}[\sum_j R_j^2 - \frac{k(k+1)^2}{4}]$$

is distributed according to χ_F^2 with $k-1$ degrees of freedom, where N is the number of datasets (30 in our case) and k is the number of methods (5 in our case). However, it is known that the Friedman statistic is often unnecessarily conservative; thus, as suggested in (Demšar, 2006; Iman and Davenport, 1980), we use a more accurate F_F statistic defined as follows:

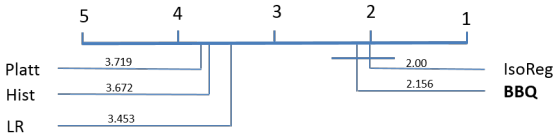
$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2}$$

Under the null hypothesis the F_F statistic is distributed according to the F distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom. If the null hypothesis is rejected, we proceed with Holm's step-down post-hoc test (Holm, 1979) to compare the $RMSE$ of our targeted method (BBQ in our case) to the $RMSE$ of the other methods. In order to use Holm's method, we define the z_i statistics as:

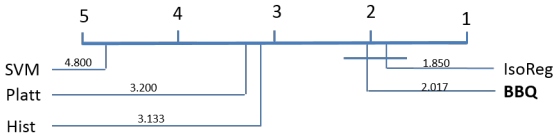
$$z_i = \frac{(R_i - R_{BBQ})}{\sqrt{\frac{k(k+1)}{6N}}},$$

where R_{BBQ} is the average rank of the target method (***BBQ***), R_i is the average rank of i'th method, k is the number of methods, and N is number of datasets. In Holm's method of testing, the z_i statistic is used to find the corresponding p_i value from the table of the normal distribution, which is compared with an adjusted α values as follows. First, the p values are sorted so that $p_{\pi_1} \leq p_{\pi_2} \ldots \leq p_{\pi_{k-1}}$. Then each p_i is compared to $\frac{\alpha}{k-i}$ sequentially. So the most significant p value, p_1, is compared with $\frac{\alpha}{k-1}$. If p_1 is below $\frac{\alpha}{k-1}$, the corresponding hypothesis is rejected and we continue to compare p_2 with $\frac{\alpha}{k-2}$, and so on. As soon as a certain null hypothesis cannot be rejected, all the remaining hypotheses are retained as well. So, if $p_j$is the first p value that is greater than $\frac{\alpha}{k-j}$, then we conclude that the rank of our target method BBQ is significantly different from the methods $\pi_1, .., \pi_{j-1}$, and it is statistically equivalent to the rest of the methods.

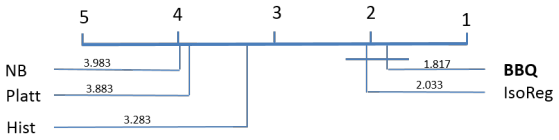
The results on real datasets are shown in the Figures 6, 7, 8, 9, and 10. In these graphs, we indicate the average rank of each method (1 is best) and we connect the methods that are statistically equivalent with our target method ***BBQ*** using a horizontal bar (e.g in Figure 8(a) the average rank of BBQ is 2.156, it is performing statistically equivalent to IsoReg ; however, its performance in



(a) RMSE results with LR as classifier



(b) RMSE results with SVM as classifier



(c) RMSE results with NB as classifier

Figure 8: Performance of each method in terms of average rank of RMSE on the real datasets. All the methods which are not connected to BBQ by the horizontal bar are significantly different from BBQ (using the F_F test followed by Holm's step-down procedure at a 0.05 significance level).

terms of RMSE is statistically superior to Hist, Platt's method, and the base classifier LR). Figure 6 shows the result of comparing the AUC of BBQ with other methods. As shown, BBQ performs significantly better than histogram binning in terms of AUC at a confidence level of $\alpha = 0.05$. Also, its performance in terms of AUC is always statistically equivalent to the base classifier (LR, SVM, NB) and isotonic regression. Note that we did not include Platt's method in our statistical test for AUC, since the AUC of the Platt's method would be the same as the AUC of the base classifier; this pattern occurs because Platt's method always uses a monotonic mapping of the base classifier output as the calibrated scores.

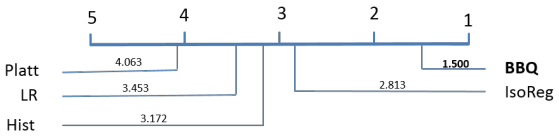
Figure 7 shows the result of comparing ACC of the BBQ with the other methods. As shown, the performance of BBQ is statistically equivalent to the rest of the calibration methods as well as the base classifier in our experiments over 30 real datasets. Figure 8 shows the results of our experiments on comparing the performance of BBQ with other calibration methods in terms of RMSE. As it shows, BBQ always outperforms the base classifier, histogram binning, and Platt's method. However, its performance is statistically equivalent to isotonic regression, whether the base classifier is LR, SVM, or NB.

Figures 9 and 10 show the results of comparing BBQ performance with the others in terms of ECE and MCE, respectively. They show that BBQ performs statistically better than all other calibration methods and the base classifier, in terms of ECE and MCE.

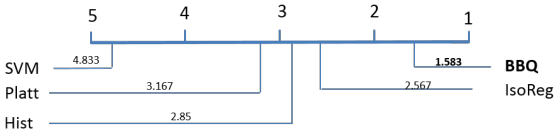
Overall, in terms of discrimination measured by AUC and ACC, the results show that the BBQ either outperforms the other calibration methods or has a performance that is not statistically significantly different from the other methods and the base classifier. In terms of calibration performance, BBQ is statistically superior to all other methods measured by ECE and MCE. Furthermore, the results show that BBQ and isotonic regression are not statistically significantly different in terms of RMSE; however, it is still statistically superior to other calibration methods and the base classifier in terms of RMSE.

4.2.3 ABB vs. BBQ

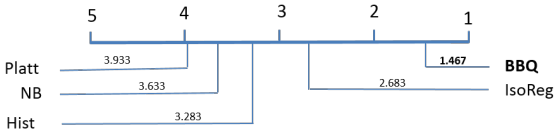
This section reports the results of a set of experiments to evaluate the performance of the full Bayesian model averaging calibration method (i.e., ABB) in comparison to the selective Bayesian



(a) ECE results with LR as classifier

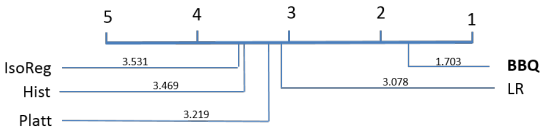


(b) ECE results with SVM as classifier

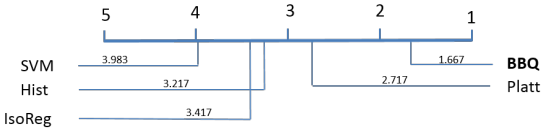


(c) ECE results with NB as classifier

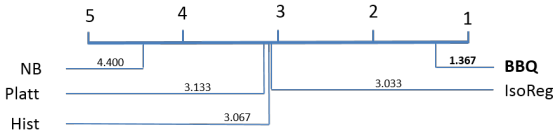
Figure 9: Performance of each method in terms of average rank of ECE on the real datasets. BBQ is statistically superior to all the compared methods (using the F_F test followed by Holm's step-down procedure at a 0.05 significance level).



(a) MCE results with LR as classifier



(b) MCE results with SVM as classifier



(c) MCE results with NB as classifier

Figure 10: Performance of each method in terms of average rank of MCE on the real datasets. BBQ is statistically superior to all the compared methods (using the F_F test followed by Holm's step-down procedure at a 0.05 significance level).

model averaging calibration method (i.e., BBQ). We subsampled real datasets to derive evaluation datasets of a relatively small size (less than 1500) in order to make it feasible to run ABB on them[5]. If these methods perform similarly, then it is worth choosing BBQ over ABB, as BBQ's running time is faster by orders of magnitude compared to ABB. If the performance of ABB is better, then it is worth using it on smaller datasets to which it can be feasibly applied.

In our experiments, we used 5-fold cross validation, and we averaged the results on the 5-folds in applying statistical significance tests. Also, we compared the performance of ABB and BBQ in two cases by using either the K2 Bayesian scoring function or the BDeu scoring function.

Table 8 indicates the results of the comparisons when we use logistic regression (LR) as the base classifier. The results show that ABB is statistically superior to BBQ in terms of accuracy (ACC) in both cases of using K2 or BDeu as the scoring function. When we use BDeu as a scoring function, ABB performs statistically significantly better than BBQ in terms of ECE. Also, when K2 is used as the scoring function, ABB is statistically significantly superior to BBQ in terms of MCE. In all other cases, according to the experimental datasets that are used, there is no statistically significant difference between the performance of ABB and BBQ, based on the Wilcoxon signed rank test at the 5% significance level.

Table 9 shows the results of experiments when we use a (linear kernel) support vector machine (SVM) as the base classifier. The results indicate that except in one case (i.e., for ECE measure when the BDeu score function is used in ABB and BBQ), there is no statistically significant difference between the performance BBQ and ABB with respect to all evaluation measures, relative to the experimental datasets that are used.

Finally, Table 10 shows the results of experiments when we use naïve Bayes (NB) as the base classifier. The results indicate that, except in one case (i.e., for evaluating accuracy when K2 is used as scoring function), in all other cases, there is no statistically significant difference between BBQ and ABB, according to the experimental datasets that are used.

[5]The name of datasets were as follows: 'SPECT', 'breast', 'mamography', 'australian', 'breast-cancer', 'colon-cancer', 'diabetes', 'duke', 'fourclass', 'german.numer', 'heart', 'ionosphere_scale', 'leu', 'liver-disorders', 'sonar_scale', 'svmguide3', 'Balance', 'solar'.

Table 8: The results of experiments in comparing ABB versus BBQ when we use LR as the base classifier. Bold face indicates the results that are significantly better based on the Wilcoxon signed rank test at the 5% significance level.

Measure	K2		BDeu	
	BBQ	ABB	BBQ	ABB
auc	0.841	0.840	0.835	0.840
accuracy	0.787	**0.808**	0.787	**0.808**
brier_score	0.156	0.147	0.151	0.146
ece	0.126	0.112	0.130	**0.111**
mce	0.343	**0.314**	0.306	0.314

4.2.4 K2 vs. BDeu

This section presents the results of experiments that evaluate the performance of the two Bayesian scoring functions that are used in our Bayesian binning methods. We are interested to know if they perform differently. Tables 11, 12, and 13 indicate the results of experiments when LR, SVM, and NB are used as the base classifier, respectively. The results show that in the case of using BBQ, there was no statistically significant difference between the performance of calibration models by changing the scoring function, except in one case (i.e., for the MCE measure when we use LR as the base classifier). Similarly, when we used the ABB model for calibration, there were only two cases (i.e., for MCE and ECE measure when SVM is used as the base classifier) when BDeu performs statistically superior to K2. Overall, the results of our experiments, which are on the particular experimental datasets that were used, show that the final performance is less sensitive on the choice of the scoring function than on the choice of running full Bayesian model averaging versus running selective Bayesian model averaging.

Table 9: The results of experiments on comparing ABB versus BBQ when we use SVM as the base classifier. Bold face indicates the results that are significantly better based on the Wilcoxon signed rank test at the 5% significance level.

Measure	K2		BDeu	
	BBQ	ABB	BBQ	ABB
auc	0.823	0.821	0.822	0.820
accuracy	0.775	0.798	0.787	0.798
brier_score	0.169	0.154	0.160	0.154
ece	0.134	0.115	0.129	**0.112**
mce	0.341	0.325	0.319	0.321

4.3 EXPERIMENTAL RESULTS ON ENIR

This section describes the set of experiments that we performed to evaluate the performance of $ENIR$ described in section 3.3 in comparison to the Isotonic Regression-based calibration (IsoReg) method (Zadrozny and Elkan, 2002), and our other proposed calibration method BBQ (Pakdaman Naeini et al., 2015a) described in section 3.2.2. We use IsoReg because it is one of the most commonly used calibration methods, which has showed good performance in real world applications (Niculescu-Mizil and Caruana, 2005b; Zadrozny and Elkan, 2002). Moreover $ENIR$ is an extension of IsoReg, and we are interested in evaluating whether it performs better than IsoReg. We also include BBQ as a state of the art binary classifier calibration model, which is a Bayesian extension of the simple histogram binning model (Pakdaman Naeini et al., 2015a). We did not include Platt's method since it is a simple and restricted parametric model and there is prior work showing that IsoReg and BBQ perform superior to Platt's method (Niculescu-Mizil and Caruana, 2005b; Zadrozny and Elkan, 2002; Pakdaman Naeini et al., 2015a). We also did not include the ACP method since it requires not only probabilistic predictions, but also a statistical confidence interval (CI) around each of those predictions, which makes it tailored to specific classifiers, such

Table 10: The results of experiments on comparing ABB versus BBQ when we use NB as the base classifier. Bold face indicates the results that are significantly better based on the Wilcoxon signed rank test at the 5% significance level.

Measure	K2		BDeu	
	BBQ	ABB	BBQ	ABB
auc	0.845	0.838	0.845	0.839
accuracy	0.785	**0.828**	0.784	0.825
brier_score	0.144	0.126	0.144	0.127
ece	0.138	0.103	0.136	0.103
mce	0.292	0.281	0.298	0.284

as LR (Jiang et al., 2012); this is counter to our goal of developing post-processing methods that can be used with any existing classification models. Finally, we did not include ABB in our experiments mainly because it is not computationally tractable for real datasets that have more than a couple of thousands instances. Moreover, even for small size datasets, we noticed that ABB performs similarly to BBQ. To evaluate the performance of the methods, we ran experiments on both simulated and on real data.

4.3.1 Simulated Data

For the simulated data experiments, we used two binary classification datasets that were used in previous work and for which the outcomes are not linearly separable (Pakdaman Naeini et al., 2015a,b). The scatter plots of these datasets are shown in Figures 5(b) and 5(a). In our experiments the data are divided into 1000 instances for training and calibrating the prediction model, and 1000 instances for testing the models. We report the average of $10-$fold cross validation results for the simulated datasets. To conduct the experiments with the simulated data, we used *support vector machines* (SVM) classifier with linear and non-linear kernels to solve the classification problem shown in Figure 5.

Table 11: The results of experiments comparing the performance of the K2 versus BDeu scoring functions when we use LR as the base classifier. Bold face indicates the results that are statistically significantly better based on the Wilcoxon signed rank test at the 5% significance level.

Measure	BBQ		ABB	
	K2	BDeu	K2	BDeu
auc	0.841	0.835	0.840	0.840
accuracy	0.787	0.787	0.808	0.808
brier_score	0.156	0.151	0.147	0.146
ece	0.126	0.130	0.112	0.111
mce	0.343	**0.306**	0.314	0.314

As seen in Tables 14 and 15, ENIR generally outperforms IsoReg on the simulation datasets, especially when the linear SVM method is used as the base learner. This is due to the monotonicity assumption of IsoReg which presumes the best calibrated estimates will match the ordering imposed by the base classifier. When we use SVM with a linear kernel, this assumption is violated due to the non-linearity of the data. Consequently, IsoReg only provides limited improvement of the calibration and discrimination performance of the base classifier. $ENIR$ performs very well in these cases since it is using the ranking information of the base classifier, but it is not anchored to it. The violation of the monotonicity assumption can happen in real data as well, especially in large scale data mining problems in which we use simple classification models due to the computational constraints. As shown in Tables 14(b) and 15(b), even if we apply a highly appropriate SVM classifier for our linearly non-separable datasets, for which IsoReg is expected to perform well (and indeed does so), $ENIR$ performs as well as IsoReg.

As can be seen in Tables 14 and 15, SVM with quadratic and RBF kernels performs very well, as measured by AUC and ACC, in terms of discriminating the classes in the simulation datasets; however, the quality of predicted probabilities are poor in terms of calibration, as measured by MCE, ECE, and RMSE. Moreover, all three calibration methods improve calibration of

Table 12: The results of experiments comparing the performance of the K2 versus BDeu scoring functions when we use SVM as the base classifier. Bold face indicates the results that are statistically significantly better based on the Wilcoxon signed rank test at the 5% significance level.

Measure	BBQ		ABB	
	K2	BDeu	K2	BDeu
auc	0.823	0.822	0.821	0.820
accuracy	0.775	0.787	0.798	0.798
brier_score	0.169	0.160	0.154	0.154
ece	0.134	0.129	0.115	**0.112**
mce	0.341	0.319	0.325	**0.321**

the base classifiers, without loosing any discrimination performance that was obtained prior to post-processing.

4.3.2 Real Data

We used 40 different baseline real dataset s from the UCI and LibSVM repositories[6] (Bache and Lichman, 2013; Chang and Lin, 2011). Five summary statistics of the size of the datasets and the percentage of the minority class are shown in Table 16.

We used three common classifiers, Logistic Regression (LR), Support Vector Machines (SVM), and Naïve Bayes (NB) to evaluate the performance ENIR. In the experiments we used the average over 10 random runs of 10-fold cross validation, and we always used the train data for calibrating the models. To compare the performance of the calibration models, we used the statistical test procedure recommended by Demsar (Demšar, 2006). More specifically, we used the Friedman non-parametric hypothesis testing method (Friedman, 1937) followed by Holm's step-down pro-

[6]The datasets used were as follows: spect, adult, breast, pageblocks, pendigits, ad, mamography, satimage, australian, code rna, colon cancer, covtype, letter unbalanced, letter balanced, diabetes, duke, fourclass, german numer, gisette scale, heart, ijcnn1, ionosphere scale, liver disorders, mushrooms, sonar scale, splice, svmguide1, svmguide3, coil2000, balance, breast cancer, leu, w1a, thyroid sick, scene, uscrime, solar, car34, car4 , protein homology.

