This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2895794, IEEE Transactions on Pattern Analysis and Machine Intelligence

Calibrating Classification Probabilities with Shape-restricted Polynomial Regression

Yongqiao Wang, Lishuai Li, and Chuangyin Dang, *Senior Member, IEEE*

***Abstract*—In many real-world classification problems, accurate prediction of membership probabilities is critical for further decision making. The probability calibration problem studies how to map scores obtained from one classification algorithm to membership probabilities. The requirement of non-decreasingness for this mapping involves an infinite number of inequality constraints, which makes its estimation computationally intractable. For the sake of this difficulty, existing methods failed to achieve four desiderata of probability calibration: universal flexibility, non-decreasingness, continuousness and computational tractability. This paper proposes a method with shape-restricted polynomial regression, which satisfies all four desiderata. In the method, the calibrating function is approximated with monotone polynomials, and the continuously-constrained requirement of monotonicity is equivalent to some semidefinite constraints. Thus, the calibration problem can be solved with tractable semidefinite programs. This estimator is both strongly and weakly universally consistent under a trivial condition. Experimental results on both artificial and real data sets clearly show that the method can greatly improve calibrating performance in terms of reliability-curve related measures.**

***Index Terms*—Classification calibration, probability prediction, isotonic regression, semidefinite programming, polynomial regression**

I. Introduction

In binary classification, one aims to find a labeling function $l : \mathscr{X} \to \{0,1\}$ based on a training data $\{(\mathbf{x}_n, y_n)\}_{n=1}^N \subset \mathscr{X} \times \{0,1\} \subset \mathbb{R}^d \times \{0,1\}$ from the distribution of $(\mathbf{X}, Y)$. To achieve this goal, commonly one first obtains a scoring function $s : \mathscr{X} \to \mathbb{R}$, then predicts the label of one test sample $\mathbf{x} \in \mathscr{X}$ according to whether its score $s(\mathbf{x})$ is above or below a threshold $\mathfrak{h} \in \mathbb{R}$. If $s(\mathbf{x}) \geq \mathfrak{h}$, $\mathbf{x}$ is predicted to be from class 1, otherwise it is predicted to be from class 0. A good scoring function is expected to rank the samples in a data set from the most probable member to the least probable member of class 1. That is, for any two samples $\mathbf{x}_1$ and $\mathbf{x}_2$, if $s(\mathbf{x}_1) > s(\mathbf{x}_2)$, then $\mathbb{P}\{Y=1|\mathbf{X}=\mathbf{x}_1\} \geq \mathbb{P}\{Y=1|\mathbf{X}=\mathbf{x}_2\}$.

However, in many real applications, labeling or ranking of samples is insufficient for further decision-making tasks. Often what is needed is an accurate estimate of the probability that a sample is a member of the class of interest. Probability prediction models for classification aim to obtain the function $g^* : \mathscr{X} \to [0,1]$ that minimizes the L_2 risk, i.e.

$$g^* := \arg \min_{g:\mathscr{X}\to[0,1]} \int [g(\mathbf{x}) - y]^2 \, \mathrm{d}G(\mathbf{x}, y) \tag{1}$$

where G denotes the underlying distribution of $(\mathbf{X}, Y)$. A probability prediction model $g(\cdot)$ is called perfectly calibrated, if it satisfies

$$\forall p \in [0,1], \quad \mathbb{P}_{(\mathbf{X},Y)\sim G}\{Y=1|g(\mathbf{X})=p\} = p. \tag{2}$$

Certainly the conditional $g(\mathbf{x}) = \mathbb{P}\{Y=1|\mathbf{X}=\mathbf{x}\}$ is perfectly calibrated. However estimating $\mathbb{P}\{Y=1|\mathbf{X}=\mathbf{x}\}$ is almost computationally prohibitive, because in practice $\mathbf{x}$ is often a mix of multiple continuous, binary and categorical attributes.

An alternative strategy is the post-processing way that relies on existing classification models and calibrates their scores with post-processing methods. It consists of two steps. First, train a state-of-the-art classification model, such as support vector machine (SVM) and neural networks (NN), to obtain classification scores $\{s(\mathbf{x}_n)\}_{n=1}^N$. Second, calibrate these scores with their class labels to obtain a calibrating (or monotone link) function for probability prediction. The objective is to seek the calibrating function $f^* : s(\mathscr{X}) \to [0,1]$ that satisfies

$$f^* := \arg \min_{f:s(\mathscr{X})\to[0,1]} \int [f(s) - y]^2 \mathrm{d}G_s(s, y) \tag{3}$$

where G_s is the joint distribution of $S := s(\mathbf{X})$ and Y. $f(\cdot)$ is called perfectly calibrated for classifier s, if it satisfies

$$\forall p \in [0,1], \quad \mathbb{P}_{(S,Y)\sim G_s}\{Y=1|f(S)=p\} = p. \tag{4}$$

An ideal calibrating function is $f(s) = \mathbb{P}\{Y=1|S=s\}$.

By transforming multivariate estimation to univarite estimation, the classifier-s calibration (3) provides a simplification of the general probability prediction problem (1). This simplification has a great advantage that it frees the modeler from modifying the learning procedure and the associated optimization method. Of course the performance of overall probability prediction of $f(s(\cdot))$ depends on the scoring function $s(\cdot)$ and the calibrating function $f(\cdot)$ [1]. This paper studies only the estimation of $f(\cdot)$ under the assumption that we have got $s(\cdot)$. The general probability prediction that estimates both $s(\cdot)$ and $f(\cdot)$, such as [2], [3] is not the purpose of this paper.

This probability calibration is pre-requisite for many state-of-the-art classifiers, such as SVM and NN. These classifiers only aim to obtain powerful discriminant power, thus generate scoring functions that have no direct connection

This work was supported by National Natural Science Foundation of China (71571163).

Y. Wang is with the School of Finance, Zhejiang Gongshang University, Hangzhou 310018, Zhejiang, P.R. China (email: wangyq@zjsu.edu.cn)

L. Li and C. Dang are with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Kowloon, Hong Kong (lishuai.li@cityu.edu.hk and mecdang@cityu.edu.hk)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2895794, IEEE Transactions on Pattern Analysis and Machine Intelligence

with membership probability [4]. One attempt to build a classification model is taking probability prediction explicitly into consideration ab initio. But this attempt requires one to carry out a major modification of the objective function, e.g., by imposing probability-related penalty term and using a different type of loss function. It could greatly increase the model complexity and computational cost of the associated optimization program.

This probability calibration is also pre-requisite for many probability-related classifiers, such as naive Bayesian (NB) and logistic regression (LR), which assign to each sample a score between 0 and 1 that can be interpreted, in practice, as an estimate of its membership probability. However, NB is well known for its extreme probability prediction, which means that most of probability predictions are either near to 0 or near to 1. The reason behind it is the inappropriate assumption that all attributes are independent given the class of the samples. LR, as the workhorse in classification probability prediction, suffers two clear shortcomings. First, its parametric form $s(\cdot)$ is too simple to be flexible for complex classification tasks in practice. Second, its probability estimate is biased [5], [6].

The criteria to use in judging a probability calibration method that we will consider in this paper are:

(1) Universal flexibility. Because of diverse underlying distributions of $(\mathbf{X}, Y)$ and various classification algorithms $s(\cdot)$, the calibrating function $f(s) = \mathbb{P}\{Y = 1|s(\mathbf{X}) = s\}$ is often very complicated. A good calibration method is expected to have the ability to fit any $f(\cdot)$. An estimator without universal flexibility fails to approximate well even when the data size approaches infinite.

(2) Non-decreasingness. The estimated calibrating function should be non-decreasing over the entire range under any scenario of training data.

(3) Continuousness. In some previous nonparametric methods, the estimated calibrating function is forced to be piecewise constant, as shown in Fig. 2. It seems unsound that an arbitrarily small change of decision value brings about a large change of classification probability. Jiang et al. [7] show that imposing the requirement of smoothness can significantly improve calibrating performance, because discontinuous functions are too free for fitting.