Table 13: The results of experiments comparing the performance of the K2 versus BDeu scoring functions when we use NB as the base classifier. Bold face indicates the results that are statistically significantly better based on the Wilcoxon signed rank test at the 5% significance level.

Measure	BBQ		ABB	
	K2	BDeu	K2	BDeu
auc	0.845	0.845	0.838	0.839
accuracy	0.785	0.784	0.828	0.825
brier_score	0.144	0.144	0.126	0.127
ece	0.138	0.136	0.103	0.103
mce	0.292	0.298	0.281	0.284

cedure (Holm, 1979) to evaluate the performance of ENIR in comparison with IsoReg and BBQ, across the 40 baseline datasets.

Tables [17,18,19] show the results of the performance of $ENIR$ in comparison with IsoReg and BBQ. In these tables, we show the average rank of each method across the baseline datasets, where boldface indicates the best performing method. In these tables, the marker $*/\circledast$ indicates whether ENIR is statistically superior/inferior to the compared method using the F_F test followed by Holm's step-down procedure at a 0.05 significance level. For instance, Table 18 shows the performance of the calibration models when we use SVM as the base classifier; the results show that ENIR achieves the best performance in terms of RMSE by having an average rank of 1.675 across the 40 baseline datasets. The result indicates that in terms of RMSE, ENIR is statistically superior to BBQ; however, it is not performing statistically differently than IsoReg.

Table 17 shows the results of comparison when we use LR as the base classifier. As shown, the performance of ENIR is always superior to BBQ and IsoReg except for MCE in which BBQ is superior to ENIR; however, the difference is not statistically significant for MCE. The results show that in terms of discrimination based on AUC, there is not a statistically significant difference between the performance of ENIR compared with BBQ and IsoReg. However, ENIR performs

Table 14: Experimental Results on a simulated dataset: Circular configuration

(a) SVM Linear Kernel

	SVM	IsoReg	BBQ	ENIR
AUC	0.52	0.65	0.85	0.85
ACC	0.64	0.64	0.78	0.79
RMSE	0.52	0.46	0.39	0.38
ECE	0.28	0.35	0.05	0.05
MCE	0.78	0.60	0.13	0.12

(b) SVM Quadratic Kernel

	SVM	IsoReg	BBQ	ENIR
AUC	1.00	1.00	1.00	1.00
ACC	0.99	0.99	0.99	0.99
RMSE	0.21	0.09	0.10	0.09
ECE	0.14	0.01	0.01	0.00
MCE	0.36	0.04	0.05	0.03

Table 15: Experimental Results on Simulated dataset: XOR configuration

(a) SVM Linear Kernel

	SVM	IsoReg	BBQ	ENIR
AUC	0.50	0.68	0.90	0.89
ACC	0.64	0.67	0.83	0.83
RMSE	0.51	0.45	0.34	0.34
ECE	0.32	0.31	0.03	0.04
MCE	0.66	0.62	0.08	0.11

(b) SVM RBF Kernel

	SVM	IsoReg	BBQ	ENIR
AUC	0.99	0.99	0.99	0.99
ACC	0.96	0.96	0.96	0.96
RMSE	0.30	0.18	0.18	0.18
ECE	0.24	0.01	0.02	0.01
MCE	0.30	0.08	0.10	0.06

Table 16: Summary statistics of the size of the real datasets and the percentage of the minority class. Q1 and Q3 defines the first quartile and thirds quartile respectively.

	Min	Q1	Median	Q3	Max
Size of data	42	683	1861	8973	581,012
Percentage of minority class	0.009	0.076	0.340	0.443	0.500

statistically better than BBQ in terms of ACC. In terms of calibration measures, ENIR is statistically superior to both IsoReg and BBQ in terms of RMSE. In terms of MCE, ENIR is statistically superior to IsoReg.

Table 18 shows the results when we use SVM as the base classifier. As shown, the performance of ENIR is always superior to BBQ and IsoReg except for MCE in which BBQ performs better than ENIR; however, the difference is not statistically significant for MCE. The results show that although ENIR is superior to IsoReg and BBQ in terms of discrimination measures, AUC and ACC, the difference is not statistically significant. In terms of calibration measures, ENIR performs statistically superior to BBQ in terms of RMSE and it is statistically superior to IsoReg in terms of MCE.

Table 19 shows the results of comparison when we use NB as the base classifier. As shown, the performance of ENIR is always superior to BBQ and IsoReg. In terms of discrimination, for AUC there is not a statistically significant difference between the performance of ENIR compared with BBQ and IsoReg; however, in terms of ACC, ENIR is statistically superior to BBQ. In terms of calibration measures, ENIR is always statistically superior to IsoReg. ENIR is also statistically superior to BBQ in terms of ECE and RMSE.

Overall, in terms of discrimination measured by AUC and ACC, the results show that ENIR either outperforms IsoReg and BBQ or it has a performance that is not statistically significantly different. In terms of calibration measured by ECE, MCE, and RMSE, ENIR either outperforms the other calibration methods, or it has a statistically equivalent performance to IsoReg and BBQ.

In addition to comparing the performance of ENIR with IsoReg and BBQ, we also show in Table 22 the 95% confidence interval for the mean of the random variable X, which is defined as

Table 17: Average rank of the calibration methods on the real datasets using LR as the base classifier. Marker $*/\circledast$ indicates whether ENIR is statistically superior/inferior to the compared method (using the F_F test followed by Holm's step-down procedure at a 0.05 significance level).

	IsoReg	BBQ	ENIR
AUC	1.963	2.225	**1.813**
ACC	1.675	2.663$*$	**1.663**
RMSE	1.925$*$	2.625$*$	**1.450**
ECE	2.125	1.975	**1.900**
MCE	2.475$*$	**1.750**	1.775

Table 18: Average rank of the calibration methods on the real datasets using SVM as the base classifier. Marker $*/\circledast$ indicates whether ENIR is statistically superior/inferior to the compared method (using the F_F test followed by Holm's step-down procedure at a 0.05 significance level).

	IsoReg	BBQ	ENIR
AUC	1.988	2.025	**1.988**
ACC	2.000	2.150	**1.850**
RMSE	1.850	2.475$*$	**1.675**
ECE	2.075	2.025	**1.900**
MCE	2.550$*$	**1.625**	1.825

Table 19: Average rank of the calibration methods on the real datasets using NB as the base classifier. Marker $*/\circledast$ indicates whether ENIR is statistically superior/inferior to the compared method (using the F_F test followed by Holm's step-down procedure at a 0.05 significance level).

	IsoReg	BBQ	ENIR
AUC	2.150	**1.925**	**1.925**
ACC	1.963	2.375$*$	**1.663**
RMSE	2.200$*$	2.375$*$	**1.425**
ECE	2.475$*$	2.075$*$	**1.450**
MCE	2.563$*$	1.850	**1.588**

the percentage of the gain (or loss) of ENIR with respect to the base classifier:

$$X = \frac{measure_{enir} - measure_{method}}{measure_{method}}, \tag{4.1}$$

where $measure$ is one of the evaluation measures AUC, ACC, ECE, MCE, or RMSE. Also, $method$ denotes one of the choices of the base classifiers, namely, LR, SVM, or NB. For instance, Table 22 shows that by post-processing the output of SVM using ENIR, we are 95% confident to gain anywhere from 17.6% to 31% average improvement in terms of RMSE. This could be a promising result, depending on the application, considering the 95% CI for the AUC which shows that by using ENIR we are 95% confident not to loose more than 1% of the SVM discrimination power in terms of AUC (Note, however, that the CI includes zero, which indicates that there is not a statistically significant difference between the performance of SVM and ENIR in terms of AUC).

Overall, the results in Table 22 show that there is not a statistically meaningful difference between the performance of ENIR and the base classifiers in terms of AUC. The results support at a 95% confidence level that ENIR improves the performance of LR or NB in terms of ACC. Furthermore, the results in Table 22 show that by post-processing the output of LR, SVM, and NB using ENIR, we can make dramatic improvements in terms of calibration measured by $RMSE$, ECE, and MCE. For instance, the results indicate that at a 95% confidence level, ENIR improved the average performance of NB in terms of ECE anywhere from 30.5% to 55.2%, which could be practically significant in many decision-making and data mining applications.

Finally, Table 21 shows a summary of the time complexity of different binary classifier calibration methods in learning for *N* training instances and the test time for only one instance.

4.4 EXPERIMENTAL RESULTS ON ELITE

This section describes the set of experiments that we performed to evaluate the performance of the ELiTE calibration method in comparison to other commonly used calibration methods. The comparison methods include histogram binning (Zadrozny and Elkan, 2001b), Platt's method (Platt, 1999), isotonic regression (Zadrozny and Elkan, 2002), and BBQ, which is a Bayesian extension to the histogram binning method (Pakdaman Naeini et al., 2015a). We did not include ABB in our

Table 20: The 95% confidence interval for the average percentage of improvement over the base classifiers(LR, SVM, NB) by using the ENIR method for post-processing. Positive entries for AUC and ACC mean ENIR is on average performing better discrimination than the base classifiers Negative entries for RMSE, ECE, and MCE mean that ENIR is on average performing better calibration than the base classifiers.

	LR	SVM	NB
AUC	[-0.008 , 0.003]	[-0.010 , 0.003]	[-0.010 , 0.000]
ACC	[0.002 , 0.016]	[-0.001 , 0.010]	[0.012 , 0.068]
RMSE	[-0.124 , -0.016]	[-0.310 , -0.176]	[-0.196 , -0.100]
ECE	[-0.389 , -0.153]	[-0.768 , -0.591]	[-0.514 , -0.274]
MCE	[-0.313 , -0.064]	[-0.591 , -0.340]	[-0.552 , -0.305]

Table 21: Note that *N* and *B* are the size of training sets and the number of bins found by the method respectively. *T* is the number of iterations required for convergence of Platt's method and *M* is defined as the total number of models used in the associated ensemble model.

	Training Time	Testing Time
Platt	$O(NT)$	$O(1)$
Hist	$O(N \log N)$	$O(\log B)$
IsoReg	$O(N \log N)$	$O(\log B)$
ACP	$O(N \log N)$	$O(N)$
ABB	$O(N^2)$	$O(N^2)$
BBQ	$O(N \log N)$	$O(M \log N)$
ENIR	$O(N \log N)$	$O(M \log B)$

experiments mainly because it is not computationally tractable for datasets that have more than couple of thousands of instances.

Similar to our previous set of experiments, we used three common classifiers, Logistic Regression (LR), Support Vector Machines (SVM), and Naïve Bayes (NB) to evaluate the performance of ELiTE. In the experiments, we used the average over 10 random runs of 10-fold cross validation, and we always used the training data for calibrating the models.

We ran two sets of experiments on 35 binary outcome classification datasets from the UCI and LibSVM repositories[7] (Bache and Lichman, 2013; Chang and Lin, 2011). In the first set of experiments we were interested in evaluating if there is experimental support for using ELiTE as a post-processing calibration method. Table 22 shows the 95% confidence interval for the mean of the random variable X, which is defined as the percentage of the gain (or loss) of ELiTE with respect to the base classifier:

$$X = \frac{measure_{elite} - measure_{method}}{measure_{method}}, \tag{4.2}$$

where $measure$ is one of the evaluation measures AUC, ACC, ECE, MCE, or RMSE. Also, $method$ denotes one of the choices of the base classifiers, namely, LR, SVM, or NB. For instance, Table 22 shows that by post-processing the output of SVM using ELiTE, we are 95% confident to gain anywhere from 16% to 30% average improvement in terms of RMSE. This could be a promising result, depending on the application, considering the 95% CI for the AUC which shows that by using ELiTE we are 95% confident not to lose more than 1% of the SVM discrimination power in terms of AUC (Note also that the CI includes zero, which indicates that there is not a statistically significant difference between the performance of SVM and ELiTE in terms of AUC).

Overall, the results in Table 22 show that there is not a statistically meaningful difference between the performance of ELiTE and the base classifiers in terms of AUC. The results support at a 95% confidence level that ELiTE improves the performance of the LR and NB base classifiers in terms of ACC. Furthermore, the results in Table 22 show that by post-processing the output of LR, SVM, and NB using ELiTE, we can make dramatic improvements in terms of calibration measured

[7]The datasets used were as follows: spect, adult, breast, pageblocks, pendigits, ad, australian, colon cancer, letter unbalanced, letter balanced, diabetes, duke, fourclass, german numer, gisette scale, heart, ionosphere scale, liver disorders, mushrooms, sonar scale, splice, svmguide1, svmguide3, coil2000, balance, breast cancer, w1a, thyroid sick, scene, uscrime, solar, car34, car4, mamography, satimage.

by RMSE, ECE, and MCE. For instance, the results indicate that at a 95% confidence level, ELiTE improved the average performance of NB in terms of ECE anywhere from 27% to 55%, which could be practically significant in many decision-making and data mining applications.

Table 22: The 95% confidence interval for the average percentage of improvement over the base classifiers (LR, SVM, NB) by using the ELiTE method for post-processing. Positive entries for AUC and ACC mean ELiTE is on average performing better discrimination than the base classifiers. Negative entries for RMSE, ECE, and MCE mean that ELiTE is on average performing better calibration than the base classifiers.

	LR	SVM	NB
AUC	[-0.01 , 0.01]	[-0.01 , 0.01]	[-0.01 , 0.01]
ACC	[0.00 , 0.02]	[0.00 , 0.01]	[0.02 , 0.08]
RMSE	[-0.14 , -0.02]	[-0.30 , -0.16]	[-0.22 , -0.11]
ECE	[-0.40 , -0.18]	[-0.76 , -0.56]	[-0.55 , -0.27]
MCE	[-0.35 , -0.12]	[-0.58 , -0.33]	[-0.62 , -0.39]

In the second set of experiments on real data, we are interested in evaluating the performance of ELiTE compared with the base classifier and other calibration methods. To evaluate the performance of models, we used the recommended statistical test procedure by Janez Demsar (Demšar, 2006). More specifically, we used the non-parametric testing method based on the F_F test statistics (Iman and Davenport, 1980), which is an improved version of Friedman non-parametric hypothesis testing method (Friedman, 1937), followed by Holm's step-down procedure (Holm, 1979) to evaluate the performance of ELiTE in comparison with other methods, across the 35 baseline datasets.

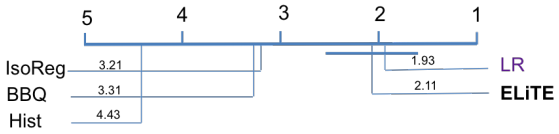
The results on real datasets are shown in the Figures 6-10. In these graphs, we indicate the average rank of each method (1 is best) and we connect the methods that are statistically equivalent with our target method ELiTE using a horizontal bar (e.g., in Figure 8(a) the average rank of ELiTE is 1.89, and it is performing statistically equivalent to isoreg in terms of RMSE; however, its performance in terms of RMSE is statistically superior to Hist, Platt's method, BBQ, and the base classifier LR). Figure 6 shows the results of comparing the AUC of ELiTE with other methods. As shown, ELiTE performs significantly better than all other calibration methods in terms of AUC at a confidence level of $\alpha = 0.05$. Also, its performance in terms of AUC is always statistically

equivalent to the base classifier (LR, SVM, NB). Note that we did not include Platt's method in our statistical test for AUC, since the AUC of the Platt's method would be the same as the AUC of the base classifier; this pattern occurs because Platt's method always uses a monotonic mapping of the base classifier's output as the calibrated score.

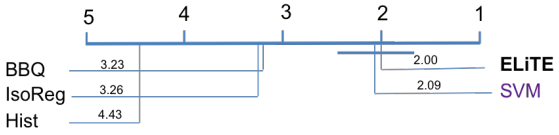
Figure 7 shows the results of comparing ACC of ELiTE with the other methods. As shown, ELiTE performs statistically better than histogram binning and Platt's method, as well as the base classifiers NB, and LR. However, ELiTE is statistically equivalent to BBQ and IsoReg, as well as the base classifier SVM, in our experiments over 35 real datasets. Figure 8 shows the results of our experiments in comparing the performance of ELiTE with other calibration methods in terms of RMSE. ELiTE always outperforms the base classifier and all other calibration methods. However, its difference with isotonic regression is not statistically significant, when the base classifier is LR or NB.

Figures 9, and 10 show the results of comparing ELiTE performance with the others in terms of ECE and MCE, respectively. They show that ELiTE performs superior to all other calibration methods and to the base classifier, in terms of ECE and MCE. However, its difference with BBQ is not statistically significant in terms of ECE when the base classifier is SVM or NB. Also, in terms of MCE, the difference between ELiTE and BBQ is not statistically significant when SVM is used as the base classifier.

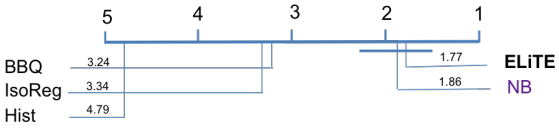
Overall, in terms of discrimination measured by AUC and ACC, the results show that the proposed non-parametric calibration method either outperforms the other calibration methods or has a performance that is not statistically significantly different from the other methods and the base classifier. In terms of calibration performance, ELiTE is often statistically superior to the other methods and is never statistically significantly worse.