(4) Computational tractability. The involved computational time should be polynomial with the training size N.

Many methods have been proposed for the probability calibration problem. Among them, the most popular parametric method is Platt [8], in which the calibrating function is assumed to be a sigmoid function. The major shortcoming of parametric methods is the lack of flexibility and the vulnerability to mis-specification. Nonparametric methods also have been proposed, including histogram binning and isotonic regression. Section II provides a comprehensive survey on these calibration methods for binary classification, which are summarized in Table I. One can also refer to Fig. 1 for a vivid comprehension of main characteristics of these methods. However, these existing methods failed to satisfy all of the above four criteria.

The motivation of this paper is to obtain a universally flexible, monotone and continuous estimate of calibrating functions. The requirement of monotonicity involves an infinite number of inequality constraints, because it should be satisfied by every point of the domain $s(\mathscr{X})$. Non-decreasing function values at finite points don't guarantee that the function is non-decreasing over the whole domain. Worse than this, semi-infinite optimization theory [17], [18] shows that the function cannot be guaranteed to be non-decreasing over the whole range, even when the monotonicity is satisfied at an infinite number of points.

This paper proposes to solve the calibration problem with shape-restricted polynomial regression (RPR). This method has the following advantages. First, compared with many existing methods, it has universal flexibility and can fit any complex continuous calibrating function as the polynomial degree increases to infinite. Second, it strictly conforms to the requirement of monotonicity, which is a great advantage over many existing nonparametric models. Its estimated calibrating function is guaranteed to be non-decreasing over the entire range, and all calibrated probabilities are guaranteed to fall in the unit interval [0,1]. Third, the estimated calibrating function is continuous over the entire range. Fourth, its estimation can be solved with a computationally tractable semidefinite program, which can be solved with off-the-shelf optimization toolboxes, e.g. CVX [19].

The remainder of this paper is structured as follows. Section II reviews previous probability calibration methods, including both parametric and nonparametric. Section III is the methodology part that presents the proposed method. Section IV is the theoretical part that verifies its universal flexibility and statistical convergence. The novelty of the proposed method is illustrated by some experiments in Section V. This paper is concluded in Section VI.

In this paper, scalars are written in normal letters, while all vectors are column written in boldfaced lowercase letters and all matrices are written in boldfaced uppercase letters. The i-th element of vector $\mathbf{a}$ is a_i, while the row-i column-j element of matrix $\mathbf{U}$ is U_{ij}. $\mathbb{R}$ denotes the real field, $\mathbb{R}_+$ denotes its subset of nonnegative real numbers, and $\mathbb{R}_+^{k\times k}$ is the cone of k-dimension positive semidefinite symmetric real matrices. For two matrices $\mathbf{U}$, $\mathbf{V} \in \mathbb{R}^{k\times k}$, $\langle \mathbf{U}, \mathbf{V}\rangle$ is the inner product of $\mathbf{U}$ and $\mathbf{V}$, i.e. $\langle \mathbf{U}, \mathbf{V}\rangle = \sum_{i=1}^{k}\sum_{j=1}^{k} U_{ij}V_{ij}$. $\mathbb{I}\{\cdot\}$ is the indicator function. $\mathbb{E}(Y)$ denotes the expectation of random variable Y. $\mathbb{P}\{A\}$ denotes the probability of event A. $N(\mu, \sigma^2)$ denotes the normal distribution with mean μ and standard deviation σ^2. $\mathcal{C}[a, b]$ is the set of functions that is continuous on the closed interval $[a, b]$.

II. Related work

For the simplicity of notation, let $s_n := s(\mathbf{x}_n)$. We also assume that these scores have been sorted in ascending order, i.e. $s_1 \leq s_2 \leq \cdots \leq s_N$. The calibration problem studies how to estimate $f(\cdot)$ from the training data $\{s_n, y_n\}_{n=1}^{N}$.

A. Parametric methods

Hastie and Tibshirani [20] calibrates the SVM scoring function $s(\cdot)$ on the assumption that two class conditional distributions $\mathbb{P}\{S|Y = 0\}$ and $\mathbb{P}\{S|Y = 1\}$ are Gaussian

TABLE I
QUALITATIVE COMPARISON BETWEEN CALIBRATION METHODS

Name	Method	Function	Flexibility	Monotonicity	Continuousness	Complexity
Individual						
Platt [8]	Logit regression	Sigmoid	-	+	+	$O(NT)$
HistBin [9]	Histogram binning	Piecewise constant	+	-	-	$O(N \log N)$
IsoReg [10]	Isotonic regression	Stepwise constant	+	+	-	$O(N \log N)$
NearIso [11], [12]	Nearly isotonic regression	Piecewise constant	o	-	-	$O(N \log N)$
LiTE [12], [13]	ℓ_1-linear trend filtering	Piecewise linear	o	-	+	$O(N \log N)$
ACP [14]	Adaptive calibration	Piecewise constant	o	-	-	$O(N \log N)$
SmoIsoReg [7]	Isotonic splines interpolation	Cubic splines	-	+	+	$O(N^2)$
RPR(this paper)	Restricted polynomial regression	Polynomial	+	+	+	$O(N^2)$
Ensemble						
BBQ [15], [16]	Ensemble of HistBin	Piecewise constant	+	-	-	$O(N \log N)$
ENIR [11], [12]	Ensemble of NearIso	Piecewise constant	o	-	-	$O(N^2)$
ELITE [12], [13]	Ensemble of LiTE	Piecewise linear	o	-	+	$O(N \log N)$

In the last four columns, +:satisfied, -: unsatisfied, o:unknown. T is the number of iterations required for convergence of the Platt method.

with different means and same standard deviation, i.e. $S|Y = 0 \sim N(\mu_0, \sigma^2)$, $S|Y = 1 \sim N(\mu_1, \sigma^2)$. Three distribution parameters can be estimated by

$$\mu_\ell = \frac{\sum_{n=1}^N s_n \mathbb{I}_{\{y_n=\ell\}}}{\sum_{n=1}^N \mathbb{I}_{\{y_n=\ell\}}}, \quad \ell \in \{0,1\} \tag{5}$$

$$\sigma^2 = \frac{1}{N-1} \sum_{n=1}^N \left(s_n - \frac{1}{N} \sum_{m=1}^N s_m \right)^2 . \tag{6}$$

With the aid of the Bayes rule, we can obtain the posterior classification probability that has the sigmoid form

$$f(s) = \mathbb{P}\{Y = 1 | S = s\} = 1/(1 + \exp\{a_0 + a_1 s\}) \tag{7}$$

where a_0 and a_1 are two constants.

Please note that in this model the standard deviations for $S|Y = 0$ and $S|Y = 1$ must be the same. Or else the posterior classification probability has the following form

$$\mathbb{P}\{Y = 1 | S = s\} = 1/(1 + \exp\{a_0 + a_1 s + a_2 s^2\}), \tag{8}$$

which evidently violates the requirement of monotonicity.

The major shortcoming of [20] is the assumption of Gaussian distributions for two class conditionals. For example, as shown in [8], the scoring values from SVM are far from Gaussian. There are discontinuities in the derivatives of both densities at two margins $s(\mathbf{x}) = \pm 1$, because the hinge loss of SVMs has discontinuities at ± 1.

To free from the Gaussian assumption, Platt [8] estimates the sigmoid function directly from $\{(s_n, y_n)\}_{n=1}^N$. Two parameters a_0 and a_1 can be obtained using maximum likelihood estimation by minimizing the negative log likelihood.

The key limitation of parametric methods is the unreliable specification of functional forms. Although Platt [8] is very popular in calibrating applications for its simplicity, [10], [14], [21] found that this form rarely fit the true distribution of scores. For this method fitting errors for samples that are nearby the separating hyperplane are surprisingly large. It must be emphasized that the mis-specification error cannot be alleviated by increasing the training size N.

Both the diversity of the distribution of $(\mathbf{X}, Y)$ and the possible complex function $s(\cdot)$, e.g. the kernel form of SVM, will make the relationship between $S = s(\mathbf{X})$ and $\mathbb{P}\{Y = 1 | S = s\}$ very complicated. This calls for a calibration model that is universally flexible and free from mis-specification.

B. Nonparametric methods

1) Spermic - HistBin: The spermic nonparametric model for calibration is histogram binning (HistBin), also known as quantile binning, which is proposed by Zadrozny and Elkan [9] for NB classifier. In this approach, first N classification scores are sorted in ascending order, then they are partitioned into B subsets of equal frequency, called bins. Given a test sample $\mathbf{x}$, this approach seeks the bin that contains $s(\mathbf{x})$ and returns $\mathbb{P}\{Y = 1 | S = s(\mathbf{x})\}$ as the fraction of positive samples in this bin. This calibrating function is piecewise constant.

HistBin has the following limitations. First, it may violate the requirement of monotonicity. The possibility of violating this requirement increases with B, because the decreasing number of samples in each bin makes the fraction of positive samples more vulnerable to noises. Second, the calibrating function is a piecewise constant and has only B output values. For a given training size N, larger B leads to a larger possibility of violating the requirement of monotonicity, while smaller B implies coarser calibration and more inaccurate probability prediction. Third, because of fixed partition, probability predictions for scores that lie on the borders of each bin are less accurate.

2) Monotone extensions - IsoReg and NearIso: IsoReg [10] proposes the first monotone extension, in which the estimation is the following optimization

$$\min_{\mathbf{p} \in \mathbb{R}^N} \frac{1}{N} \sum_{n=1}^N [p_n - y_n]^2 \quad s.t. \quad p_1 \leq \cdots \leq p_N . \tag{9}$$

There is a well-known method, named pair adjacent violators (PAV) algorithm [22], for computing isotonic regression in statistics. In this algorithm, first assign all positive samples with probability 1 and all other samples with probability 0; second recursively replace a pair-adjacent violator with their average. If $f(s_n) > f(s_{n+1})$ (pair-adjacent violator), update both with their average. The above replacement is executed recursively until there is no pair adjacent violator.

Another closely related method is NearIso [11], [12], which relaxes the monotonicity-related constraint $p_1 \leq \cdots \leq p_N$ with a soft penalty:

$$\min_{\mathbf{p}\in\mathbb{R}^N} \frac{1}{N}\sum_{n=1}^{N}[p_n - y_n]^2 + \lambda \sum_{n=1}^{N-1}(p_n - p_{n+1})\mathbb{I}_{\{p_n>p_{n+1}\}} \tag{10}$$

where $\lambda \in \mathbb{R}_+$ is a parameter for the tradeoff between good-of-fitness and monotonicity. When $\lambda = +\infty$, NearIso is equivalent to IsoReg.

Both IsoReg and NearIso can be regarded as a special binning algorithm where B and bin sizes are selected automatically. All scores that fall in the same bin obtain the same probability prediction. The calibrating function of IsoReg is stepwise constant, while that of NearIso is piecewise constant, because NearIso imposes only a soft penalty term.

3) Continuous extension - LiTE: A continuous extension is LiTE [12], [13], which models the calibration with the ℓ_1 (linear) trend filtering signal approximation [23]. The estimated continuous calibrating function is piecewise linear, instead of piecewise constant. This method increases the smoothness of the calibrating function by imposing a penalty on the number of slopes changes

$$\min_{\mathbf{p}\in\mathbb{R}^N} \frac{1}{N}\sum_{n=1}^{N}[p_n - y_n]^2 \quad s.t. \quad \|\mathbf{v}\|_0 < B - 1 \tag{11}$$

where $\|\cdot\|_0$ is ℓ_0 norm, $\|\mathbf{v}\|_0 = \sum_i \mathbb{I}_{\{v_i \neq 0\}}$, the vector $\mathbf{v} \in \mathbb{R}^{N-1}$ is defined as the second order finite difference vector $v_i = \left|\frac{p_{i+2}-p_{i+1}}{s_{i+2}-s_{i+1}} - \frac{p_{i+1}-p_i}{s_{i+1}-s_i}\right|$, and $B \in \mathbb{N}$ is the maximum number of pieces. To make the problem convex, this model relaxes this penalty term from ℓ_0-norm to ℓ_1-norm using the sparsity property of the ℓ_1 norm

$$\min_{\mathbf{p}\in\mathbb{R}^N} \frac{1}{N}\sum_{n=1}^{N}[p_n - y_n]^2 + \lambda\|\mathbf{v}\|_1. \tag{12}$$

Though LiTE can obtain a piecewise linear calibrating function, LiTE cannot guarantee: (1) $\mathbf{p}^*$ are monotone; (2) $\forall n$, $p_n^* \in [0, 1]$.

4) Adaptive extension - ACP: An adaptive extension of [9] is Adaptive Calibration of Predictions (ACP) [14]. ACP requires the foregoing classifier produce two values for each test sample $\mathbf{x}$: score $s(\mathbf{x}) \in [0, 1]$ and 95% confidence interval $\text{CI}(\mathbf{x}) \subset [0, 1]$. The probability prediction for $s(\mathbf{x})$ is the fraction of class 1 among all samples $s(\mathbf{x}_n) \in \text{CI}(\mathbf{x})$. This method can be regarded as a K-nearest neighbors algorithm, but K is adaptive for each s. The calibrating function estimated from ACP is not monotone. Moreover, this method cannot calibrate classifiers other than logistic regression, because they cannot produce confidence intervals.

5) Monotone and smooth extension - SmoIsoReg: A smooth monotone calibrating function can be obtained by splines regression

$$\min_{f\in\mathcal{F}} \quad \frac{1}{N}\sum_{n=1}^{N}[f(s_n) - y_n]^2 + \lambda \int_{\underline{s}}^{\overline{s}} f^{(m)}(s)\mathrm{d}s. \tag{13}$$

where $\underline{s} := \inf s(\mathscr{X})$, $\overline{s} := \sup s(\mathscr{X})$, $f^{(m)}$ is the order-m derivative of f and $\mathcal{F}$ is the set of monotone functions in $L_2[\underline{s}, \overline{s}]$. If $m = 2$, the optimal solution f^* is a cubic splines. However, due to the difficult in determining the location of additional two knots, this cubic splines regression problem is ill-posed and has no tractable algorithm [24].

In [7] a smooth monotone regression method, abbreviated as SmoIsoReg, is proposed for simplifying problem (13). Different from monotone splines regression, monotone splines interpolation is tractable. SmoIsoReg consists of two steps. First, an IsoReg model is trained with $\{(s_n, y_n)\}_{n=1}^N$. Second, $f(\cdot)$ is estimated by applying a smooth monotone interpolation algorithm, e.g. PCHIP [25], to the set of points $\{s_{[b]}, p_{[b]}\}_{b=1}^B$, where $s_{[b]}$ and $p_{[b]}$ are the representative point and the probability of b-th bin obtained by IsoReg. Simulation results in [24] shows that the loss of optimality from true monotone splines regression to monotone splines interpolation is about 30%.

6) Ensemble-based models - BBQ, ENIR and ELiTE: BBQ [15], [16] is an ensemble of multiple HistBin models with different Bs. ENIR [11], [12] is an ensemble of multiple NearIso models with different λs. Elite [12], [13] is an ensemble of multiple LiTE models with different λs. All above ensemble-based models consist of two steps. First, estimate several individual calibrating functions independently. Second, estimate the final function with the weighted sum of these individual functions. The weights are based on the performance of individual models measured the Bayesian score derived from the BDeu score [26].