(a) AUC Results on LR

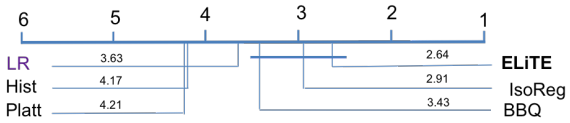


(b) AUC results on SVM

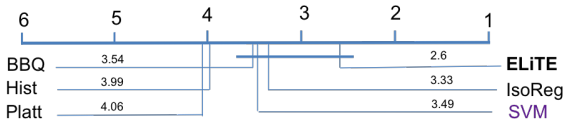


(c) AUC results on NB

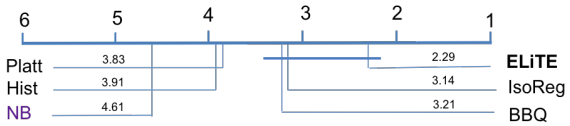
Figure 11: Performance of each method in terms of average rank of AUC on the real datasets. All the methods which are not connected to ELiTE by the horizontal bar are statistically significantly worse than ELiTE (using an improved Friedman test followed by Holm's step-down procedure at a 0.05 significance level).



(a) ACC Results on LR

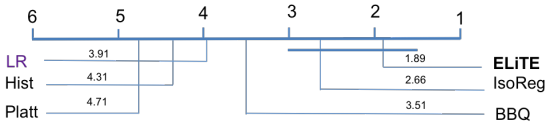


(b) ACC results on SVM

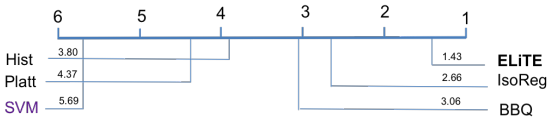


(c) ACC results on NB

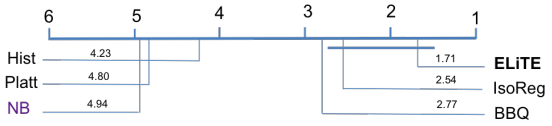
Figure 12: Performance of each method in terms of average rank of ACC on the real datasets. All the methods which are not connected to ELiTE by the horizontal bar are statistically significantly worse than ELiTE (using an improved Friedman test at a 0.05 significance level).



(a) RMSE Results on LR

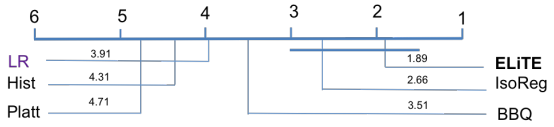


(b) RMSE results on SVM

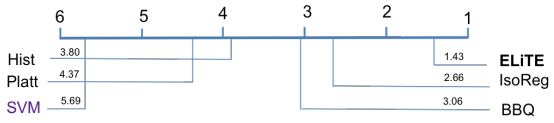


(c) RMSE results on NB

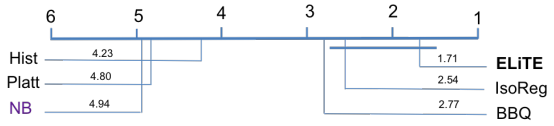
Figure 13: Performance of each method in terms of average rank of RMSE on the real datasets. All the methods which are not connected to ELiTE by the horizontal bar are statistically significantly worse than ELiTE (using an improved Friedman test followed by Holm's step-down procedure at a 0.05 significance level).



(a) ECE Results on LR

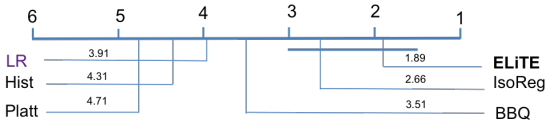


(b) ECE results on SVM

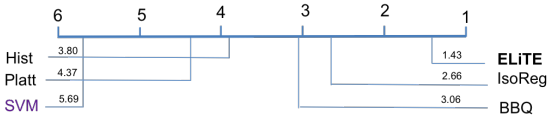


(c) ECE results on NB

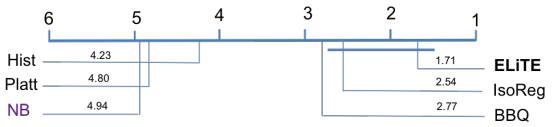
Figure 14: Performance of each method in terms of average rank of ECE on the real datasets. All the methods which are not connected to ELiTE by the horizontal bar are statistically significantly worse than ELiTE (using an improved Friedman test followed by Holm's step-down procedure at a 0.05 significance level).



(a) MCE Results on LR



(b) MCE results on SVM



(c) MCE results on NB

Figure 15: Performance of each method in terms of average rank of MCE on the real datasets. ELiTE is almost always statistically superior to all other competing methods (using the improved Friedman test followed by Holm's step-down procedure at a 0.05 significance level).

4.5 EMPIRICAL EVALUATION OF KDE, DPM, ENIR, AND ELITE

This section reports the results of a set of experiments on real datasets to evaluate the performance of ELiTE, as our final calibration method, in comparison to several other leading calibration methods introduced in this dissertation. In particular, EliTE is compared to calibration methods based on kernel density estimation (KDE) and Dirichlet Process Mixtures (DPM) introduced in Section 3.1, and the ensemble of near-isotonic regression (ENIR) method introduced in Section 3.3. We did not include BBQ in our experiments because we have already shown that ELiTE and ENIR perform superior to BBQ in Section 4.4 and Section 4.3.

Similar to our previous experiments, we consider three commonly used binary classifiers: logistic regression (LR), support vector machine (SVM), and naïve Bayes(NB). In the experiments, we used 35 binary outcome classification datasets from the UCI and LibSVM repositories [8] (Bache and Lichman, 2013; Chang and Lin, 2011). To compare the performance of the calibration methods, we used the Friedman non-parametric hypothesis testing method (Friedman, 1937) followed by Holm's step-down procedure (Holm, 1979) to evaluate the performance of ELiTE in comparison with ENIR, KDE, and DPM, across the 35 baseline datasets.

Tables 23, 24, and 25 indicate the results of our experiments when we use LR, SVM, and NB as the base classifier, respectively. In these tables, the bold face indicates the method that is performing the best in terms of average rank over the 35 baseline datasets that are used in our experiments. Also, in these tables, the marker $*/\circledast$ indicates whether ELiTE is statistically superior/inferior to the compared method (using the F_F test followed by Holm's step-down procedure at a 0.05 significance level). In addition, the bottom row of each table shows the overall running time of each method in minutes over the 10 runs of the 10-fold cross validation experiments.

As indicated in Tables 23, 24, and 25, in terms of evaluation measures, ELiTE performed well both in terms of calibration and discrimination measures in comparison to the other competing calibration methods. In particular, on the 35 real datasets used in our experiments, ELiTE commonly performs statistically significantly better than the other methods, and never worse. However, in

[8] The datasets used were as follows: spect, adult, breast, pageblocks, pendigits, ad, australian, colon cancer, letter unbalanced, letter balanced, diabetes, duke, fourclass, german numer, gisette scale, heart, ionosphere scale, liver disorders, mushrooms, sonar scale, splice, svmguide1, svmguide3, coil2000, balance, breast cancer, w1a, thyroid sick, scene, uscrime, solar, car34, car4, mamography, satimage.

terms of running time, ELiTE runs slower in comparison to the competing methods even though its running time complexity is still $O(NlogN)$ (Ramdas and Tibshirani, 2014).

Table 23: Average rank of the calibration methods on the 35 real datasets using LR as the base classifier. Marker $*/\circledast$ indicates whether ELiTE is statistically superior/inferior to the compared method (using the F_F test followed by Holm's step-down procedure at a 0.05 significance level). The bottom row of the table shows the overall Running Time (RT) of each method in minutes over the 10 runs of the 10-fold cross validation experiments, using a single core of a MacBook Pro with a 2.5 GHz Intel Core i7 CPU and a 16 GB RAM memory.

	KDE	DPM	ENIR	ELiTE
AUC	3.10$*$	2.53$*$	2.76$*$	**1.61**
ACC	2.69	2.87$*$	2.44	**2.00**
RMSE	2.51$*$	2.94$*$	2.66$*$	**1.89**
ECE	2.49	**2.29**	2.86	2.37
MCE	2.29	2.71$*$	3.00$*$	**2.00**
RT	262	205	226	1808

4.6 EFFECT OF CALIBRATION SIZE

This section presents the results of our experiments on the effect of calibration size on the performance of different calibration methods. In these experiments, we used our simulated dataset with a circular configuration. The scatter plot of the simulated dataset is shown in Figure 5(b). We used a SVM classifier with linear and quadratic kernels in this set of experiments. Due to non-linearity of data, the SVM with a linear kernel is expected to perform poorly in terms of calibration and discrimination. In contrast, the SVM with a quadratic kernel is a perfect choice for discriminating the patterns according to the circular separation boundary between the classes.

We run two set of experiments in this section based on the choice of the training and the calibration instances. In the first configuration, we used 5000 randomly generated data to learn the

Table 24: Average rank of the calibration methods on the 35 real datasets using SVM as the base classifier. Marker $*/\circledast$ indicates whether ELiTE is statistically superior/inferior to the compared method (using the F_F test followed by Holm's step-down procedure at a 0.05 significance level). The bottom row of the table shows the overall Running Time (RT) of each method in minutes over the 10 runs of the 10-fold cross validation experiments, using a single core of a MacBook Pro with a 2.5 GHz Intel Core i7 CPU and a 16 GB RAM memory.

	KDE	DPM	ENIR	ELiTE
AUC	2.90$*$	2.07	2.97$*$	**2.06**
ACC	2.43	2.70	2.66	**2.21**
RMSE	2.37	2.77	2.69	**2.17**
ECE	**1.86**	2.66	3.06	2.43
MCE	**1.97**	2.83	3.06	2.14
RT	290	102	225	2082

SVM base classifiers. The calibration data were also generated randomly and were separate from the 5000 cases used to train the base classifier. We changed the number of instances that are used for calibrating the models and set them to be 10, 20, 50, 100, 500, 1000, 2000, and 5000. In the second configuration, the training data and the calibration data are the same, and similar to the first configuration we set the size data to be 10, 20, 50, 100, 500, 1000, 2000, and 5000. Note that in both configurations, for each evaluation measure, we reported the average of 30 random samples on a separate test dataset with 10000 samples.

Figure 16 shows the performance of different calibration methods on various sizes of the calibration dataset, when we use a SVM with a linear kernel. There are a few important observations here. First, the results show that the Platt's method and isotonic regression have limited performance in improving calibration and discrimination of the predictions even in the presence of a large calibration dataset. Second, as we expected, the binning based models such as BBQ and ENIR, require a fair amount of calibration instances (i.e., at least around 500) to reach close to

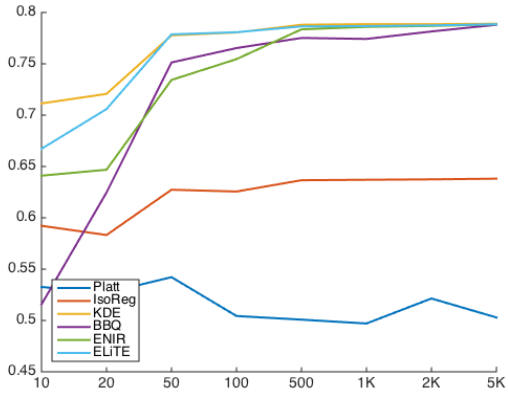
Table 25: Average rank of the calibration methods on the 35 real datasets using NB as the base classifier. Marker $*/\circledast$ indicates whether ELiTE is statistically superior/inferior to the compared method (using the F_F test followed by Holm's step-down procedure at a 0.05 significance level). The bottom row of the table shows the overall Running Time (RT) of each method in minutes over the 10 runs of the 10-fold cross validation experiments, using a single core of a MacBook Pro with a 2.5 GHz Intel Core i7 CPU and a 16 GB RAM memory.

	KDE	DPM	ENIR	ELiTE
AUC	2.71$*$	2.77$*$	3.00$*$	**1.51**
ACC	2.53	2.89	2.43	**2.16**
RMSE	2.51	3.09$*$	2.49	**1.91**
ECE	2.60	2.46	2.77	**2.17**
MCE	2.49	2.66$*$	2.94$*$	**1.91**
RT	286	188	235	2056

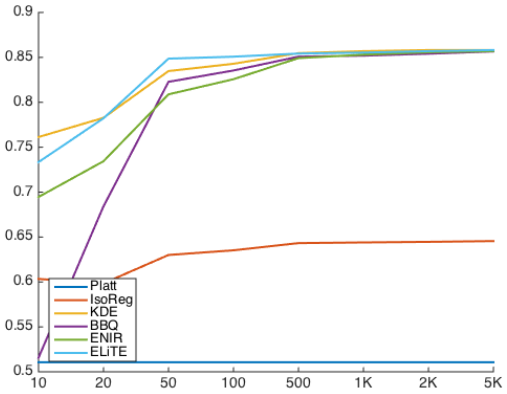
their best performance. Finally, the results show that ELiTE and KDE based calibration perform very well over all ranges of calibration data sizes.

Figure 17 shows the performance of different calibration methods on various sizes of calibration data when we use a SVM with a quadratic kernel. As we expected, isotonic regression performs very well in this case since the monotonicity assumption is valid (i.e., the AUC of the base classifier is close to 1). We note that the performance of Platt's method improves across discrimination and calibration measures by increasing the calibration size. However, the rate of improvement is still inferior compared to the rate of improvement for other methods. We note that ELiTE still performs among the best models for calibration, especially when we have a fair amount data for calibration (i.e., calibration size is around 500).

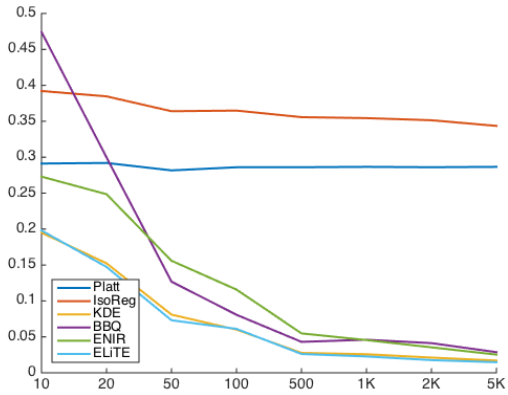
Figure 18, and 19 show the performance of different calibration methods on various sizes of the (calibration/training) dataset, when we use a SVM classifier with a linear kernel and quadratic kernel, respectively. We notice a similar pattern of observations in these graphs compared to the



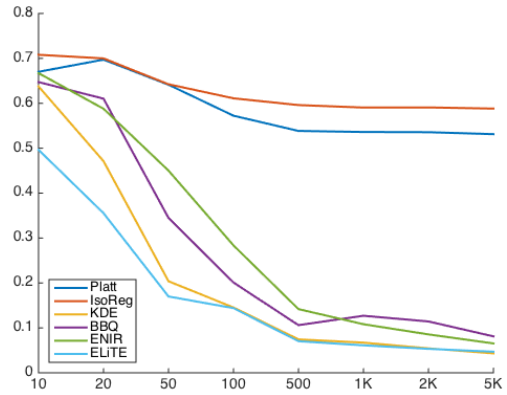
(a) Accuracy



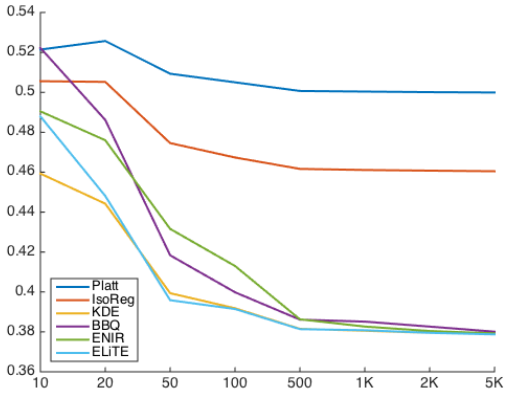
(b) AUC



(c) ECE

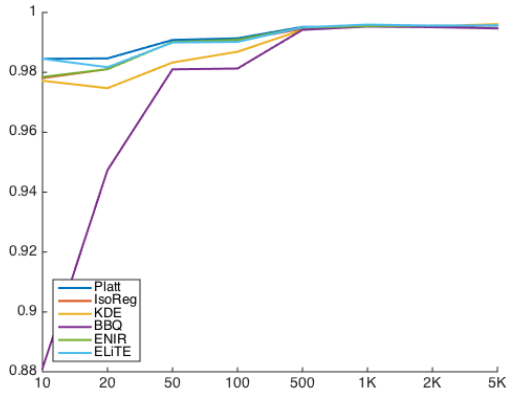


(d) MCE

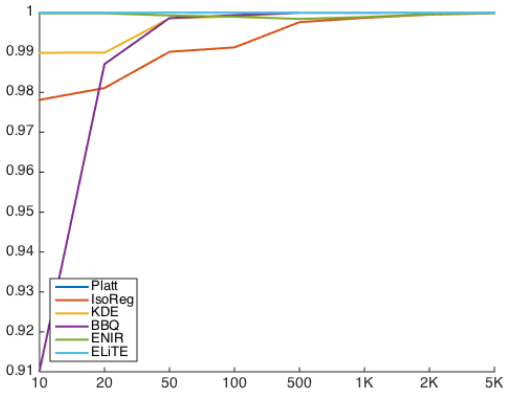


(e) RMSE

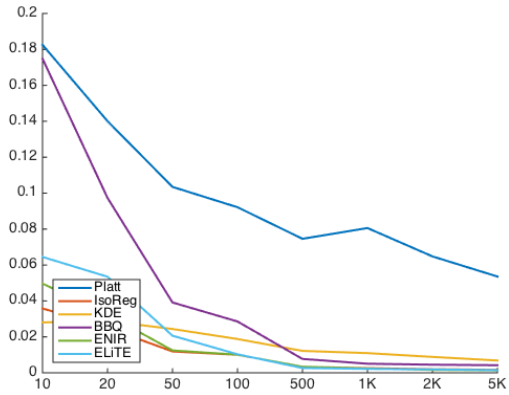
Figure 16: The effect of calibration size when using simulated data and a linear kernel SVM as the base classifier. The x axis shows the number of calibration instances that are used in learning the calibration models.



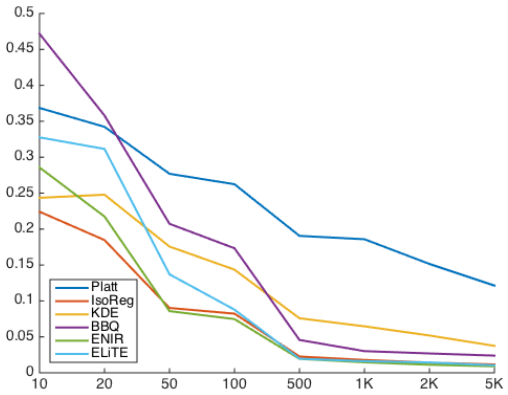
(a) Accuracy



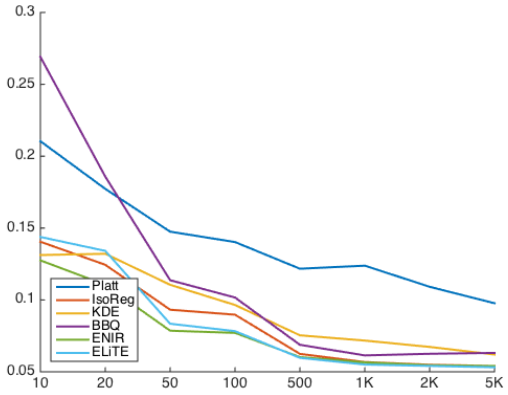
(b) AUC



(c) ECE

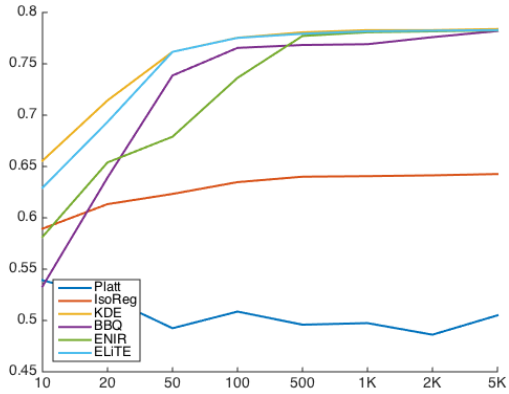


(d) MCE

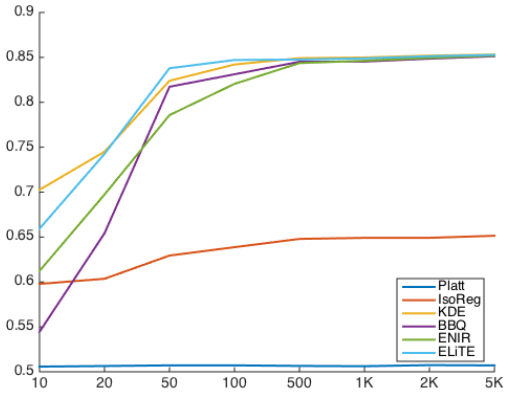


(e) RMSE

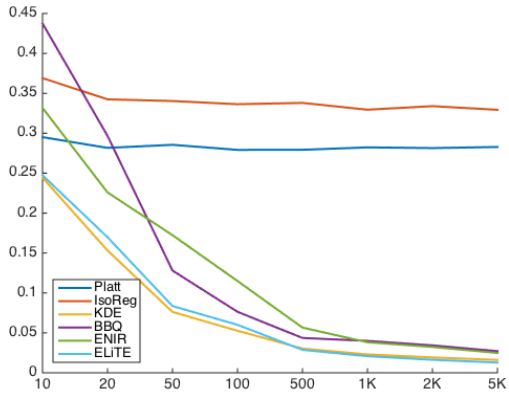
Figure 17: The effect of calibration size when using simulated data and a quadratic kernel SVM as the base classifier. The x axis shows the number of calibration instances that are used in learning the calibration models.



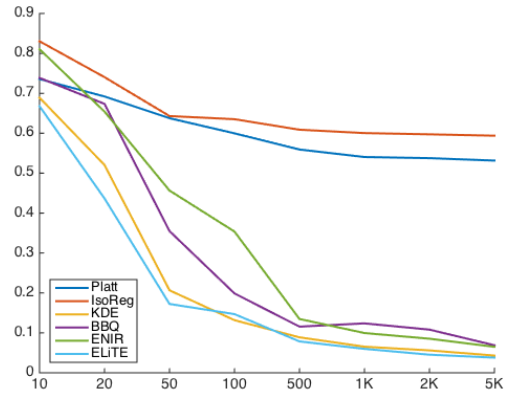
(a) Accuracy



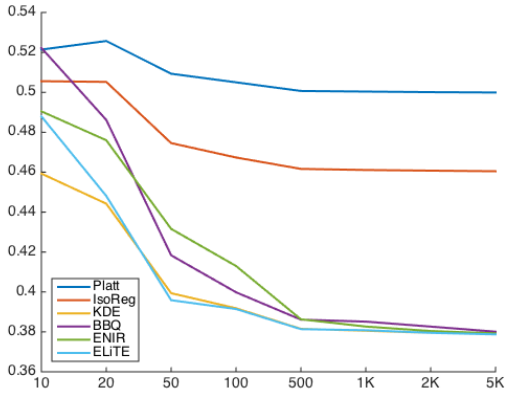
(b) AUC



(c) ECE



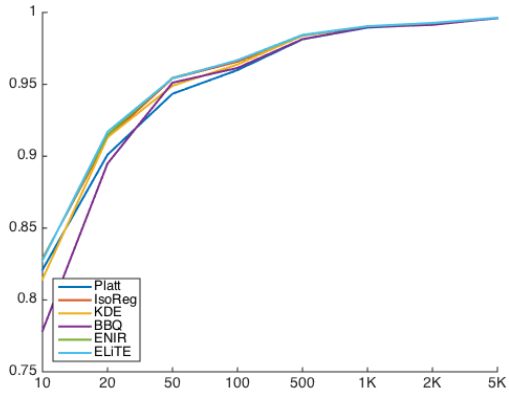
(d) MCE



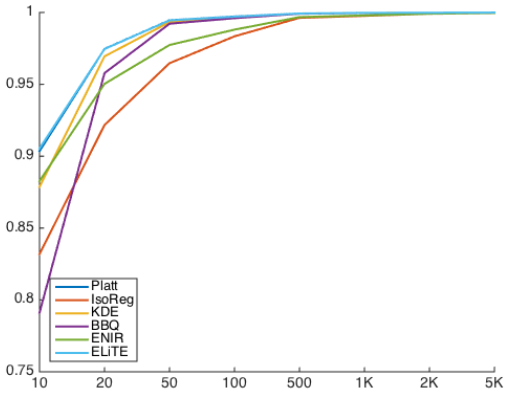
(e) RMSE

Figure 18: The effect of calibration size when using simulated data and a linear kernel SVM as the base classifier. The x axis shows the number of (calibration/training) instances that are used in learning the calibration models. In these set of experiments, we used the same set of data for learning the parameters of the classification model and the calibration model.

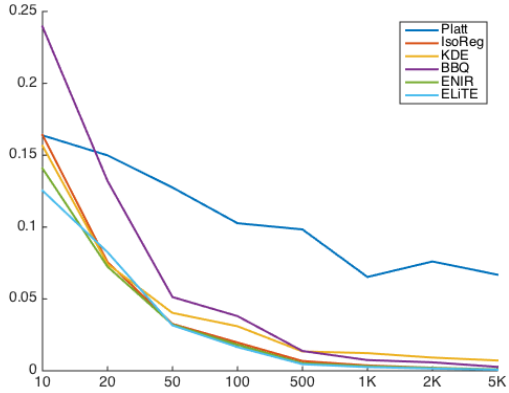
case that we used separate training and calibration datasets. However, there are some differences. First, comparing the results of linear kernel SVM in Figures 16 and 18, the results indicate that when we use a small sample size for building the calibration model (e.g., 10 or 20), the performance of calibration models is boosted by using separate data for training and calibration. However, note that according to these graphs when we use a large number of calibration instances (e.g., 2000 or 5000), the performance of calibration models in the two configurations is not different. We note similar observations comparing Figures 17 and 19 for the case that we use a quadratic kernel SVM as the base classifier. In addition, in the case of using a quadratic kernel SVM, we note that Platt's method has superior performance, in particular in terms of AUC and MCE, when we use a small (training/calibration) sample size in our second configuration in which the training and calibration data are the same. Overall, with a few exceptions on small sample sizes, the results support that using the same versus separate calibration and training datasets produces about the same results over different dataset sizes.



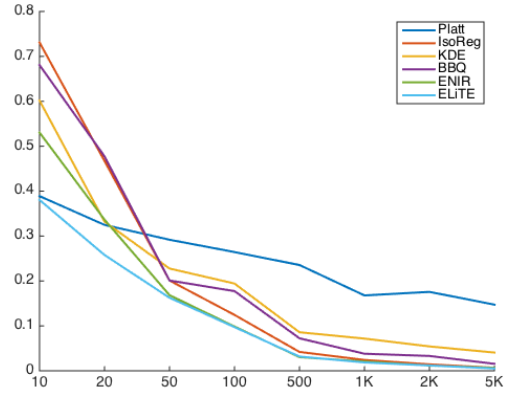
(a) Accuracy



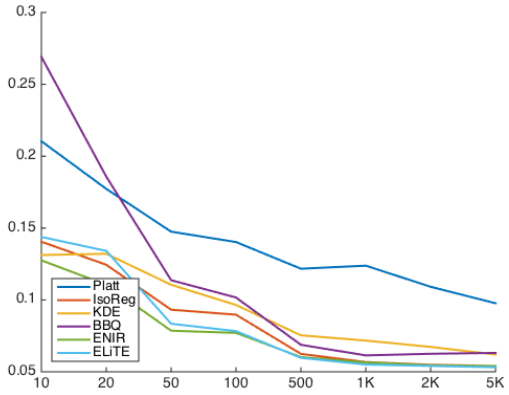
(b) AUC



(c) ECE



(d) MCE



(e) RMSE

Figure 19: The effect of calibration size when using simulated data and a quadratic kernel SVM as the base classifier. The x axis shows the number of (calibration/training) instances that are used in learning the calibration models. In these set of experiments, we used the same set of data for learning the parameters of the classification model and the calibration model.

5.0 THEORETICAL FINDINGS ON BINARY CLASSIFIER CALIBRATION

In this chapter we describe our theoretical findings to date for histogram binning models stating that in the presence of large number of samples it is possible to make a classifier perfectly calibrated without sacrificing the discrimination performance in terms of AUC. Before going into the details of the theorems and the proofs, we present some notation we will use in the proofs.

5.1 NOTATION AND ASSUMPTIONS:

Assume a binary classifier is defined as a mapping $\phi : R^d \rightarrow [0, 1]$. As a result, for every input instance $x \in R^d$ the output of the classifier is $y = \phi(x)$, where $y \in [0, 1]$. For calibrating the classifier $\phi(.)$, we assume there is a training set $\{(x_i, y_i, z_i)\}_{i=1}^{N}$ where $x_i \in R^d$ is the i'th instance and $y_i = \phi(x_i) \in [0, 1]$, and $z_i \in \{0, 1\}$ is the true class of i'th instance. Also we define $\hat{y}_i$ as the probability estimate for instance x_i achieved by using the histogram binning calibration method, which is intended to yield a more calibrated estimate than y_i. In addition, we have the following notation and assumptions that are used in the remainder of this section:

- N is total number of instances
- m is total number of positive instances
- n is total number of negative instances
- p_{in} is the space of uncalibrated probabilities $\{y_i\}$ which is defined by the classifier output
- p_{out} is the space of transformed probability estimates $\{\hat{y}_i\}$ using histogram binning
- B is the total number of bins defined on p_{in} in the histogram binning model
- B_i is the i'th bin defined on p_{in}

- N_i is total number of instances x_k for which the predicted value y_k is located inside B_i
- m_i is number of positive instances x_k for which the predicted value y_k is located inside B_i
- n_i is number of negative instances x_k for which the predicted value y_k is located inside B_i
- $\hat{\eta}_i = \frac{N_i}{N}$ is an empirical estimate of $\mathbb{P}\{y \in B_i\}$
- η_i is the value of $\mathbb{P}\{y \in B_i\}$ as N goes to infinity
- $\hat{\theta}_i = \frac{m_i}{N_i}$ is an empirical estimate of $\mathbb{P}\{z = 1 | y \in B_i\}$
- θ_i is the value of $\hat{\theta}_i$ as N goes to infinity

5.2 CALIBRATION THEOREMS

In this section we study the properties of the histogram-binning calibration method. We prove three theorems that show that this method can improve the calibration power of a classifier without sacrificing its discrimination capability.

5.2.1 Convergence Results on MCE

This section describes a new theorem and its proof that shows the MCE of the histogram binning method is concentrated around zero as $N \to \infty$:

Theorem 5.2.1. *Using histogram binning calibration, with probability at least* $1 - \delta$ *we have* $MCE \leq \sqrt{\frac{2B \log \frac{2B}{\delta}}{N}}$.

Proof. For proving this theorem, we first use a concentration result for $\hat{\theta}_i$. Using Hoeffding's inequality we have the following:

$$\mathbb{P}\{|\hat{\theta}_i - \theta| \geq \epsilon\} \leq 2e^{\frac{-2N\epsilon^2}{B}} \tag{5.1}$$

Let's assume $\tilde{B}_i$ is a bin defined on the space of transformed probabilities p_{out} for calculating the MCE of histogram binning method. Assume after using histogram binning over p_{in} (space of uncalibrated probabilities which is generated by the classifier ϕ), $\hat{\theta}_{i1}, .., \hat{\theta}_{ik_i}$ will be mapped into $\tilde{B}_i$. We define o_i as the true fraction of positive instances in bin $\tilde{B}_i$, and e_i as the mean of the

post-calibrated probabilities for the instances in bin $\tilde{B}_i$. Using the notation defined in section 5.1, we can write o_i and e_i as follows:

$$o_i = \frac{\eta_{i1}\theta_{i1} + \ldots + \eta_{ik_i}\theta_{ik_i}}{\eta_{i1} + \ldots + \eta_{ik_i}},\ e_i = \frac{\eta_{i1}\hat{\theta}_{i1} + \ldots + \eta_{ik_i}\hat{\theta}_{ik_i}}{\eta_{i1} + \ldots + \eta_{ik_i}}$$

By defining $\alpha_{it} = \frac{\eta_{it}}{\eta_{i1}+\ldots+\eta_{ik_i}}$ and using the triangular inequality we have that:

$$|o_i - e_i| \leq \sum_{t\in\{1,\ldots,k_i\}} \alpha_{it}|\hat{\theta}_{it} - \theta_{it}| \ \leq \ max_{t\in\{1,\ldots,k_i\}}|\hat{\theta}_{it} - \theta_{it}| \tag{5.2}$$

Using the above result and the concentration inequality 5.1 for $\hat{\theta}_i$ we can conclude:

$$\mathbb{P}\{|o_i - e_i| > \epsilon\} \leq \mathbb{P}\{max_{t\in\{1,\ldots,k_i\}}|\hat{\theta}_{it} - \theta_{it}| > \epsilon\} \leq 2k_i e^{\frac{-N\epsilon^2}{2B}}, \tag{5.3}$$

Where the last inequality is obtained by using a union bound and k_i is the number of bins on the space p_{in} for which their calibrated probability estimate will be mapped into the bin $\tilde{B}_i$.

Using a union bound again over different bins like $\tilde{B}_i$ defined on the space p_{out}, we achieve the following probability bound for MCE over the space of calibrated estimates p_{out} :

$$\mathbb{P}\{\max_{i=1}^{B}|o_i - e_i| \geq \epsilon\} \leq 2(k_1 + \ldots + k_B)e^{\frac{-N\epsilon^2}{2B}} \implies \mathbb{P}\{MCE \geq \epsilon\} \leq 2Be^{\frac{-N\epsilon^2}{2B}}$$

By setting $\delta = 2Be^{\frac{-N\epsilon^2}{2B}}$ we can show that with probability at least $1-\delta$ the following inequality holds $MCE \leq \sqrt{\frac{2B\log\frac{2B}{\delta}}{N}}$. □

Corollary 5.2.1.1. *Using histogram binning calibration method, MCE converges to zero at a rate of* $O(\sqrt{\frac{B\log B}{N}})$.

5.2.2 Convergence Results on ECE

This section describes a new theorem and its proof that shows the ECE of the histogram binning method is also concentrated around zero as $N \to \infty$:

Theorem 5.2.2. *Using the histogram binning calibration method, ECE converges to zero with the rate of* $O(\sqrt{\frac{B}{N}})$.