C. Comments and challenge

Existing calibration methods and our proposed method are summarized in Table I. All the existing methods failed to achieve four criteria: universal flexibility, monotonicity, continuousness and computational tractability. This situation results mainly from the continuously-constrained requirement of monotonicity, which is imposed on every point of $s(\mathscr{X})$. The existing methods have to compromise between these four criteria.

As shown by many efforts in mathematical statistics, incorporating continuousness into isotonic regression is much involved. With moving average, [27] seemingly significantly increases the smoothness of fitted function by increasing the number of discontinuous points, but it is still a stepwise constant function. In [24] this problem becomes smoothing splines regression, but its computation is intractable for the search of the location of additional two unknown knots. [28] and [29] impose the monotone constraint at only finitely many points, which cannot guarantee the monotonicity over the entire range. [30] proposes a Bernstein polynomial based monotone regression, which satisfies both smoothness and monotonicity. Nevertheless, it is only a sufficient, but not necessary, implementation, i.e. it is over-restricted for a given polynomial degree. [31] transforms the stepwise constant function obtained from PAV to a piecewise linear function by linear interpolation between discontinuous segments. This kind of remedy makes the estimated curve rather rough.

III. Methodology

Without loss of generality, all scores are assumed to be in $[\underline{s}, \overline{s}]$, where $\underline{s} := \inf \mathcal{S}$, $\overline{s} := \sup \mathcal{S}$, $\mathcal{S} := s(\mathscr{X})$. In applications, we can let $\underline{s} := \min s(\{\mathbf{x}_1, \cdots, \mathbf{x}_N\})$ and $\overline{s} := \max s(\{\mathbf{x}_1, \cdots, \mathbf{x}_N\})$. For a test score $s(\mathbf{x})$ that is beyond $[\underline{s}, \overline{s}]$, if $s(\mathbf{x}) < \underline{s}$, let $s(\mathbf{x}) = \underline{s}$; if $s(\mathbf{x}) > \overline{s}$, let $s(\mathbf{x}) = \overline{s}$.

Different from IsoReg [10], NearIso [11], [12] and LiTE [12], [13] that only estimate calibrated probabilities at $\{s_n\}_{n=1}^N$, this paper estimates $f(\cdot)$ by nonparametric regression with random design

$$\min_{f \in \mathcal{F}} \quad \frac{1}{N} \sum_{n=1}^{N} [f(s_n) - y_n]^2 \tag{14}$$

where $\mathcal{F}$ be the family of continuous calibrating functions

$$\mathcal{F} := \left\{ f \in \mathcal{C}[\underline{s}, \overline{s}] \;\middle|\; \begin{array}{l} f(\underline{s}) \geq 0, \quad f(\overline{s}) \leq 1 \\ f(s) \text{ is non-decreasing over } [\underline{s}, \overline{s}] \end{array} \right\}. \tag{15}$$

The main challenge of the optimization problem (14) comes from the requirement of monotonicity, because it should be satisfied by every point in $[\underline{s}, \overline{s}]$. It consists of an infinite number of inequality constraints, which make this problem very hard to solve. In [32] the requirement is replaced with $f(s_1) \leq \cdots \leq f(s_N)$. In [28] the requirement is replaced with the monotonicity of function values at equidistant grid points. However, monotone function values at finite points do not guarantee that the function is monotone over the entire range. Worse than this, semi-infinite optimization theory [17], [18] shows that the function cannot be guaranteed to be non-decreasing over the whole range, even when the monotonicity is satisfied by infinitely many points.

This estimation problem (14) can be regarded as a shape-restricted nonparametric regression that has a long history in statistical literature with seminal works dating back half a century, e.g. [33] and [34]. Common shapes analyzed in nonparametric regression are monotone [35] and convex [36], [37]. This area has found successful applications in many areas, including econometrics [38] and change detection [39].

This paper proposes to solve (14) with shape-restricted polynomial regression, in which f has the following form

$$f(s) = a_0 + a_1 s + \cdots + a_k s^k = \sum_{\ell=0}^{k} a_\ell s^\ell. \tag{16}$$

The family $\mathcal{F}$ is replaced with a set of restricted degree-k polynomials

$$\mathcal{F}_N := \left\{ \begin{array}{l} P_k : [\underline{s}, \overline{s}] \to \mathbb{R} \\ P_k(s) = \displaystyle\sum_{\ell=0}^{k} a_\ell s^\ell \end{array} \;\middle|\; \begin{array}{l} \displaystyle\sum_{\ell=0}^{k} a_\ell \underline{s}^\ell \geq 0, \ \sum_{\ell=0}^{k} a_\ell \overline{s}^\ell \leq 1 \\ \displaystyle\sum_{\ell=1}^{k} \ell a_\ell s^{\ell-1} \geq 0, \forall s \in [\underline{s}, \overline{s}] \\ \displaystyle\sum_{\ell=0}^{k} |a_\ell| \leq \lambda \end{array} \right\}. \tag{17}$$

The requirement of monotonicity becomes $\sum_{\ell=1}^{k} a_\ell \ell s^{\ell-1} \geq 0$, $\forall s \in [\underline{s}, \overline{s}]$, which arises from the differentiability of $f(s) = \sum_{\ell=0}^{k} a_\ell s^\ell$. Here we use the ℓ_1-norm regularization of $\mathbf{a}$ for overcoming over-fitting.

Thus the calibration problem becomes

$$\min_{\mathbf{a} \in \mathbb{R}^{k+1}} \quad \frac{1}{N} \sum_{n=1}^{N} \left[\sum_{\ell=0}^{k} a_\ell s_n^\ell - y_n \right]^2 \tag{18a}$$

$$s.t. \quad \sum_{\ell=0}^{k} a_\ell \underline{s}^\ell \geq 0, \quad \sum_{\ell=0}^{k} a_\ell \overline{s}^\ell \leq 1 \tag{18b}$$

$$\sum_{\ell=1}^{k} a_\ell \ell s^{\ell-1} \geq 0, \quad \forall s \in [\underline{s}, \overline{s}] \tag{18c}$$

$$\sum_{\ell=0}^{k} |a_\ell| \leq \lambda. \tag{18d}$$

Because $\sum_{\ell=1}^{k} a_\ell \ell s^{\ell-1} \geq 0$ is continuously constrained over $[\underline{s}, \overline{s}]$, problem (18) is a semi-infinite program with $k+1$ decision variables and infinitely many inequality constraints. Semi-infinite programs are well-known computationally intractable in numerical optimization. Generally, they can only resort to heuristic algorithms, such as discretization-based methods. These methods replace these infinitely many constraints with finitely many constraints, but fail to guarantee the feasibility of their approximated solutions.

Fortunately, thanks to the following lemma [40, Theorem 9, Theorem 10], this requirement of monotonicity (18c) has an equivalent semidefinite representation. Therefore, this semi-infinite problem (18) can be transformed to a semidefinite program with finite decision variables and finite inequality constraints. Let $\mathbf{H}_{n,\ell}$ be the $n \times n$ Hankel matrix with row-i column-j element

$$H_{n,\ell}^{ij} := \begin{cases} 1, & i + j = \ell \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

Lemma 1. *Consider the polynomial $p(s) = a_0 + a_1 s + \cdots + a_k s^k$, $s \in [\underline{s}, \overline{s}]$.*

(1) When k is even, e.g. $k = 2k_1$, $k_1 \in \mathbb{N}$, $p(s)$ is nonnegative on the closed interval $[\underline{s}, \overline{s}]$, if and only if there exist positive semidefinite real symmetric matrices $\mathbf{U} \in \mathbb{R}^{(k_1+1)\times(k_1+1)}$ and $\mathbf{V} \in \mathbb{R}^{k_1 \times k_1}$ satisfying

$$\begin{aligned} a_\ell = \langle \mathbf{H}_{k_1+1,\ell+2}, \mathbf{U} \rangle + \langle -\underline{s}\overline{s}\mathbf{H}_{k_1,\ell+2}\mathbb{I}_{\{\ell \leq 2k_1-2\}} \\ + (\underline{s} + \overline{s})\mathbf{H}_{k_1,\ell+1}\mathbb{I}_{\{1 \leq \ell \leq 2k_1-1\}} - \mathbf{H}_{k_1,\ell}\mathbb{I}_{\{\ell \geq 2\}}, \mathbf{V} \rangle \end{aligned} \tag{20}$$

for all $\ell = 0, \cdots, 2k_1$.