Proof. Here we show that using histogram binning calibration method, ECE converges to zero with the rate of $O(\sqrt{\frac{B}{N}})$. Let's define E_i as the expected calibration loss on bin $\tilde{B}_i$ for the histogram binning method. Following the assumptions mentioned in Section 3 about MCE bound theorem, we have $E_i = E(|e_i - o_i|)$. Also, using the definition of ECE and the notations in Section 2, we can rewrite ECE as the convex combination of E_is. As a result, in order to bound ECE it suffices to show that all of its E_i components are bounded. Using the concentration results proved in MCE bound theorem, we have:

$$\mathbb{P}\{|o_i - e_i| > \epsilon\} \leq 2k_i e^{\frac{-N\epsilon^2}{2B}}, \tag{5.4}$$

also let's recall the following two identities:

Lemma 5.2.3. *If X is a positive random variable then* $E[X] = \int_0^\infty \mathbb{P}(X > t)dt$

Lemma 5.2.4. $\int_0^\infty e^{-x^2}dx = \frac{\sqrt{\pi}}{2}$

Now, using the concentration result in Equation 5.4 and applying the two above identities we can bound E_i to write $E_i \leq C\sqrt{\frac{B}{N}}$, where C is a constant. Finally, since ECE is the convex combination of E_i's we can conclude that using histogram binning method, ECE converges to zero with the rate of $O(\sqrt{\frac{B}{N}})$. □

5.2.3 Convergence Results on AUC Loss

The above two theorems show that we can bound the calibration error of a binary classifier, which is measured in terms of MCE and ECE, by using a histogram-binning post-processing method. In this section we show that in addition to gaining calibration power, by using histogram binning we are guaranteed not to sacrifice discrimination performance of the base classifier $\phi(.)$ measured in terms of AUC. Recall the definitions of y_i and $\hat{y}_i$, where $y_i = \phi(x_i)$ is the probability prediction

of the base classifier $\phi(.)$ for the input instance x_i, and $\hat{y}_i$ is the transformed estimate for instance x_i that is achieved by using the histogram-binning calibration method.

We can define the AUC_Loss of the histogram-binning calibration method as:

Definition 5.2.1. *(AUC_Loss) AUC_Loss* is the difference between the AUC of the base classifier estimate and the AUC of transformed estimate using the histogram-binning calibration method. Using the notation in Section 5.1, it is defined as $AUC\_Loss = AUC(y) - AUC(\hat{y})$

Using the above definition, our third theorem bounds the AUC_Loss of histogram binning classifier as follows:

Theorem 5.2.5. *Using the histogram-binning calibration method, the worst case AUC_Loss is upper bounded by $O(\frac{1}{B})$ as $N \to \infty$*

Proof. For proving the theorem, let's first recall the concentration results for $\hat{\eta}_i$ and $\hat{\theta}_i$. Using Hoeffding's inequality we have the following:

$$\mathbb{P}\{|\hat{\theta}_i - \theta| \geq \epsilon\} \leq 2e^{\frac{-2N\epsilon^2}{B}} \tag{5.5}$$

$$\mathbb{P}\{|\hat{\eta}_i - \eta| \geq \epsilon\} \leq 2e^{-2N\epsilon^2} \tag{5.6}$$

The above concentration inequalities show that with probability $1 - \delta$ we have the following inequalities:

$$|\hat{\theta}_i - \theta_i| \leq \sqrt{\frac{B}{2N}\log(\frac{2}{\delta})} \tag{5.7}$$

$$|\hat{\eta}_i - \eta_i| \leq \sqrt{\frac{1}{2N}\log(\frac{2}{\delta})} \tag{5.8}$$

The above results show that for the large amount of data with high probability, $\hat{\eta}_i$ is concentrated around η_i and $\hat{\theta}_i$ is concentrated around around θ_i.

Based on Agarwal et al. (2006), the empirical AUC of a classifier $\phi(.)$ is defined as follows:

$$\hat{AUC} = \frac{1}{mn}\sum_{i:z_i=1}\sum_{j:z_j=0} I(y_i > y_j) + \frac{1}{2}I(y_i = y_j) \tag{5.9}$$

Where m and n as mentioned in section 5.1 (notation and assumptions) are respectively the total number of positive and negative examples. Computing the expectation of Equation 5.9 gives the AUC as follows:

$$AUC = Pr\{y_i > y_j | z_i = 1, z_j = 0\} + \frac{1}{2} Pr\{y_i = y_j | z_i = 1, z_j = 0\} \tag{5.10}$$

Using the MacDiarmid concentration inequality it is also possible to show that the empirical $\hat{AUC}$ is highly concentrated around true AUC (Agarwal et al., 2006), although we do not so here.

Recall p_{in} is the space of outputs of base classifier (ϕ). Also, p_{out} is the space of outputs of the transformed probability estimates from histogram binning. Assume $B_1, \ldots, B_B$ are the non-overlapping bins defined on the p_{in} in the histogram binning approach. Also, assume y_i and y_j are the base classifier outputs for two different instance where $z_i = 1$ and $z_j = 0$. In addition, assume $\hat{y}_i$ and $\hat{y}_j$ are respectively, the transformed probability estimates for y_i and y_j using histogram binning method.

Now using the above assumptions we can write the AUC loss of using histogram binning method as follows:

$$\begin{aligned} AUC\_Loss &= AUC(y) - AUC(\hat{y}) \\ &= \mathbb{P}\{y_i > y_j | z_i = 1, z_j = 0\} + \frac{1}{2}\mathbb{P}\{y_i = y_j | z_i = 1, z_j = 0\} \\ &- (\mathbb{P}\{\hat{y}_i > \hat{y}_j | z_i = 1, z_j = 0\} + \frac{1}{2}\mathbb{P}\{\hat{y}_i = \hat{y}_j | z_i = 1, z_j = 0\}) \end{aligned} \tag{5.11}$$

By partitioning the space of uncalibrated estimates p_{in} one can write the AUC_Loss as follows:

$$\begin{aligned} &AUC\_Loss = \\ &\sum_{K,L} (\mathbb{P}\{y_i > y_j, y_i \in B_K, y_j \in B_L | z_i = 1, z_j = 0\} - \mathbb{P}\{\hat{y}_i > \hat{y}_j, y_i \in B_K, y_j \in B_L | z_i = 1, z_j = 0\}) \\ &+ \sum_{K} (\mathbb{P}\{y_i > y_j, y_i \in B_K, y_j \in B_K | z_i = 1, z_j = 0\} \\ &+ \frac{1}{2}\mathbb{P}\{y_i = y_j, y_i \in B_K, y_j \in B_K | z_i = 1, z_j = 0\} \\ &- \frac{1}{2}\mathbb{P}\{\hat{y}_i = \hat{y}_j, y_i \in B_K, y_j \in B_K | z_i = 1, z_j = 0\}), \end{aligned} \tag{5.12}$$

where B_K and B_L are two bins defined in the histogram binning method. Also, we make the following reasonable assumption that simplifies our calculations:

- Assumption 1 : $\hat{\theta}_i \neq \hat{\theta}_j$ if $i \neq j$

Now we will show that the first summation part in equation 5.12 will be less than or equal to zero. Also, the second summation part will go to zero with the convergence rate of $O(\frac{1}{B})$ as $N \to \infty$.

5.2.3.1 First Summation Part Recall that in the histogram binning method the calibration estimate $\hat{y} = \hat{\theta}_K$ if $y \in B_K$. Also, notice that if $y_i \in B_K$, $y_j \in B_L$ and $K > L$ then we have $y_i > y_j$ for sure. So, using the above facts we can rewrite the first summation part in equation 5.12 as follows:

$$\begin{aligned} Loss_1 = &\sum_{K>L} \mathbb{P}\{y_i \in B_K, y_j \in B_L | z_i = 1, z_j = 0\} \\ &- \sum_{K,L} \mathbb{P}\{\hat{\theta}_K > \hat{\theta}_L, y_i \in B_K, y_j \in B_L | z_i = 1, z_j = 0\} \end{aligned} \tag{5.13}$$

We can rewrite the above equation as following:

$$\begin{aligned} Loss_1 = \sum_{K>L} (&\mathbb{P}\{y_i \in B_K, y_j \in B_L | z_i = 1, z_j = 0\} \\ &- \mathbb{P}\{\hat{\theta}_K > \hat{\theta}_L, y_i \in B_K, y_j \in B_L | z_i = 1, z_j = 0\} \\ &- \mathbb{P}\{\hat{\theta}_L > \hat{\theta}_K, y_i \in B_L, y_j \in B_K | z_i = 1, z_j = 0\}) \end{aligned} \tag{5.14}$$

Next by using Bayes' rule and omitting the common denominators among the terms we have the following:

$$
\begin{aligned}
Loss_1 \propto \sum_{K>L} \Big(& \mathbb{P}\{z_i = 1, z_j = 0 | y_i \in B_K, y_j \in B_L\} \\
& - \mathbb{P}\{\hat{\theta}_K > \hat{\theta}_L, z_i = 1, z_j = 0 | y_i \in B_K, y_j \in B_L\} \\
& - \mathbb{P}\{\hat{\theta}_L > \hat{\theta}_K, z_i = 1, z_j = 0 | y_i \in B_L, y_j \in B_K\} \Big) \times \mathbb{P}\{y_i \in B_L, y_j \in B_K\}
\end{aligned}
\tag{5.15}
$$

We next show that the term inside the parentheses in equation 5.15 is less or equal to zero by using the i.i.d. assumption and the notation we mentioned in Section 5.1, as follows:

$$
\begin{aligned}
Inside\_Term(IT) = (& \theta_K(1-\theta_L) \\
& - \mathbb{I}\{\hat{\theta}_K > \hat{\theta}_L\}\theta_K(1-\theta_L) \\
& - \mathbb{I}\{\hat{\theta}_L > \hat{\theta}_K\}\theta_L(1-\theta_K))
\end{aligned}
\tag{5.16}
$$

Now if we have the case $\hat{\theta}_K > \hat{\theta}_L$ then IT term would be exactly zero. If we have the case that $\hat{\theta}_L > \hat{\theta}_K$ then the inside term would be equal to:

$$
\begin{aligned}
IT &= \theta_K(1-\theta_L) - \theta_L(1-\theta_K) \\
&\simeq \hat{\theta}_K(1-\hat{\theta}_L) - \hat{\theta}_L(1-\hat{\theta}_K) \qquad \text{as } N \to \infty \\
&\leq 0
\end{aligned}
\tag{5.17}
$$

where the last inequality is true with high probability which comes from the concentration results for $\hat{\theta}_i$ and θ_i in equation 5.5.

5.2.3.2 Second Summation Part Using the fact that in the second summation part $\hat{y_i} = \hat{\theta}_K$ and $\hat{y_j} = \hat{\theta}_K$, we can rewrite the second summation part as:

$$
\begin{aligned}
&Loss_2 = \\
&\sum_K ((\mathbb{P}\{y_i > y_j, y_i \in B_K, y_j \in B_K | z_i = 1, z_j = 0\} \\
&+ \frac{1}{2}\mathbb{P}\{y_i = y_j, y_i \in B_K, y_j \in B_K | z_i = 1, z_j = 0\}) \\
&- (\frac{1}{2}\mathbb{P}\{y_i \in B_K, y_j \in B_K | z_i = 1, z_j = 0\})) \\
&\leq \sum_K (\mathbb{P}\{y_i \in B_K, y_j \in B_K | z_i = 1, z_j = 0\} - \frac{1}{2}\mathbb{P}\{y_i \in B_K, y_j \in B_K | z_i = 1, z_j = 0\}) \\
&= \frac{1}{2}\sum_K \mathbb{P}\{y_i \in B_K, y_j \in B_K | z_i = 1, z_j = 0\})
\end{aligned}
\tag{5.18}
$$

Using the Bayes rule and iid assumption of data we can rewrite the equation 5.18 as following:

$$
\begin{aligned}
Loss_2 &\leq \frac{1}{2}\frac{\sum_K \mathbb{P}\{z_i = 1, z_j = 0 | y_i \in B_K, y_j \in B_K\} \times \mathbb{P}\{y_i \in B_K, y_j \in B_K\}}{\mathbb{P}\{z_i = 1, z_j = 0\}} \\
&= \frac{1}{2}\frac{\sum_K \mathbb{P}\{z_i = 1, z_j = 0 | y_i \in B_K, y_j \in B_K\} \times \mathbb{P}\{y_i \in B_K\}\mathbb{P}\{y_j \in B_K\}}{\sum_{K,L} \mathbb{P}\{z_i = 1, z_j = 0 | y_i \in B_K, y_j \in B_L\} \times \mathbb{P}\{y_i \in B_K\}\mathbb{P}\{y_j \in B_L\}} \\
&= \frac{1}{2}\frac{\sum_K \mathbb{P}\{z_i = 1, z_j = 0 | y_i \in B_K, y_j \in B_K\} \times \eta_K^2}{\sum_{K,L} \mathbb{P}\{z_i = 1, z_j = 0 | y_i \in B_K, y_j \in B_L\} \times \eta_K \eta_L} \\
&= \frac{1}{2}\frac{\sum_K \mathbb{P}\{z_i = 1, z_j = 0 | y_i \in B_K, y_j \in B_K\}}{\sum_{K,L} \mathbb{P}\{z_i = 1, z_j = 0 | y_i \in B_K, y_j \in B_L\}},
\end{aligned}
\tag{5.19}
$$

where the last equality comes from the fact that η_K and η_L are concentrated around their empirical estimates $\hat{\eta}_K$ and $\hat{\eta}_L$ which are equal to $\frac{1}{B}$ by construction (we build our histogram model based on equal frequency bins).

Using the i.i.d. assumptions about the calibration samples, we can rewrite the equation 5.19 as following:

$$\begin{aligned} Loss_2 &\leq \frac{\sum_K \mathbb{P}\{z_i = 1 | y_i \in B_K\}\mathbb{P}\{z_j = 0 | y_j \in B_K\}}{2\sum_K \mathbb{P}\{z_i = 1 | y_i \in B_K\} \times \sum_L \mathbb{P}\{z_j = 0 | y_j \in B_L\}} \\ &= \frac{\sum_{k=1}^{B} \theta_k(1-\theta_k)}{2\sum_{k=1}^{B} \theta_k \times \sum_{l=1}^{B}(1-\theta_l)} \\ &\leq \frac{1}{2B}, \end{aligned} \tag{5.20}$$

where the last inequality comes from the fact that the order of $\{(1-\theta_1), \ldots, (1-\theta_B)\}$'s is completely reverse in comparison to the order of $\{\theta_1, \ldots, \theta_B\}$ and using Chebychev's Sum inequality as follows (Engel, 1998):

Theorem 5.2.6. (Chebyshev's sum inequality) *if* $a_1 \leq a_2 \leq \ldots \leq a_n$ *and* $b_1 \geq b_2 \geq \ldots \geq b_n$ *then* $\frac{1}{n}\sum_{k=1}^{n} a_k b_k \leq (\frac{1}{n}\sum_{k=1}^{n} a_k)(\frac{1}{n}\sum_{k=1}^{n} b_k)$

Now the facts we proved above about $Loss_1$ and $Loss_2$ in equations 5.20 and 5.17 show that as $N \to \infty$ the worst case AUC_Loss is upper bounded by $O(\frac{1}{B})$ using the histogram binning calibration method. □

Remark. It should be noticed, the above proof shows that the worst case AUC loss in the presence of large number of training data point is bounded by $O(\frac{1}{B})$. However, it is possible that we even gain AUC power by using the histogram binning calibration method as we showed in our experiments on simulated datasets in Chapter 4.

Using the above three Theorems 5.2.1, 5.2.2, and 5.2.5, we can conclude that, as $N \to \infty$ and $B \to \infty$, by using the histogram-binning calibration method we can improve calibration performance of a classifier measured in terms of MCE and ECE without losing discrimination performance of the base classifier measured in terms of AUC. Note that by setting $B = \theta(\sqrt[3]{N})$, which is the rate used for the number of bins in BBQ, the AUC_Loss, ECE, and MCE all converge to zero at a rate of $\theta(\frac{1}{\sqrt[3]{N}})$ as $N \to \infty$.

In addition to our theoretical findings presented in this section, there are other related theoretical results that support the setting of $B = \theta(\sqrt[3]{N})$. For instance, D. Freeman and Persi Diaconis showed that, under mild assumptions over the true density function, the mean square difference

between the estimated empirical density by an equal-size histogram method and the true density function is minimized by setting $B = \theta(\sqrt[3]{N})$ (Freedman and Diaconis, 1981). Also, it has been shown that if equal size histogram binning is used for binary classification, then the setting $B = \theta(\sqrt[3]{N})$ achieves the best convergence rate for minimizing the excess risk (i.e., the difference between the best empirical-risk-minimizer classifier and the Bayes classifier) (Scott and Nowak, 2002; Scott et al., 2006; singh, 2011; Nowak, 2009). Although these results are related to histogram binning models with equal bin-size, we conjecture that similar theoretical results could be obtained for equal frequency histogram binning models. This is an area for our future research.

5.3 EMPIRICAL EVALUATION

We also perform an empirical study on the effect of the size training dataset on the calibration performance of the histogram binning method. Table 26 shows the results of experiments on using the histogram-binning calibration method for different sizes of calibration sets on the simulated data showed in Section 5(b) using SVM classifier with linear and quadratic kernels. In these experiments the number of bins to be equal to $B = \sqrt[3]{N}$. Also, we set the size of training data to be 1000 and we fixed 10000 instances for testing the methods. For capturing the effect of calibration size, we change the size of calibration data from 10^2 up to 10^6, running the experiment 10 times for each calibration set and averaging the results. As seen in Table 26, by having more calibration data, we have a steady decrease in the values of the MCE and ECE errors.

Table 26: Experimental results on the size of calibration data using histogram-binning method on the simulated dataset 5(b).

(a) SVM Linear

	10^2	10^3	10^4	10^5	10^6	Base SVM
AUC	0.82	0.84	0.85	0.85	0.85	0.49
MCE	0.40	0.15	0.07	0.05	0.03	0.52
ECE	0.14	0.05	0.03	0.02	0.01	0.28

(b) SVM Quadratic Kernel

	10^2	10^3	10^4	10^5	10^6	Base SVM
AUC	0.99	1.00	1.00	1.00	1.00	1.00
MCE	0.14	0.09	0.03	0.01	0.01	0.36
ECE	0.03	0.01	0.00	0.00	0.00	0.15

6.0 MULTI-CLASS CLASSIFIER CALIBRATION

This chapter is focused on the multi-class classifier calibration problem. Section 6.1 introduces a new multi-class classifier calibration method. Section 6.1.1 presents the results and findings of the method compared with baseline calibration methods on a set of real datasets. Finally, Section 6.2 presents the results of using the method in a causal network discovery application.