(2) When k is odd, $k = 2k_1 - 1$, $k_1 \in \mathbb{N}$, $p(t)$ is nonnegative on $[\underline{s}, \overline{s}]$, if and only if there exist positive semidefinite real symmetric matrices $\mathbf{U} \in \mathbb{R}^{k_1 \times k_1}$ and $\mathbf{V} \in \mathbb{R}^{k_1 \times k_1}$ satisfying

$$\begin{aligned} a_\ell = & \langle -\underline{s}\mathbf{H}_{k_1,\ell+2}\mathbb{I}_{\{\ell \leq 2k_1-2\}} + \mathbf{H}_{k_1,\ell+1}\mathbb{I}_{\{\ell \geq 1\}}, \mathbf{U} \rangle \\ & + \langle \overline{s}\mathbf{H}_{k_1,\ell+2}\mathbb{I}_{\{\ell \leq 2k_1-2\}} - \mathbf{H}_{k_1,\ell+1}\mathbb{I}_{\{\ell \geq 1\}}, \mathbf{V} \rangle \end{aligned} \tag{21}$$

for all $\ell = 0, \cdots, 2k_1 - 1$.

With a straightforward application of Lemma 1, problem (18) can be transformed one of the following two programs according to the parity of k.

In case of even k. Let $k_1 = k/2$. The coefficients $\mathbf{a}$ can be obtained by solving the following semidefinite program

$$\min_{\mathbf{a},\mathbf{U},\mathbf{V}} \sum_{n=1}^{N} \left[\sum_{\ell=0}^{k} a_\ell s_n^\ell - y_n \right]^2 \tag{22a}$$

$$s.t. \sum_{\ell=0}^{k} a_\ell \underline{s}^\ell \geq 0, \quad \sum_{\ell=0}^{k} a_\ell \overline{s}^\ell \leq 1 \tag{22b}$$

$$\begin{aligned} \ell a_\ell = & \langle -\underline{s}\mathbf{H}_{k_1,\ell+1}\mathbb{I}_{\{\ell \leq 2k_1-1\}} + \mathbf{H}_{k_1,\ell}\mathbb{I}_{\{\ell\geq 2\}}, \mathbf{U}\rangle \\ & + \langle \overline{s}\mathbf{H}_{k_1,\ell+1}\mathbb{I}_{\{\ell \leq 2k_1-1\}} - \mathbf{H}_{k_1,\ell}\mathbb{I}_{\{\ell\geq 2\}}, \mathbf{V}\rangle \\ & \forall \ell = 1, \cdots, 2k_1 \end{aligned} \tag{22c}$$

$$\sum_{\ell=0}^{k} |a_\ell| \leq \lambda \tag{22d}$$

$$\mathbf{a} \in \mathbb{R}^{2k_1+1}, \mathbf{U}, \mathbf{V} \in \mathbb{R}_+^{k_1 \times k_1}. \tag{22e}$$

In case of odd k. Let $k_1 = (k+1)/2$. The coefficients $\mathbf{a}$ can be obtained by solving the following semidefinite program

$$\min_{\mathbf{a},\mathbf{U},\mathbf{V}} \sum_{n=1}^{N} \left[\sum_{\ell=0}^{k} a_\ell s_n^\ell - y_n \right]^2 \tag{23a}$$

$$s.t. \sum_{\ell=0}^{k} a_\ell \underline{s}^\ell \geq 0, \quad \sum_{\ell=0}^{k} a_\ell \overline{s}^\ell \leq 1 \tag{23b}$$

$$\begin{aligned} \ell a_\ell = & \langle \mathbf{H}_{k_1+1,\ell+1}, \mathbf{U}\rangle + \langle -\underline{s}\overline{s}\mathbf{H}_{k_1,\ell+1}\mathbb{I}_{\{\ell \leq 2k_1-3\}} \\ & + (\underline{s}+\overline{s})\mathbf{H}_{k_1,\ell}\mathbb{I}_{\{1\leq\ell\leq 2k_1-2\}} - \mathbf{H}_{k_1,\ell-1}\mathbb{I}_{\{\ell\geq 2\}}, \mathbf{V}\rangle \\ & \forall \ell = 1, \cdots, 2k_1 - 1 \end{aligned} \tag{23c}$$

$$\sum_{\ell=0}^{k} |a_\ell| \leq \lambda \tag{23d}$$

$$\mathbf{a} \in \mathbb{R}^{2k_1}, \mathbf{U} \in \mathbb{R}_+^{k_1 \times k_1}, \mathbf{V} \in \mathbb{R}_+^{(k_1-1)\times(k_1-1)}. \tag{23e}$$

If the optimal solution is $(\hat{\mathbf{a}}, \hat{\mathbf{U}}, \hat{\mathbf{V}})$, the estimated calibrating function is

$$\hat{f}_N(s) = \sum_{\ell=0}^{k} \hat{a}_\ell s^\ell, \tag{24}$$

and the probability prediction for a test sample $\mathbf{x}$ is

$$\hat{\mathbb{P}}\{Y = 1 | \mathbf{X} = \mathbf{x}] = \sum_{\ell=0}^{k} \hat{a}_\ell s(\mathbf{x})^\ell. \tag{25}$$

The calibration problem (18) has exactly one optimal solution for the following two reasons. First, the feasible set of (18) is closed and convex, because it is the intersection of an infinite number of closed half-spaces. Second, the objective function (18a) is strictly convex with respect to decision variables $\mathbf{a}$.

Semidefinite programming is a generic convex optimization that be efficiently solved by off-the-shelf toolboxes, e.g. CVX [19]. The worst case number of iterations required to solve SDP (22) or (23) to a given accuracy grows with problem size as $O(N^{1/2})$. In practice the algorithms behave very similarly and much better than predicted by the worst-case analysis. It has been observed by many researchers that the number of iterations required grows much more slowly than $N^{1/2}$, and can often be assumed to be almost constant (see [41, §6.4.4] or [42] for comments on the average behavior). For a wide variety of problems and a large range of problem sizes, the methods described above typically require between 5 and 50 iterations. Each iteration can be solved in $O(N^2)$ operations using direct methods. Therefore, regularly the computational cost of this estimation is as cheap as $O(N^2)$.

IV. Theoretical analysis

The first subsection verifies the universal flexibility of the proposed method, and the second subsection proves the weak and strong convergence of the proposed estimator.