6.1 EXTENDING PLATT'S METHOD FOR MULTI-CLASS CALIBRATION

In designing a multi-class calibration method, we made a simple extension to Platt's method which is a commonly used parametric binary classifier calibration method. As we described in detail in Section 2, Platt's method uses the sigmoid transformation to map the output of a binary classifier to a calibrated probability (Platt, 1999). It then optimizes a logistic loss function to learn the two parameters of the model. The method has two advantages: (1) it has only two parameters that make it a viable choice for low sample size calibration datasets, and (2) it runs in $O(1)$ at test time, and thus, is fast. A natural extension to Platt's method for the multi-class calibration task is to use a combination of a softmax transfer function and a cross-entropy loss function instead of a sigmoid function and a logistic loss function, respectively. Minimizing the cross entropy is equivalent to minimizing the empirical Kullback–Leibler divergence of the estimated probabilities and the observed ones. The minimum will be achieved by the true probability distribution, and minimizing the cross entropy function will result in finding the closest distribution parameterized by the model to the observed distribution of the data (Nielsen, 2015).

The model that uses the softmax transfer function and optimizes the cross entropy loss function is called softmax regression (or multinomial logistic regression) (Nielsen, 2015). The softmax

regression-based calibration model inherits the desirable properties of Platt's method. However, similar to Platt's method, the mapping that the softmax regression-based calibration method can learn is restrictive since the final separating boundaries between each pair of classes is always linear. A simple relaxation of this restriction is to use a shallow neural network with one hidden layer (SNN) (Nielsen, 2015). Figure 20 shows the architecture of the shallow neural network model that we used to post-process the uncalibrated probabilities. In our experiments, we trained 10 different

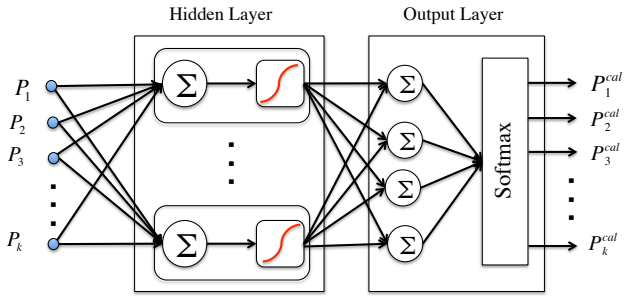


Figure 20: The structure of the multi-class post-processing calibration method. The inputs on the left are the k jointly exhaustive and mutually exclusive class probabilities generated by a multi-class classifier. The outputs on the right are the corresponding post-processed probabilities that are intended to be better calibrated.

such shallow neural networks by setting the number of neurons in the hidden layer to be 4, 5, 6, or 7 randomly. At test time, we use the average of the 10 different outputs generated by these models as the final calibrated probability estimates. The averaging is helpful since it reduces the variance error of the predictions and improves the final performance of the post-processed probabilities (Domingos, 2000). We implemented our model using the scikit flow python package[1], which uses the tensor flow machine learning package (Abadi et al., 2016). We used the cross-entropy loss function and the adagrad optimization method (Duchi et al., 2011) to learn the parameters, and we set the learning rate and the batch size (two parameters of the stochastic gradient search) (Abadi et al., 2016) to be equal to 0.1, and 10, respectively.

[1]https://github.com/tensorflow/skflow

6.1.1 Experimental Results on UCI Datasets

This section describes the set of experiments that we conducted to evaluate the performance of the SNN calibration method. We used 56 multi-class classification data sets collected from the UCI and LibSVM repositories[2] (Bache and Lichman, 2013; Chang and Lin, 2011).

We used three common classifiers, multi-class Logistic Regression (LR), multi-class Support Vector Machines (SVM) (Smola and Schölkopf, 2004), and Naïve Bayes (NB), to evaluate the performance of our proposed multi-class calibration model. We used the average over five random runs of 5-fold cross validation, and we always used the train data for calibrating the models.

To evaluate the performance of the model, we used a variety of evaluation measures including the Brier_score, cross_entropy, accuracy (ACC), and multi-class AUC (Hand and Till, 2001). We also used MCE_micro, and ECE_micro, which are explained next.

Assume a multi-class classification problem with k number of classes. Any probabilistic multi-class classification model that is designed to solve this problem outputs k exhaustive and mutually exclusive probabilities distribution vector $p_1, \ldots, p_k$. To compute MCE_micro and ECE_micro, we augmented the k-element probability distribution vectors $p_1, \ldots, p_k$ for all test instances to form an aggregated vector P_{all}. We also augmented their corresponding one-hot binary labels (Nielsen, 2015; Abadi et al., 2016) to form an aggregated binary vector Z_{all}. The ECE-micro and MCE-micro are defined as the expected calibration error and the maximum calibration error calculated based on P_{all} and Z_{all} (See Section 2.4.3 for the definition of the MCE and ECE measures)..

Similar to our experiments on binary classifier calibration models in Section 4.4, we ran two sets of experiments on the baseline multi-class outcome classification datasets. [3] In the first set of experiments, we were interested in evaluating if there was experimental support for using the SNN method as a post-processing calibration method. Table 27 shows the 95% confidence interval for

[2]The datasets used were as follows: sector, letter, dna, connect_4, glass, acoustic_scale, iris, wine, vehicle, digits, seismic, usps, svmguide4, svmguide2, vowel, segment, satimage, pendigits, shuttle, sensorless_drive, KingRook_vs_King, ecoli, contraceptive_method_choice, CNAE_9, cardiotocography, car_evaluation, breast_tissue, balance_scale, abalone, zoo_4class, yeast, white_wine_quality, red_wine_quality, waveform, wall_following_robot, vertebral_column, user_knowledge, urban_land_cover, teaching_assistant_evaluation, steel_plates_faults, HAPT, semeion, seeds, page_blocks, optdigits, plant_species, libras_movement, leaf_data, gesture_phase_segmentation, human_activity_recognition, forest_type, flag, ISOLET, firm_teacher_clave, online_news_popularity, PUC_Rio

[3]Note that, however, we filter out those datasets that the base classifier (LR, SVM, or NB) is achieving a calibration error of less than 0.05 in terms of ECE_micro. In these cases, we assume the classifier is already well-calibrated and there is no need for further calibration post processing.

the mean of the random variable X, which is defined as the percentage of the gain (or loss) of SNN with respect to the base classifier:

$$X = \frac{measure_{SNN} - measure_{method}}{measure_{method}}, \tag{6.1}$$

In the above equation, $measure$ is one of the evaluation measures described earlier in this section. Also, $method$ denotes one of the choices of the base classifiers: LR (Softmax regression), SVM, or NB. For instance, Table 27 shows that by post-processing the output of SVM using the SNN calibration method, we are 95% confident to gain anywhere from 23% to 37% average improvement in terms of Brier-score. This could be a promising result, depending on the application, considering the 95% CI for the AUC which shows that by using the proposed calibration method, we are 95% confident not to lose more than 1% of the SVM discrimination power in terms of AUC (also note that the CI includes zero, which indicates that there is not a statistically significant difference between the performance of SVM and the SNN calibration method in terms of AUC).

Overall, the results in Table 27 show that there is not a statistically meaningful difference between the performance of the SNN calibration method and the base classifiers in terms of AUC. The results support at a 95% confidence level that our post-processing method improves the performance of all of the base classifiers in terms of ACC. Furthermore, the results in Table 27 show that by post-processing the output of LR, SVM, and NB, we can make significant improvements in terms of calibration measured by ECE_micro and MCE_micro, as well as Brier_Score and Cross_Entropy.

In our second set of experiments, we used the Friedman non-parametric hypothesis testing method (Friedman, 1937) followed by Holm's step-down procedure (Holm, 1979) to evaluate the performance of the SNN calibration method in comparison with the simple baseline model introduced by Zadrozny and Elkan (Zadrozny and Elkan, 2002), across the baseline datasets. In implementing the baseline calibration method, we utilize a binary calibration method such as isotonic regression(Zadrozny and Elkan, 2002) or Platt's method (Platt, 1999) to post-process the corresponding output probabilities of each class separately. This is performed in a one-versus-remainder fashion by considering the corresponding output probability of class i as a binary classification score. We then use one of the binary classifier calibration methods (e.g., the isotonic regression or Platt's method) to calibrate the corresponding probabilities. Finally, we normalized the resulting

Table 27: The 95% confidence interval for the average percentage of improvement over the base classifiers (LR, SVM, NB) by using the SNN method for post-processing. Positive entries for AUC and ACC mean SNN is on average performing better discrimination than the base classifiers. Negative entries for Brier_score, ECE_micro, and MCE_micro mean that SNN is on average performing better calibration than the base classifiers.

	LR	SVM	NB
AUC	[-0.01, 0.02]	[-0.01, 0.03]	[0.00, 0.07]
ACC	[0.00, 0.14]	[0.01, 0.08]	[0.03, 0.24]
cross_entropy	[-0.55, -0.19]	[-0.42, -0.26]	[-0.59, -0.34]
Brier_score	[-0.51, -0.18]	[-0.37, -0.23]	[-0.32, -0.22]
ECE_micro	[-0.64, -0.27]	[-0.74, -0.57]	[-0.81, -0.65]
MCE_micro	[-0.63, -0.25]	[-0.74, -0.60]	[-0.77, -0.58]

calibrated binary class probabilities so that they added to one as described in (Zadrozny and Elkan, 2002).

Table 28: Average rank of the calibration methods on the benchmark datasets using multi-class LR as the base classifier. Marker $*/\circledast$ indicates whether SNN is statistically superior/inferior to the compared method (using an improved Friedman test followed by Holm's step-down procedure at a 0.05 significance level).

	IsoReg	Platt	Softmax	SNN
AUC	3.23$*$	2.67	2.00	2.10
ACC	2.47	2.30	2.83	2.40
CrosEntropy	3.60$*$	2.07	2.87$*$	1.47
Brier_Score	2.33	2.47	3.47$*$	1.73
ECE_micro	1.73	2.60	3.53$*$	2.13
MCE_micro	1.87	2.33	3.60$*$	2.20

Tables [28,29,30] show the results of the performance of SNN in comparison with the two baseline calibration methods using IsoReg and Platt, as well as softmax regression, which is a multi-class extension to Platt's method. In these tables, we show the average rank of each method across the baseline datasets. In these tables, the marker $*/\circledast$ indicates whether SNN is statistically

superior/inferior to the compared method using the improved F_F test followed by Holm's step-down procedure at a 0.05 significance level. For instance, Table 30 shows the performance of the calibration models when we use the Naïve Bayes as the base classifier; the results show that SNN achieves the best performance in terms of Brier_Score by having an average rank of 1.94 across the baseline datasets. The result indicates that in terms of Brier_Score, SNN is statistically superior to the baseline calibration method using Platt's method; however, it is not performing statistically differently than the softmax regression calibration and the baseline calibration method using IsoReg.

Table 29: Average rank of the calibration methods on the benchmark datasets using multi-class SVM as the base classifier. Marker $*$/$\circledast$ indicates whether SNN is statistically superior/inferior to the compared method (using an improved Friedman test followed by Holm's step-down procedure at a 0.05 significance level).

	IsoReg	Platt	Softmax	SNN
AUC	2.89	2.73	2.19	2.19
ACC	2.11	2.68	2.91	2.31
CrosEntropy	3.92$*$	1.70	2.41	1.97
Brier_Score	1.86	2.41	3.22	2.51
ECE_micro	1.81	2.38	3.35$*$	2.46
MCE_micro	1.73$\circledast$	2.38	3.30$*$	2.59

Overall, the SNN method is often either outperforming the competing methods or there is not a statistically meaningful difference between its performance compared with other methods; however, there is one case in Table 29 that the SNN method performs statistically inferior to the baseline calibration method using IsoReg for MCE_micro measure when we use SVM as the base classifier. Also, the SNN method always performs statistically significantly superior to the baseline model using IsoReg in terms of cross_entropy measure.

Table 30: Average rank of the calibration methods on the benchmark datasets using Naïve Bayes as the base classifier. Marker $*/\circledast$ indicates whether SNN is statistically superior/inferior to the compared method (using an improved Friedman test followed by Holm's step-down procedure at a 0.05 significance level).

	IsoReg	Platt	Softmax	SNN
AUC	2.73	2.64	2.39	2.24
ACC	2.45	3.27$*$	2.21	2.06
CrosEntropy	4.00$*$	2.76	1.70	1.55
Brier_Score	2.39	3.45$*$	2.21	1.94
ECE_micro	2.39	3.18$*$	2.64$*$	1.79
MCE_micro	1.91	3.30$*$	2.82$*$	1.97

6.2 APPLICATION IN CAUSAL NETWORK DISCOVERY

In this section we present the results of applying the proposed SNN calibration model in a causal network discovery application. Section 6.2.1 briefly introduces and explains the motivation for the causal network discovery problem. Section 6.2.5 presents the results of our experiments. Section 6.2.6 discusses the results and presents future research directions.

6.2.1 Problem Definition and Motivation

Much of science consists of discovering and modeling causal relationships. Increasingly, scientists have available multiple complex data types and a large number of samples, each of which has an enormous number of measurements recorded, thanks to rapid advancements in sophisticated measurement technology. Some such data may result from experiments in which one or more variables were experimentally manipulated. Often, however, these data are purely observational.

In the past 25 years, there has been tremendous progress in developing general computational methods for representing and discovering causal knowledge from observational data (Glymour and Cooper, 1999; Spirtes et al., 2000; Pearl, 2003; Spirtes, 2010; Illari et al., 2011). A primary use of such methods is to analyze observational data to generate novel causal hypotheses that are likely to

be correct when subjected to experimental validation; such an approach can significantly increase the efficiency of causal discovery in science.

To make informed decisions about which novel causal hypotheses to investigate experimentally, scientists need to know how likely the hypotheses are to be true. In probabilistic terms, this means they need to have the posterior probabilities of the hypotheses be calibrated.

In this section we focus on the discovery of causal Bayesian network (CBN) structures from observational data. In particular, we focus on the discovery of the causal relationships (edge types) between pairs of measured variables. If a causal arc is novel and important, it may be worthwhile to experimentally investigate it. The extent to which it is worth doing so depends on how high is the calibrated probability that the causal arc is present. In this initial investigation of the topic, we consider CBNs without latent confounders, although the extension to latent confounders is straightforward. The method requires the following: (1) a method for generating initial **probability estimates** of the edge types for each pair of variables; in general those estimates need not be well-calibrated, (2) the truth status of a relatively small, unbiased sample of the causal relationships in the network, which we call a **calibration training set**, and (3) a **calibration method** for using the uncalibrated probability estimates and the calibration training set to generate calibrated probabilities for the relatively large number of remaining pairs of variables.

We use a bootstrapping method (Efron and Tibshirani, 1994) for generating probability estimates of edge types. This method resamples a dataset n times with replacement and learns a model for each dataset. In particular, for each dataset we use a heuristic search to find a relatively high scoring CBN. For any given pair of nodes (A, B), the probability they have a given edge type (e.g., $A \rightarrow B$) is estimated as the fraction of that edge type for (A, B) in the n networks. Previously, Friedman, et al. successfully applied this approach for estimating the probabilities of edge types in Bayesian networks (Friedman et al., 1999). These bootstrap estimates are not guaranteed to represent unbiased posterior probabilities, even in the large sample limit of the dataset size and the number of bootstrap samples. A key reason is that heuristic search, while practically necessary, may get stuck in local maxima, even in the large sample limit[4]. Thus, there is a need to map those estimates to calibrated probabilities, which is the focus of the current paper.

[4]Some search methods, such as GES (Chickering, 2003), provide asymptotic guarantees, while other methods, such as greedy forward search, do not.

The bootstrapping approach described above provides an empirically derived estimate of edge-type posterior probabilities for both constraint-based and Bayesian CBN-structure-learning algorithms. Constraint-based CBN learning methods, such as PC and FCI (Spirtes, 2010), do not produce such probability estimates directly. Bayesian model averaging (Madigan et al., 1995; Friedman and Koller, 2003; Koivisto and Sood, 2004; Eaton and Murphy, 2007; Koivisto, 2012) provides an alternative approach for estimating edge probabilities. However, these methods are typically applicable when using real world datasets in which the number of random variables (nodes) is in the double digits (for exact methods) to triple digits (for heuristic methods). In contrast, we are interested in providing calibrated estimates of edge probabilities for datasets that may be much larger. We also note that Bayesian model averaging methods are sensitive to the method applied for heuristic search (Friedman et al., 1999) and to the structure and parameter priors that are used, even if they are non-informative. Consequently, their generated probabilities are still subject to being uncalibrated.

We assume the availability of a calibration training set that allows us to induce a mapping from bootstrap probability estimates to unbiased posterior probabilities. The training set should contain the truth status for the edge types of a sample of node pairs. In the domain of biomedical applications, the truth status might come, for example, from the existing published results in the literature. We emphasize that the calibration training set can be very small, relative to the number of total node pairs. In the experiments we performed, it consists of less than 0.001% of all the node pairs. Using it, we can generate better calibrated probabilities for the remaining 99.999% of node pairs.

We applied SNN to the calibration training set to construct a mapping from bootstrap probability estimates to calibrated posterior probabilities of edge types for all the node pairs in a CBN (except those few that are used for training). We applied that mapping to all of those remaining node pairs.