A. Universal flexibility

Let $\mathcal{P}_k$ be the set of restricted algebraic polynomials with degree $\leq k$

$$\mathcal{P}_k := \left\{ \begin{array}{l|l} P_k : [\underline{s}, \overline{s}] \to \mathbb{R} & \displaystyle\sum_{\ell=0}^{k} a_\ell \underline{s}^\ell \geq 0, \ \sum_{\ell=0}^{k} a_\ell \overline{s}^\ell \leq 1 \\ P_k(s) = \displaystyle\sum_{\ell=0}^{k} a_\ell s^\ell & \displaystyle\sum_{\ell=1}^{k} \ell a_\ell s^{\ell-1} \geq 0, \forall s \in [\underline{s}, \overline{s}] \end{array} \right\}. \tag{26}$$

If $P_k(t) = \sum_{\ell=0}^{k} a_\ell s^\ell \in \mathcal{P}_k$, $\sum_{\ell=0}^{k} a_\ell s^\ell + 0 \times s^{k+1} \in \mathcal{P}_{k+1}$. Hence, the sequence of function sets $\mathcal{P}_1, \mathcal{P}_2, \cdots$, is nested in $\mathcal{F}$, i.e.

$$\mathcal{P}_1 \subset \mathcal{P}_2 \subset \mathcal{P}_3 \subset \cdots \subset \mathcal{F} \subset \mathcal{C}[\underline{s}, \overline{s}] \subset L_2[\underline{s}, \overline{s}]. \tag{27}$$

To verify the universal flexibility of $\cup_{k=1}^{\infty} \mathcal{P}_k$, we must show that for any calibrating function $f \in \mathcal{F}$ and any arbitrary error $\epsilon > 0$, there exist one $k \in \mathbb{N}$ and a degree-k monotone polynomial $P_k \in \mathcal{P}_k$ such that

$$\|f - P_k\|_\infty := \sup_{s \in [\underline{s}, \overline{s}]} |f(s) - P_k(s)| < \epsilon. \tag{28}$$

To obtain this, one cannot resort to the Weierstrass approximation theorem, which states that every continuous function defined on a closed interval can be uniformly approximated as closely as desired by a polynomial function. Even though the approximating function provides a nice approximation, there is still a possibility of violating the requirement of monotonicity at some points of the range.

The universal flexibility is equivalent to the denseness of $\cup_{k=1}^{\infty} \mathcal{P}_k$ in $\mathcal{F}$ with respect to the uniform norm on $[\underline{s}, \overline{s}]$. Before presenting the theorem on denseness, we introduce the definition of the modulus of smoothness [43]. The modulus of smoothness of order n of a function $f \in \mathcal{C}[\underline{s}, \overline{s}]$ is the function ω_n defined as

$$\omega_n(f, h) := \sup_{s \in [\underline{s}, \overline{s} - nh]} |\Delta_h^n(f, s)| \text{ for } 0 \leq h \leq \frac{\overline{s} - \underline{s}}{n}, \tag{29}$$

and

$$\omega_n(f, h) := \omega_n\left(f, \frac{\overline{s} - \underline{s}}{n}\right), \text{ for } h > \frac{\overline{s} - \underline{s}}{n}, \tag{30}$$

where $\Delta_h^n(f, s_0)$ is the finite difference (n-th order forward difference) defined as

$$\Delta_h^n(f, s_0) := \sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} f(s_0 + ih). \tag{31}$$

For example, when $n = 2$,

$$\Delta_h^2(f, s_0) := f(s_0 + 2h) - 2f(s_0 + h) + f(s_0). \tag{32}$$

Theorem 2. *$\cup_{k=1}^{\infty} \mathcal{P}_k$ is dense in $\mathcal{F}$ with respect to sup-norm.*

Proof of Theorem 2. This theorem means that, for any $f \in \mathcal{F}$,

$$\lim_{k\to\infty} \min_{P_k \in \mathcal{P}_k} \|f - P_k\|_\infty = 0. \tag{33}$$

This proof is mainly built on [44, Theorem 3.5], which analyzes the approximation error of monotone polynomials for a monotone function in the uniform norm. This theorem shows that for any non-decreasing function $g \in \mathcal{C}[-1, 1]$ there is an algebraic polynomial G_k with degree $\leq k$ that is non-decreasing on [-1,1] and satisfies

$$\|g - G_k\|_\infty \leq C_g \omega_2(g, 1/k) \tag{34}$$

where C_g is an absolute constant. For any $f \in \mathcal{F}$, if

$$g(z) := f\left(\underline{s} + \frac{z+1}{2}(\overline{s} - \underline{s})\right), \tag{35}$$

$g : [-1, 1] \to [0, 1]$ is a non-decreasing function. According to the above theorem, there is a non-decreasing polynomial $G_k : [-1, 1] \to \mathbb{R}$ with degree $\leq k$ that satisfies (34).

(1) When $\{G_k(z) : z \in [-1, 1]\} \subset [0, 1]$, if we let

$$\tilde{P}_k(s) := G_k\left(-1 + 2\frac{s - \underline{s}}{\overline{s} - \underline{s}}\right), \tag{36}$$

we have $\tilde{P}_k \in \mathcal{P}_k$. Therefore,

$$|f(s) - \tilde{P}_k(s)| = |g(z) - G_k(z)| \leq C_g \omega_2(g, 1/k). \tag{37}$$

(2) When $\{G_k(z) : z \in [-1, 1]\} \not\subset [0, 1]$, $\underline{G} := G_k(-1) < 0$ or $\overline{G} := G(1) > 1$. So $\tilde{P}_k$ obtained from (36) doesn't satisfy the requirement of boundary, i.e. $\exists s \in [\underline{s}, \overline{s}]$, $\tilde{P}_k(s) \notin [0, 1]$, and thus $\tilde{P}_k \notin \mathcal{P}_k$. However, by scaling into [0,1] the following function

$$\tilde{P}(s) := \frac{G_k\left(-1 + 2\frac{s-\underline{s}}{\overline{s}-\underline{s}}\right) - \underline{G}}{\overline{G} - \underline{G}} \tag{38}$$

satisfies $\tilde{P}_k \in \mathcal{P}_k$. Moreover,

$$\begin{aligned}
&|f(s) - \tilde{P}(s)| \\
=&\left|g(z) - \frac{G_k(z) - \underline{G}}{\overline{G} - \underline{G}}\right| \\
=&\frac{|g(z)\overline{G} - G_k(z)\overline{G} + G_k(z)\overline{G} - G_k(z) - g(z)\underline{G} + \underline{G}|}{|\overline{G} - \underline{G}|} \\
\leq&\frac{|g(z) - G_k(z)||\overline{G}| + |G_k(z)||\overline{G} - 1| + |\underline{G}||g(z) - 1|}{|\overline{G} - \underline{G}|} \\
\leq&\frac{C_g \omega_2(g, 1/k)(3 + 2C_g \omega_2(g, 1/k))}{1 - 2C_g \omega_2(g, 1/k)}.
\end{aligned} \tag{39}$$

The last inequality follows from the fact

$$|\underline{G}| \leq C_g \omega_2(g, 1/k), \qquad |\overline{G} - 1| \leq C_g \omega_2(g, 1/k). \tag{40}$$

Summarizing (37) and (39), we have

$$|f(s) - \tilde{P}(s)| \leq C_g \omega_2(g, 1/k) \left\{1, \frac{3 + 2C_g \omega_2(g, 1/k)}{1 - 2C_g \omega_2(g, 1/k)}\right\}. \tag{41}$$

Because the inequality holds for each $s \in [\underline{s}, \overline{s}]$,

$$\|f - \tilde{P}\|_\infty \leq C_g \omega_2(g, 1/k) \left\{1, \frac{3 + 2C_g \omega_2(g, 1/k)}{1 - 2C_g \omega_2(g, 1/k)}\right\}. \tag{42}$$

From the property of modulus of smoothness $\lim_{h\to 0^+} \omega_2(g, h) = 0$, we obtain

$$\lim_{k\to\infty} C_g \omega_2(g, 1/k) \left\{1, \frac{3 + 2C_g \omega_2(g, 1/k)}{1 - 2C_g \omega_2(g, 1/k)}\right\} = 0. \tag{43}$$

Since $\min_{P_k \in \mathcal{P}_k} \|d - P_k\|_\infty \leq \|d - \tilde{P}_k\|_\infty$, we obtain Eq.(33) and the denseness of $\cup_{k=1}^{\infty} \mathcal{P}_k$ in $\mathcal{F}$. □

Let the two parameters of $\mathcal{F}_N$ (17) vary with the data size N, and rewrite k as k_N and λ and λ_N.

Theorem 3. *If $k_N \uparrow \infty$ and $\lambda_N \uparrow \infty$, $\cup_{N=1}^{\infty} \mathcal{F}_N$ is dense in $\mathcal{F}$ with respect to sup-norm.*

Proof of Theorem 3. First we note that

$$\mathcal{F}_N = \mathcal{P}_{k_N} \cap \left\{\sum_{\ell=0}^{k_N} a_\ell s^\ell : \sum_{\ell=0}^{k_N} |a_\ell| \leq \lambda_N\right\}. \tag{44}$$

According to (27), if $k_N \uparrow \infty$ and $\lambda_N \uparrow \infty$,

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \cdots \subset \mathcal{F} \subset \mathcal{C}[\underline{s}, \overline{s}] \subset L^2[\underline{s}, \overline{s}]. \tag{45}$$

By Theorem 2, $\cup_{N=1}^{\infty} \mathcal{F}_N$ is dense in $\mathcal{F}$ with respect to sup-norm. □

B. Universal Consistency

In this paper the calibration problem is solved by nonparametric regression estimation with random design, which means that all s_ns are random variables. Assume (S, Y), (s_1, y_1), (s_2, y_2), $\cdots$ are independent and identically distributed (i.i.d.) random variables. Let $\mathcal{D}_N$ be the set of data defined by $\mathcal{D}_N = \{(s_1, y_1), \cdots, (s_N, y_N)\}$. In the proposed model (24) one uses the data $\mathcal{D}_N$ in order to construct an estimate $\hat{f}_N : [\underline{s}, \overline{s}] \to [0, 1]$ of the regression function

$$f(s) = \mathbb{E}(Y = 1|S = s) = \arg\min_g \mathbb{E}|Y - g(S)|^2. \tag{46}$$

Theorem 4. *If $\{k_N\}$ and $\{\lambda_N\}$ satisfy*

$$\lambda_N \uparrow \infty,\ k_N \uparrow \infty,\ k_N/N \to 0 \tag{47}$$

then,

(a) $\{\hat{f}_N\}$ is weakly universally consistent, i.e.

$$\lim_{N\to\infty} \mathbb{E}\left\{\int (f(s) - \hat{f}_N(s))^2 \mu(\mathrm{d}s)\right\} = 0 \tag{48}$$

for all distributions of (S, Y).

(b) $\{\hat{f}_N\}$ is strongly universally consistent, i.e.

$$\lim_{N\to\infty} \int (f(s) - \hat{f}_N(s))^2 \mu(\mathrm{d}s) = 0, \text{ with probability 1} \tag{49}$$

for all distributions of (S, Y).

Proof of Theorem 4. This proof is mainly based on [45, Chapter 9 and 10].

According to [45, Lemma 10.1], we have, for every $f \in \mathcal{F}$

$$\begin{aligned}
&\int |\hat{f}_N(s) - f(s)|^2 \mu(\mathrm{d}s) \\
&\le \inf_{g \in \mathcal{F}_N} \int |g(s) - f(s)|^2 \mu(\mathrm{d}s) \\
&+2 \sup_{g \in \mathcal{F}_N} \left| \frac{1}{N} \sum_{n=1}^{N} |g(s_n) - y_n|^2 - \mathbb{E}\{(g(S) - Y)^2\} \right|. \quad (50)
\end{aligned}$$

In order to get universal consistency of $\{\hat{f}_N\}$ it suffices to show that both terms converge to 0 for all distributions of (S, Y).

(i) This part verifies the convergence of the first term in (50). From Theorem 3, for every $f \in \mathcal{F}$, if $\lambda_N \uparrow \infty$ and $k_N \uparrow \infty$, we have

$$\lim_{N \to \infty} \min_{g \in \mathcal{F}_N} \|f - g\|_\infty = 0, \quad (51)$$

which implies that

$$\lim_{N \to \infty} \inf_{g \in \mathcal{F}_N} \int |g(s) - f(s)|^2 \mu(\mathrm{d}s) = 0 \quad (52)$$

for all probability measure μ with support $[\underline{s}, \overline{s}]$.

(ii) This part verifies the convergence of the second term in (50), which is more technical.

Let $P_N(\epsilon) :=$

$$\mathbb{P}\left\{ \sup_{g \in \mathcal{F}_N} \left| \frac{1}{N} \sum_{n=1}^{N} |g(s_n) - y_n|^2 - \mathbb{E}\{(g(S) - Y)^2\} \right| > \epsilon \right\}$$

and

$$\mathcal{H}_N := \{h : h(s, y) = |g(s) - y|^2, g \in \mathcal{F}_N\}. \quad (53)$$

According to Theorem 9.1 and Lemma 9.2 in [45], we have

$$P_N(\epsilon) \le 8\mathbb{E}\mathcal{M}_1(\epsilon/8, \mathcal{H}_N, \mathcal{D}_N) \cdot \exp\{-N\epsilon^2/128\}, \quad (54)$$

where $\mathcal{M}_1(\epsilon/8, \mathcal{G}_N, \mathcal{D}_N)$ is the L_1 $\epsilon/8$-packing number of $\mathcal{H}_N$ on $\mathcal{D}_N$ ([45, Definition 9.4]).

Because $g(s) \in [0, 1]$ for all $g \in \mathcal{F}_N$, we have $|g(s) - y|^2 \in [0, 1]$ for all $s \in [\underline{s}, \overline{s}], y \in \{0, 1\}, g \in \mathcal{F}_N$. Let $h_i(s, y) = |g_i(s) - y|^2$ for some $g_i \in \mathcal{F}_N$. Then

$$\begin{aligned}
&\frac{1}{N} \sum_{n=1}^{N} |h_1(s_n, y_n) - h_2(s_n, y_n)| \\
=&\frac{1}{N} \sum_{n=1}^{N} |g_1(s_n) - g_2(s_n)| \cdot |g_1(s_n) + g_2(s_n) - 2y_n| \\
\le&2\frac{1}{N} \sum_{n=1}^{N} |g_1(s_n) - g_2(s_n)|. \quad (55)
\end{aligned}$$

Thus, if $\{h_1, \cdots, h_l\}$ is an $\epsilon/8$-packing of $\mathcal{H}_N$ on $\mathcal{D}_N$, then $\{g_1, \cdots, g_l\}$ is an $\epsilon/16$-packing of $\mathcal{F}_N$ on $\{s_n\}_{n=1}^N$. Thus

$$\mathbb{E}\mathcal{M}_1(\epsilon/8, \mathcal{H}_N, \mathcal{D}_N) \le \mathbb{E}\mathcal{M}_1(\epsilon/16, \mathcal{F}_N, \{s_n\}_{n=1}^N). \quad (56)$$

By Theorem 9.4 of [45] we can bound the latter term

$$\mathcal{M}_1(\epsilon/16, \mathcal{F}_N, \{s_n\}_{n=1}^N) \le 3\left(\frac{32e}{\epsilon} \log \frac{48e}{\epsilon}\right)^{V_{\mathcal{F}_N^+}}, \quad (57)$$

where $V_{\mathcal{F}_N^+}$ is the Vapnik-Chervonenkis (VC) dimension of

$$\mathcal{F}_N^+ := \{\{(s, y) : y \le g(s)\} : g \in \mathcal{F}_N\}. \quad (58)$$

Because

$$\begin{aligned}
\mathcal{F}_N^+ &\subseteq \left\{ \{(s, y) : y \le g(s)\} : g(s) = \sum_{\ell=0}^{k_N} a_\ell s^\ell \right\} \\
&\subseteq \left\{ \left\{ (s, y) : \sum_{\ell=0}^{k_N} a_\ell s^\ell + by \ge 0 \right\} : b \in \mathbb{R}, \mathbf{a} \in \mathbb{R}^{k_N} \right\},
\end{aligned}$$

Theorem 9.5 of [45] implies

$$V_{\mathcal{F}_N^+} \le k_N + 1. \quad (59)$$

Therefore

$$\begin{aligned}
P_N(\epsilon) &\le 24\left(\frac{32e}{\epsilon} \log \frac{48e}{\epsilon}\right)^{k_N+1} \exp\left\{-\frac{N\epsilon^2}{128}\right\} && (60) \\
&\le 24\left(\frac{32e}{\epsilon} \frac{48e}{\epsilon}\right)^{k_N+1} \exp\left\{-\frac{N\epsilon^2}{128}\right\} && (61) \\
&\le 24 \exp\left\{2(k_N + 1) \log \frac{48e}{\epsilon} - \frac{N\epsilon^2}{128}\right\} && (62) \\
&= 24 \exp\left\{-N\left(\frac{\epsilon^2}{128} - 2\frac{k_N + 1}{N} \log \frac{48e}{\epsilon}\right)\right\}, && (63)
\end{aligned}$$

where (60) ⇒ (61), we have used $\log(x) \le x - 1 \le x$ $(x > 0)$.