In our experiments, we used simulated data to investigate two main questions. First, how calibrated are the bootstrap probabilities of edge types? Second, how calibrated are the probabilities produced by our proposed SNN calibration method? Given a finite calibration training set, the latter method is not guaranteed to always output perfectly calibrated probabilities either. *Our main hypothesis in this section is that this calibration method will output probabilities that are more*

calibrated than are the bootstrap probabilities, while being at least as discriminative in terms of precision and recall.

6.2.2 Overview of Greedy Equivalent Search

In this section we briefly describe the causal network discovery algorithm that we applied in our experiments. Chickering (Chickering, 2003) developed an algorithm called Greedy Equivalence Search (GES), which identifies the generative structure of the data by searching over the equivalence classes of Bayesian network structures, which are Directed Acyclic Graphs (DAGs). The equivalence class of DAGs represents a set of DAGs that have the same d-separation properties and can be represented by Partially Directed Acyclic Graphs (PDAGs), also known as patterns. A PDAG is a mixed graph that contains both directed and undirected edges.

GES is a two-phase score-based algorithm that includes a forward equivalence search (FES) and backward equivalence search (BES). It performs as follows: let ε be the current equivalence class, i.e. the current search state or PDAG, and $\boldsymbol{\varepsilon}^{+}(\varepsilon)$ be all equivalence-class neighbors of ε during FES. $\boldsymbol{\varepsilon}^{+}(\varepsilon)$ contains all PDAGs that can be generated by adding a single edge to DAGs in ε (this transformation is based on (Chickering, 1995, 2003)). Similarly, $\boldsymbol{\varepsilon}^{-}(\varepsilon)$ is the equivalence-class neighbors of ε during BES, which is acquired by deleting a single edge from all DAGs in the current state.

The first phase of GES starts with an empty graph (i.e. $\varepsilon = \emptyset$) and replaces the current state with an equivalence class in $\boldsymbol{\varepsilon}^{+}(\varepsilon)$ that has the highest score. It continues this phase until no further local improvement can be achieved. The second phase starts from the local maximum achieved at the first step and performs a backward search by replacing ε with the highest scored equivalence class in $\boldsymbol{\varepsilon}^{-}(\varepsilon)$. It stops when it reaches a local maximum. For more information about this method see (Chickering, 2003).

6.2.3 Bootstrapped Greedy Equivalent Search

Considering the mixed graph generated by GES, it is possible to partition all pairs of nodes (A, B) into the following four possible classes:

1. $A \cdots B$: There is no edge between A and B (i.e., they are marginally independent),

2. $A \rightarrow B$: A has an arc into B,
3. $B \rightarrow A$: B has an arc into A,
4. A — B: A and B are directly dependent, but the direction is indeterminate.

If we assume no latent common causes of the measured variables (i.e., no latent confounding) then the directed arcs above can be interpreted as direct causation, although this assumption and interpretation are not required.

The Bootstrapped Greedy Equivalent Search (BGES) method that we apply has three main steps. In the first step, it performs bootstrap sampling over the training data n times ($n = 200$ in our experiments) to create n different bootstrapped training datasets. In the second step, it runs GES on each of those n datasets, which results in n PDAGs. Finally, for every pair of nodes, it uses the frequency counts of each edge type for that pair over the generated PDAGs to determine a probability distribution for the four possible edge types. Figure 21 shows the graphical representation of the BGES output for two random variables V_1 and V_2 in a hypothetical PDAG.

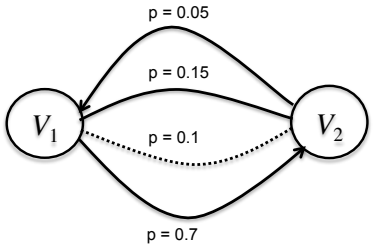


Figure 21: The graphical representation of the BGES output for two hypothetical nodes V_1 and V_2. The labels show the associated probability of each edge type, where the dotted line indicates no edge between V_1 and V_2.

For a pair of nodes (A, B), the resulting output of the BGES method will be four jointly exhaustive and mutually exclusive class probabilities that correspond to the four classes of $A \cdots B$, $A \rightarrow B$, $B \rightarrow A$, and A—B, where the dotted line indicates no edge between A and B. Therefore, to calibrate the generated probabilities by the BGES method, we use our proposed SNN calibration method that post-processes a multi-class classification score (in this case four classes).

6.2.4 Experimental Methods

This section describes the experimental methods that we used to evaluate the performance of the proposed calibrated network discovery method. The evaluation involves the following steps:

1. Create a random Bayesian network, BN.
2. Simulate a dataset, D, of size 1000 from BN, subject to constraints that are described below.
3. Generate 200 bootstrapped datasets, $DB[1..200]$ from D.
4. For each of the bootstrapped datasets, learn a Bayesian network structure (PDAG) using a fast implementation of GES; let $PDAG[1..200]$ designate these PDAGs.
5. For each node pair (A, B), calculate the probability distribution $P_e(A, B)$ of the edge types of (A, B) using maximum likelihood estimates on the counts in $PDAG[1..200]$.
6. Randomly partition the node pairs into train and test sets, where the train set is much smaller than the test set. The training set is intended to represent causal relationships that are known to exist, as reported in the literature.
7. Learn the calibration function f_{cal} from the training data.
8. For each node pair (A, B) in the test set, derive $P_e^{cal}(A, B) = f_{cal}\left(P_e(A, B)\right)$.
9. Compare the performance of P_e^{cal} versus P_e in correctly predicting the data-generating structure of BN for the test set pairs, and doing so in a manner that is well calibrated.

In order to perform the first step, we set the number of nodes in the BN to be 1000 or 5000. We also set the graph density, i.e. average edge per node, to be 1, 2, or 3. In each network, the nodes correspond to discrete random variables that might take two or three values randomly. We parametrize the conditional probability tables (CPTs) for each child node given its parents by uniformly randomly sampling the probability distribution for the child given each state of its parents.

In performing the fourth step, we used a parallelized and optimized implementation of GES, which is called Fast Greedy Search (FGS) (Ramsey, 2015). FGS is much faster than GES and makes it feasible to implement the proposed evaluation framework.

To compute P_e in step 5, we considered four different edge classes for each pair of nodes (A, B) as we described in section 6.2.3: $A \cdots B$, $A \rightarrow B$, $B \rightarrow A$,and A — B. Then, we calculated the

probabilities of these edge classes by counting the observed frequencies in the $PDAG[1..200]$ that are constructed in step 4.

In order to perform step 6, we set the size of the calibration training set to N, where $N = 40, 80,$ or 120. This allows us to evaluate the effect of calibration training size on the prediction performance. To collect N calibration training samples, we selected $N/4$ samples from each of the four edge classes to be observed using stratified random sampling[5]. In particular, we first sorted the probability scores of edges in each edge class according to the bootstrap probabilities. We then partitioned the instances into 5 bins of $\{[0, 0.2), [0.2, 0.4), [0.4, 0.6), [0.6, 0.8), [0.8, 1]\}$ based on their probability scores. Finally, we sampled separately from each bin with equal frequency.

Two common evaluation measures in multi-class classification problems are the accuracy and the Brier-score (a.k.a., mean square error). However, due to the severe class imbalance in our problem, the performance of the predictions before and after calibration were almost always the same in terms of these two measures, up to three decimal points (accuracy for both measures was always close to one and the Brier-score was always close to zero). Thus, to evaluate the performance of the generated probabilities before and after calibration, we used four different edge-type-based evaluation measures. Note that although there are four different edge classes, we consider only three edge types for performance evaluation, because $A \rightarrow B$ and $B \rightarrow A$ are both directed edge types. Table 31 shows the average number of each edge types for each configuration of the BN graphs.

The first two edge-type based evaluation metrics are *precision* (P) and *recall* (R). To compute these metrics for each edge type, we calculated the four basic statistics of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for each of the types separately. Calculating these statistics for the no-edge and undirected-edge type is straightforward since they only consist of the instances that belong to only one of the four target classes[6]. For the directed edge type (which consist of the classes $(A \rightarrow B)$ and $(B \rightarrow A)$), Table 32 shows how we define these statistics. After calculating the four basic statistics for each edge type, the precision will be the ratio of $\frac{TP}{TP+FP}$. Recall is defined as the ratio of $\frac{TP}{TP+FN}$. We also report the F1-score, which is

[5]Note that it is not feasible to use random sampling due to severe class imbalance of the data (i.e., more than 99% of the pairs belong to the no-edge class.)

[6]For instance, the number of true positives for the no-edge type is equal to the number of instances that the predicted class is no-edge (i.e., the associated class probability is the highest among all the four predicted class probabilities), and the true label of the instances are also no-edge.

Table 31: Average number of no-edge, directed-edge, and undirected-edge types over the 10 randomly generated datasets for every configuration of BN. **V** is the number of nodes and **E** is the number of edges.

V, E	**no-edge**	**directed-edge**	**undirected-edge**
1K, 1K	498500	746.6	253.4
1K, 2K	497500	1833.1	166.9
1K, 3K	496500	2886.5	113.5
5K, 5K	12492500	3718.4	1281.6
5K, 10K	12487500	9142.5	857.5
5K, 15K	12482500	14397.6	602.4

defined as the harmonic mean of the precision and recall. F1 is a summary measure that shows the overall performance of the predictions in terms of both precision and recall.

Table 32: True positive (TP), false positive (FP), true negative (TN), and false negative (FN) statistics for the directed-edge type pairs.

Prediction / Truth	$A \cdots B$	$A \rightarrow B$	$B \rightarrow A$	$A — B$
$A \cdots B$	TN	FN	FN	TN
$A \rightarrow B$	FP	TP	FP	FP
$B \rightarrow A$	FP	FP	TP	FP
$A — B$	TN	FN	FN	TN

We also evaluate the edge-type based predictions in terms of maximum calibration error (MCE) (Pakdaman Naeini et al., 2015a). We calculated the MCE for each edge type by partitioning the output space of the associated estimated edge type probabilities, which is the interval [0, 1], into equal frequency bins with 100 instances. The estimated probability for each instance is located in one of the bins. For each bin, we define the associated calibration error as the absolute difference between the mean value of the predictions and the actual observed frequency of positive instances. The MCE calculates the maximum calibration error among the bins. The lower the value of MCE, the better the calibration of the probability scores.

We also report the overall MCE as a summary measure that shows the overall performance of the predictions in terms of calibration. To compute this measure, we augmented the four-element

probability distribution vectors, $P_e(A, B)$ for all test instances to form an aggregated vector P_{all}. We also augmented their corresponding one-hot binary labels (Nielsen, 2015; Abadi et al., 2016) to form an aggregated binary vector Z_{all}. The overall MCE is defined as the maximum calibration error calculated based on P_{all} and Z_{all}.

6.2.5 Experimental Results

This section presents the results of our experiments in evaluating the performance of the generated probabilities for the three edge types before and after calibration. For each set of configurations (e.g., N=80, V = 1000, E = 2000), we report the average results of 10 random runs on 10 randomly simulated Bayesian networks. Tables 33 and 34 report the results of our experiments for random CBNs with 1000 and 5000 nodes, respectively. In these tables bold face indicates the results that are statistically significantly better than the others, based on a two-sided Wilcoxon signed rank test at the 5% significance level. Note that we report only MCE measure for the no-edge type because the precision, recall, and F1 are always very close to 1 for it.

The experimental results in Tables 33 and 34 show that by post-processing the bootstrapped probabilities, we can often improve the overall edge-type performance both in terms of discrimination (i.e., measured by precision and recall) and calibration (i.e., measured by MCE). Also, these tables demonstrate that using a larger calibration training set will result in more accurate and calibrated probabilities.

6.2.6 Discussion

In this section, we introduced a new approach for improving the calibration of CBN structure discovery. We used a bootstrapping method to obtain probabilities of the causal relationships between each pair of random variables. Although we applied the bootstrapping method to the output of the FGS network discovery algorithm (i.e., an optimized and parallelized implementation of the GES method), it can be applied on any other type of the network discovery method, as long as the method is sufficiently fast to run hundreds of times on a dataset. We plan to investigate the performance of this basic approach using other causal discovery algorithms, such as RFCI (Colombo et al., 2012), which models latent confounders.

To calibrate the bootstrapped probabilities, we devised a natural extension of Platt's calibration method that supports multi-class calibration using a shallow neural network. A key advantage of the shallow neural network approach for post-processing the estimated probabilities is that we can readily condition on other types of features for learning a calibration mapping (e.g., features extracted from the structure of the predicted PDAG by the FGS method, such as the indegree of B when we are generating a calibrated probability for the edge type $A \rightarrow B$). Such conditioning on local or global features of the learned graph could potentially yield improvements in the post-processed calibrated probabilities. This is an area for future research. Our experiments show that by using only a small set of instances for training the calibration model, we can obtain substantial improvements in terms of precision, recall, and calibration, relative to the bootstrapped probabilities. Also, as the calibration training set size increased, the performance increased.

Table 33: The results of experiments on CBNs with 1000 variables (i.e., V=1K). N is the number of instances in the calibration training set and E is number of edges in the CBN. Bold face indicates the results that are significantly better, based on the Wilcoxon signed rank test at the 5% significance level. The lower the value of MCE, the better the calibration of the probability scores.

N	E	method	directed edge				undirected edge				no edge	overall
			P	R	F1	MCE	P	R	F1	MCE	MCE	MCE
40	1K	before	0.67	**0.60**	0.63	**0.20**	**0.74**	0.07	0.12	0.47	0.35	0.25
		after	**0.80**	0.58	**0.67**	0.27	0.67	**0.52**	**0.57**	**0.26**	**0.22**	0.23
	2K	before	0.82	0.41	0.55	0.21	0.50	0.02	0.04	0.32	0.54	0.27
		after	0.81	**0.47**	**0.60**	0.25	0.56	**0.35**	**0.38**	**0.25**	**0.24**	**0.23**
	3K	before	0.85	0.25	0.39	0.29	0.37	0.01	0.03	0.18	0.59	0.31
		after	0.84	**0.31**	**0.45**	**0.23**	0.39	**0.24**	**0.27**	0.22	**0.31**	**0.25**
80	1K	before	0.67	**0.59**	0.63	0.21	0.72	0.04	0.08	0.49	0.32	0.26
		after	**0.83**	0.58	**0.68**	0.16	0.68	**0.57**	**0.61**	**0.16**	**0.14**	**0.16**
	2K	before	0.82	0.41	0.54	0.20	0.15	0.00	0.01	0.32	0.55	0.27
		after	**0.84**	**0.47**	**0.60**	**0.16**	**0.55**	**0.35**	**0.40**	**0.17**	**0.18**	**0.17**
	3K	before	0.86	0.25	0.39	0.28	0.03	0.00	0.00	0.18	0.58	0.31
		after	0.86	**0.30**	**0.45**	**0.18**	**0.37**	**0.21**	**0.24**	0.15	**0.26**	**0.21**
120	1K	before	0.67	**0.59**	0.63	0.21	0.60	0.02	0.04	0.48	0.28	0.26
		after	**0.86**	0.56	**0.68**	**0.12**	0.68	**0.63**	**0.65**	**0.12**	**0.10**	**0.12**
	2K	before	0.83	0.40	0.54	0.23	0.07	0.00	0.00	0.31	0.54	0.26
		after	**0.85**	**0.46**	**0.60**	**0.15**	**0.55**	**0.33**	**0.40**	**0.13**	**0.17**	**0.15**
	3K	before	0.86	0.25	0.38	0.29	0.00	0.00	0.00	0.17	0.59	0.32
		after	0.87	**0.30**	**0.44**	**0.18**	**0.46**	**0.26**	**0.31**	**0.12**	**0.26**	**0.22**

Table 34: The results of experiments on CBNs with 5000 variables (i.e., V=5K). N is the number of instances in the calibration training set and E is number of edges in the CBN. Bold face indicates the results that are significantly better, based on the Wilcoxon signed rank test at the 5% significance level. The lower the value of MCE, the better the calibration of the probability scores.

N	E	method	directed edge				undirected edge				no edge	overall
			P	R	F1	MCE	P	R	F1	MCE	MCE	MCE
40	5K	before	0.64	**0.57**	0.60	0.32	0.00	0.00	0.00	0.68	0.48	0.36
		after	**0.81**	0.52	**0.63**	0.32	**0.63**	**0.57**	**0.59**	**0.33**	**0.35**	0.31
	10K	before	0.79	0.38	0.52	**0.25**	0.00	0.00	0.00	0.48	0.62	0.35
		after	**0.82**	**0.43**	**0.57**	0.30	**0.47**	**0.28**	**0.31**	**0.35**	**0.32**	0.30
	15K	before	0.83	0.24	0.37	0.32	0.00	0.00	0.00	0.34	0.64	0.37
		after	0.83	**0.28**	**0.42**	0.33	**0.35**	**0.22**	**0.24**	0.30	**0.36**	0.34
80	5K	before	0.64	**0.57**	0.60	0.32	0.00	0.00	0.00	0.67	0.49	0.35
		after	**0.83**	0.54	**0.65**	**0.23**	**0.66**	**0.62**	**0.63**	**0.24**	**0.24**	**0.19**
	10K	before	0.80	0.38	0.52	0.25	0.00	0.00	0.00	0.47	0.62	0.35
		after	**0.84**	**0.43**	**0.57**	**0.22**	**0.51**	**0.34**	**0.38**	**0.25**	**0.25**	**0.22**
	15K	before	0.83	0.24	0.37	0.31	0.00	0.00	0.00	0.35	0.63	0.38
		after	0.84	**0.29**	**0.43**	**0.23**	**0.40**	**0.23**	**0.27**	**0.24**	**0.29**	**0.24**
120	5K	before	0.64	**0.57**	0.60	0.31	0.00	0.00	0.00	0.67	0.48	0.36
		after	**0.85**	0.53	**0.65**	**0.19**	**0.66**	**0.64**	**0.65**	**0.19**	**0.19**	**0.16**
	10K	before	0.80	0.38	0.52	0.25	0.00	0.00	0.00	0.48	0.61	0.36
		after	**0.85**	**0.43**	**0.57**	**0.18**	**0.51**	**0.38**	**0.42**	**0.19**	**0.23**	**0.20**
	15K	before	0.83	0.24	0.37	0.32	0.00	0.00	0.00	0.35	0.63	0.38
		after	**0.86**	**0.28**	**0.42**	**0.23**	**0.42**	**0.27**	**0.32**	**0.19**	**0.31**	**0.25**

7.0 CONCLUSION AND FUTURE WORK

In this dissertation, we introduced several new binary classifier calibration methods by extending the most commonly used calibration methods namely, histogram binning, isotonic regression, and Platt's method. The methods we introduced are classifier independent; thus, they can be readily applied on the output of any existing binary classification model to calibrate the associated classification scores. We also proved theorems that collectively show, in the presence of large calibration samples, that it is possible to obtain perfectly calibrated predictions using a simple equal frequency histogram binning method, while the worst-case AUC deterioration between the post-processed calibrated estimates, and the original predictions made by the base classifier, converges to zero. These theoretical results support our hypothesis that *by using simple post-processing calibration methods, it is possible to improve the calibration capability of a classifier without sacrificing much discrimination capability*. In addition, using experiments on a wide range of simulated and real data, we showed that our newly introduced calibration methods, including BBQ, ENIR, and ELiTE, can improve the calibration performance of predictions without losing any statistically significant discrimination in terms of area under ROC curve (AUC) and accuracy (ACC).

The experimental results of the newly introduced binary classifier calibration methods support our second hypothesis in Section 1.1 that *"one can systematically produce probabilities that are more accurate than those obtained from existing calibration methods by using an ensemble of calibration models that are generated based on rational, realistic assumptions about the output of the classifier."* For instance, BBQ can be considered as an ensemble of equal frequency histogram binning models. ENIR is also an ensemble of near-isotonic regression-based models. In addition, ENIR makes a more reasonable assumption in finding the calibration mapping compared to IsoReg and BBQ. Unlike IsoReg, which assumes that the probabilities output by a classifier are strictly isotonic, ENIR assumes that the predictions made by the classifier are approximately (near) isotonic.

Our experiments in Section 4.3 show that the near-isotonic assumption is less biased compared to isotonicity assumption made by IsoReg.

The ELiTE algorithm is an ensemble of linear trend-filtering signal-approximation models. ELiTE extends the binning-based calibration methods by finding a calibration mapping that is piecewise linear. In contrast, in all the binning based calibration methods —including histogram binning, isotonic regression, ABB, BBQ, and ENIR— the final calibration mapping is a piecewise constant function. Our experimental results show that while the piecewise linear assumption is more flexible and improves the overall performance of the post-processed estimates, it is constrained enough to not to overfit the training data, even when we use the training data for both model construction and calibration mapping. Our experimental results show that our newly introduced calibration methods are usually superior to commonly used calibration methods.

We also introduced the SNN calibration model that is an extension to the Platt's method using a shallow neural network and cross-entropy loss function for multi-class calibration. Our results show that the proposed calibration model performs well in terms of improving calibration performance of predictions while maintaining discrimination performance in terms of accuracy and AUC. The proposed model also performs well when the number of calibration training cases is small (i.e., in order of tens of cases). In particular, we showed its superior performance on an extrinsic application in obtaining well-calibrated causal network discovery methods in which there are 4 different target classes. Our experiments show that by using only a small set of instances for training the calibration model (less than 0.001% of existing node pairs), we can obtain substantial improvements in terms of precision, recall, and calibration, relative to bootstrapped probabilities. A key advantage of the shallow neural network (SSN) approach in calibrating causal discovery methods (e.g., Greedy Equivalent Search (Chickering, 2003)) is that we can readily condition on other types of features for learning a calibration mapping (e.g., features extracted from the structure of the predicted casual network, such as the indegree of B when we are generating a calibrated probability for the edge type $A \rightarrow B$). Such conditioning on local or global features of the learned graph could potentially yield improvements in the post-processed calibrated probabilities. This is an area for future research.

There are a number of areas for future research. In terms of calibration evaluation measures, in our experiments, we always used $K = 10$ equal frequency intervals in defining the MCE and ECE

evaluation measures. Further development of methods for evaluating calibration, beyond using just 10 intervals, is an interesting future research area. Even though our convergence theorems for ECE and MCE for the quantile binning method remain valid as long as the number of intervals is constant, in the finite sample situation, ECE and MCE may be affected by changing the number of intervals similar to what has been observed in some applications of Hosmer-Lemeshow test measure (Allison, 2014).

In terms of theoretical works, it would be interesting to show convergence results, similar to those that we proved for histogram binning, for BBQ, ENIR, and ELiTE. Our experimental results on simulated and real data show that these methods are superior to histogram binning, isotonic regression, and Platt's method. Also, our results on a wide range of simulated and real datasets show that they are able to improve calibration performance of the baseline classifiers (i.e., LR, SVM, NB) without losing statistically significant discrimination in terms of ACC and AUC. As we described in Chapter 5, there are also existing theoretical results for the convergence of excess risk for the equal-size histogram binning method. We conjecture that similar convergence theorems can be developed for quantile binning methods. It is also possible to develop ways for deriving error bounds on the calibrated probabilities. This can be readily done with histogram binning, ABB, SBB, and BBQ, using concentration of measure inequalities (e.g., using Hoeffding inequality). However, it is less obvious how to do so with the other new methods (e.g., ELiTE).

Another future research direction would be developing large-scale (in terms of the number of classes) multi-class and multi-label calibration methods. Also, all of the experiments in this thesis used all of the classifier training data for calibration training as well. It would be interesting to study how using a separate dataset for calibration training would affect calibration performance.

Finally, in terms of the application in causal network discovery, it would be interesting to explore methods that construct a calibration dataset automatically, without the need for a dataset for which the causal truth-status is known. This is an important capability in domains in which obtaining the ground truth is not practical. It would also be beneficial to investigate the performance of our proposed framework on other score-based and constrained-based causal discovery methods.

8.0 BIBLIOGRAPHY

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., and Roth, D. (2006). Generalization bounds for the area under the roc curve. *Journal of Machine Learning Research*, 6(1):393.

Allison, P. D. (2014). Measures of fit for logistic regression. In *Proceedings of the SAS Global Forum 2014 Conference*.

Antoniak, C. (1974). Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, pages 1152–1174.

Bache, K. and Lichman, M. (2013). UCI machine learning repository.

Barlow, R. E., Bartholomew, D. J., Bremner, J., and Brunk, H. D. (1972). *Statistical inference under order restrictions: The theory and application of isotonic regression*. Wiley New York.

Bella, A., Ferri, C., Hernández-Orallo, J., and Ramírez-Quintana, M. J. (2009). Similarity-binning averaging: a generalisation of binning calibration. In *Intelligent Data Engineering and Automated Learning-IDEAL 2009*, pages 341–349. Springer.

Bella, A., Ferri, C., Hernández-Orallo, J., and Ramírez-Quintana, M. J. (2013). On the effect of calibration in classifier combination. *Applied Intelligence*, 38(4):566–585.

Bennett, P. N. (2000). Assessing the calibration of naive bayes posterior estimates. Technical report, DTIC Document.

Bennett, P. N. (2003). Using asymmetric distributions to improve text classifier probability estimates. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 111–118. ACM.

Bickel, J. E. (2007). Some comparisons among quadratic, spherical, and logarithmic scoring rules. *Decision Analysis*, 4(2):49–65.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Caruana, R. and Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168.

Cavanaugh, J. E. (1997). Unifying the derivations for the Akaike and corrected Akaike information criteria. *Statistics & Probability Letters*, 33(2):201–208.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

Chickering, D. M. (1995). A transformational characterization of equivalent bayesian network structures. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 87–98. Morgan Kaufmann Publishers Inc.

Chickering, D. M. (2003). Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554.

Cohen, I. and Goldszmidt, M. (2004). Properties and benefits of calibrated classifiers. In *Knowledge Discovery in Databases: PKDD 2004*, pages 125–136. Springer.

Colombo, D., Maathuis, M. H., Kalisch, M., Richardson, T. S., et al. (2012). Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40(1):294–321.

Cooper, G. F. and Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347.

DeGroot, M. and Fienberg, S. (1983). The comparison and evaluation of forecasters. *The Statistician*, pages 12–22.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning*, pages 231–238.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Eaton, D. and Murphy, K. P. (2007). Exact Bayesian structure learning from uncertain interventions. In *International Conference on Artificial Intelligence and Statistics*, pages 107–114.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499.

Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.

Engel, A. (1998). *Problem-solving strategies*. Springer New York.

Escobar, M. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, pages 577–588.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Fawcett, T. (2004). Roc graphs: Notes and practical considerations for researchers. *Machine learning*, 31:1–38.

Fawcett, T. and Niculescu-Mizil, A. (2007). Pav and the roc convex hull. *Machine Learning*, 68(1):97–106.

Feng, C., Sutherland, A., King, R., Muggleton, S., and Henery, R. (1993). Comparison of machine learning classifiers to statistics and neural networks. *AI&Statistics-93*, 6:41.

Ferguson, T. (1973). A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230.

Fine, M., Auble, T., Yealy, D., Hanusa, B., Weissfeld, L., Singer, D., Coley, C., Marrie, T., and Kapoor, W. (1997). A prediction rule to identify low-risk patients with community-acquired pneumonia. *New England Journal of Medicine*, 336(4):243–250.

Freedman, D. and Diaconis, P. (1981). On the histogram as a density estimator: L 2 theory. *Probability theory and related fields*, 57(4):453–476.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701.

Friedman, N., Goldszmidt, M., and Wyner, A. (1999). Data analysis with bayesian networks: A bootstrap approach. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 196–205. Morgan Kaufmann Publishers Inc.

Friedman, N. and Koller, D. (2003). Being Bayesian about network structure. a Bayesian approach to structure discovery in Bayesian networks. *Machine learning*, 50(1-2):95–125.

Gama, J. and Brazdil, P. (2000). Cascade generalization. *Machine Learning*, 41(3):315–343.

Garczarek, U. (2002). *Classification rules in standardized partition spaces*. PhD thesis, Universität Dortmund.

Garthwaite, P. H., Kadane, J. B., and O'Hagan, A. (2005). Statistical methods for eliciting probability distributions. *Journal of the American Statistical Association*, 100(470):680–701.

Gebel, M. and Weihs, C. (2007). Calibrating classifier scores into probabilities. In *Advances in Data Analysis*, pages 141–148. Springer.

Gill, P. E., Murray, W., and Wright, M. H. (1981). *Practical optimization*, volume 5. Academic press London.

Glymour, C. N. and Cooper, G. F. (1999). *Computation, causation, and discovery*. MIT Press.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Good, I. J. (1952). Rational decisions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 107–114.

Hand, D. J. and Till, R. J. (2001). A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45(2):171–186.

Hashemi, H. B., Yazdani, N., Shakery, A., and Naeini, M. P. (2010). Application of ensemble models in web ranking. In *5th International Symposium on Telecommunications (IST)*, pages 726–731. IEEE.

Heckerman, D., Geiger, D., and Chickering, D. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243.

Hestenes, M. R. (1969). Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). Bayesian model averaging: a tutorial. *Statistical Science*, pages 382–401.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, pages 65–70.

Illari, P., Russo, F., and Williamson, J. (2011). *Causality in the Sciences*. OUP Oxford.

Iman, R. L. and Davenport, J. M. (1980). Approximations of the critical region of the friedman statistic. *Communications in Statistics-Theory and Methods*, 9(6):571–595.

Jiang, L., Zhang, H., and Su, J. (2005). Learning k-nearest neighbor naive bayes for ranking. In *Advanced Data Mining and Applications*, pages 175–185. Springer.

Jiang, X., Osl, M., Kim, J., and Ohno-Machado, L. (2012). Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274.

Johnson, N. A. (2013). A dynamic programming algorithm for the fused lasso and ℓ_0 segmentation. *Journal of Computational and Graphical Statistics*, 22(2):246–260.

Kim, S.-J., Koh, K., Boyd, S., and Gorinevsky, D. (2009). \ell_1 trend filtering. *SIAM review*, 51(2):339–360.

Klemela, J. (2009). Multivariate histograms with data-dependent partitions. *Statistica Sinica*, 19(1):159.

Koivisto, M. (2012). Advances in exact Bayesian structure discovery in Bayesian networks. *arXiv preprint arXiv:1206.6828*.

Koivisto, M. and Sood, K. (2004). Exact Bayesian structure discovery in Bayesian networks. *The Journal of Machine Learning Research*, 5:549–573.

Korb, K. B. and Nicholson, A. E. (2010). *Bayesian artificial intelligence*. CRC press.

Kurihara, K., Welling, M., and Vlassis, N. (2007). Accelerated variational dirichlet process mixtures. *Advances in Neural Information Processing Systems*, 19:761.

Lachiche, N. and Flach, P. (2003). Improving accuracy and cost of two-class and multi-class probabilistic classifiers using roc curves. In *ICML*, pages 416–423.

Lustgarten, J., Visweswaran, S., Gopalakrishnan, V., and Cooper, G. (2011). Application of an efficient bayesian discretization method to biomedical data. *BMC Bioinformatics*, 12.

MacEachern, S. and Muller, P. (1998). Estimating mixture of dirichlet process models. *Journal of Computational and Graphical Statistics*, pages 223–238.

Madigan, D., York, J., and Allard, D. (1995). Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, pages 215–232.

Mayer, U. F. and Sarkissian, A. (2003). Experimental design for solicitation campaigns. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 717–722. ACM.

Menon, A., Jiang, X., Vembu, S., Elkan, C., and Ohno-Machado, L. (2012). Predicting accurate probabilities with a ranking loss. In *Proceedings of the International Conference on Machine Learning*, pages 703–710.

Niculescu-Mizil, A. and Caruana, R. (2005a). Obtaining calibrated probabilities from boosting. In *UAI*, page 413.

Niculescu-Mizil, A. and Caruana, R. (2005b). Predicting good probabilities with supervised learning. In *Proceedings of the International Conference on Machine Learning*, pages 625–632.

Nielsen, M. A. (2015). Neural networks and deep learning. *Determination Press*.

Nowak, R. D. (2009). Lecture notes: Decision trees and classification. [Online; accessed 26-June-2016].

Pakdaman Naeini, M., Cooper, G., and Hauskrecht, M. (2015a). Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Pakdaman Naeini, M., Cooper, G. F., and Hauskrecht, M. (2015b). Binary classifier calibration using a bayesian non-parametric approach. In *SIAM Data Mining (SDM)*.

Pakdaman Naeini, M., Cooper, G. F., and Hauskrecht, M. (2015c). Binary classifier calibration using a Bayesian non-parametric approach. In *SIAM Data Mining (SDM)*.

Parmigiani, G. and Inoue, L. (2009). *Decision theory: Principles and approaches*, volume 812. John Wiley & Sons.

Pearl, J. (2003). Causality: models, reasoning and inference. *Economet. Theor*, 19:675–685.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74.

Ramdas, A. and Tibshirani, R. J. (2014). Fast and flexible admm algorithms for trend filtering. *arXiv preprint arXiv:1406.2082*.

Ramsey, J. D. (2015). Scaling up greedy equivalence search for continuous variables. *arXiv preprint arXiv:1507.07749*.

Rüping, S. (2004). A simple method for estimating conditional probabilities for svms. Technical report, Universität Dortmund.

Rüping, S. (2006). Robust probabilistic calibration. In *Machine Learning: ECML 2006*, pages 743–750. Springer.

Russell, S. and Norvig, P. (2009). *Artificial intelligence: A modern approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Scott, C. and Nowak, R. (2002). Dyadic classification trees via structural risk minimization. In *Advances in Neural Information Processing Systems*, pages 359–366.

Scott, C. and Nowak, R. (2003). Near-minimax optimal classification with dyadic classification trees. *Advances in Neural Information Processing Systems*, 16.

Scott, C., Nowak, R. D., et al. (2006). Minimax-optimal classification with dyadic decision trees. *IEEE transactions on information theory*, 52(4):1335–1353.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*, volume 26. Chapman & Hall/CRC.

singh, A. (2011). Lecture notes: Analysis of histogram and decision tree classifiers. [Online; accessed 26-June-2016].

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222.

Spirtes, P. (2010). Introduction to causal inference. *The Journal of Machine Learning Research*, 11:1643–1662.

Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, prediction, and search*. MIT press.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.

Tibshirani, R. and Taylor, J. (2011). The solution path of the generalized lasso. *The Annals of Statistics*, 39(3):1335–1371.

Tibshirani, R. J. (2014). Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323.

Tibshirani, R. J., Hoefling, H., and Tibshirani, R. (2011). Nearly-isotonic regression. *Technometrics*, 53(1):54–61.

Wasserman, L. (2006). *All of nonparametric statistics*. Springer.

Zadrozny, B. and Elkan, C. (2001a). Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 204–213. ACM.

Zadrozny, B. and Elkan, C. (2001b). Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *International Conference on Machine Learning*, pages 609–616.

Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 694–699.

Zhang, H. and Su, J. (2004). Naive bayesian classifiers for ranking. In *Machine Learning: ECML 2004*, pages 501–512. Springer.

Zhong, L. W. and Kwok, J. T. (2013). Accurate probability calibration for multiple classifiers. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1939–1945. AAAI Press.

Zonneveldt, S., Korb, K., and Nicholson, A. (2010). Bayesian network classifiers for the german credit data. Technical report, Technical report, 2010/1, Bayesian Intelligence. http://www. Bayesian-intelligence. com/publications. php.