(iii) If $k_N/N \to 0$, we have $\sum_{N=1}^\infty P_N(\epsilon) < \infty$. This, together with the Borel-Cantelli lemma, we obtain

$$\lim_{N \to \infty} \sup_{g \in \mathcal{F}_N} \left| \frac{1}{N} \sum_{n=1}^{N} |g(s_n) - y_n|^2 - \mathbb{E}\{(g(S) - Y)^2\} \right| = 0$$

with probability 1. Since (50) and (52), condition (47) is sufficient for the strong universal consistency of $\{\hat{f}_N\}$.

If ξ is a nonnegative random variable, for an arbitrary $\epsilon > 0$, we have

$$\mathbb{E}(\xi) = \int_0^\infty P\{\xi > x\}\mathrm{d}x \le \epsilon + \int_\epsilon^\infty P\{\xi > x\}\mathrm{d}x. \quad (64)$$

Following this,

$$\begin{aligned}
&\mathbb{E}\left\{ \sup_{g \in \mathcal{F}_N} \left| \frac{1}{N} \sum_{n=1}^{N} |g(s_n) - y_n|^2 - \mathbb{E}\{(g(S) - Y)^2\} \right| \right\} \\
\le&\epsilon + \int_\epsilon^\infty 24\left(\frac{48e}{t}\right)^{2(k_N+1)} \exp\left\{-\frac{Nt^2}{128}\right\}\mathrm{d}t && (65) \\
\le&\epsilon + 24\left(\frac{48e}{\epsilon}\right)^{2(k_N+1)} \int_\epsilon^\infty \exp\left\{-\frac{N\epsilon t}{128}\right\}\mathrm{d}t && (66) \\
=&\epsilon + 24\left(\frac{48e}{\epsilon}\right)^{2(k_N+1)} \frac{128}{N\epsilon} \exp\left\{-\frac{N\epsilon^2}{128}\right\} && (67) \\
=&\epsilon + \frac{3072}{N\epsilon} \exp\left\{-N\left(\frac{\epsilon^2}{128} - 2\frac{k_N + 1}{N} \log \frac{48e}{\epsilon}\right)\right\} && (68) \\
\to&0 \quad (N \to \infty). && (69)
\end{aligned}$$

With $\epsilon \to 0$, together with (50) and (52), this result follows that $\{\hat{f}_N\}$ is universally weakly consistent. □

At the end of this subsection, we must emphasize that, to achieve statistical convergence, one doesn't need to impose any constraint on λ_N except $\lambda_N \uparrow \infty$. This merit stems from the uniform boundedness of $|g(S) - Y|^2$ over $\mathcal{F}_N$.

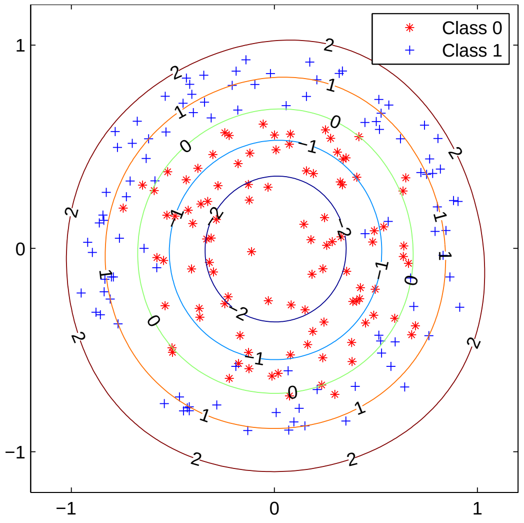


Fig. 1. Toy experiment 1: the data set (in scatter plot) and the scoring function (in contour plot)

V. Experiments

A. Experiment 1: illustrative comparisons

The data, shown in Fig. 1, were generated in the following way:

$$\mathbf{X} = (R\sin\theta, R\cos\theta)', \theta \sim \text{Unif}(0, 2\pi)$$

$$R|Y = 0 \sim \text{Beta}(2, 5), R|Y = +1 \sim \text{Beta}(5, 2)$$

where Unif(a, b) denotes the uniform distribution with support $[a, b]$, and Beta(α, β) is the Beta distribution with shape parameters α and β. This data contain 100 samples from class 0 and 100 samples from class 1. The foregoing classifier is SVM with RBF kernel ($\sigma^2 = 1$). The estimated scoring function $s(\cdot)$ is shown as contour plot in Fig. 1.

Because the true membership probability is never unknown, we must resort to the empirical probability estimated with a very large test data, which are generated in the way same as the training data. In this experiment, the test data have 400,000 samples: 200,000 from class 0 and 200,000 from class 1. Because the test size is very large, empirical probabilities can be regarded as true probabilities for model comparison in this experiment.

For piecewise constant calibrating functions, which are estimated from HistBin, IsoReg, NearIso, and ACP, the empirical probability of $s \in [0, 1]$ is the fraction of class-1 samples among the samples with $s(\mathbf{x}) = s$. Because continuous calibrating functions, which are estimated from Platt, LiTE, SmoIsoReg, and RPR, have infinitely many values, we should use the common discretization method. The 400,000 predicted probabilities are partitioned into 100 equal-frequency bins. For each bin, the predicted probability is the simple average of predicted probabilities in this bin, and the empirical probability is the fraction of class-1 samples among the samples whose probabilities fall in this bin.

In this experiment, we consider all eight individual calibration methods listed in Table I. In HistBin, $B = 25$. In NearIso, $\lambda = 2$. In LiTE, $\lambda = 2$. In RPR, $k = 16$ and $\lambda = 10^3$. Because SVM cannot output a confidence interval for each sample that ACP needs, ACP predicts the membership probability of $\mathbf{x}$ to be the fraction of positive samples among the training samples whose scores $s(\mathbf{x}_n)$ are $N \times 5\%$ most closest to $s(\mathbf{x})$. The above hyperparameter settings are arbitrary just for illustrating purpose.

Experimental results are shown in Fig. 2(a)-2(h). In each subfigure, the left panel plots the calibrating function, i.e. $s \to f(s) = \hat{\mathbb{P}}\{Y = 1|S = s\}$, and the right panel is the reliability curve that shows the relation between predicted probability $\hat{\mathbb{P}}\{Y = 1|S = s\}$ and the empirical probability that is regarded as the true probability $\mathbb{P}\{Y = 1|S = s\}$. In general, perfect calibration corresponds to a straight line from (0,0) to (1,1). The closer a calibration curve is to this line, the better calibrated is the associated prediction method.

In terms of monotonicity, HistBin, NearIso, LiTE, and ACP fail to obtain monotone calibrating functions due to the lack of the requirement of monotonicity. In terms of continuousness, the estimated calibrating function from HistBin, IsoReg, NearIso, and ACP are piecewise constant, and that of LiTE is piecewise linear and continuous. Platt, SmoIsoReg, and RPR achieve continuous and monotone calibrating function. Among the eight models, RPR achieves the best calibrating function in terms of both qualitative and quantitive criteria.

For Platt, calibrating errors at about 25% and 75% quantiles are very large, which agrees with many previous publications. Compared with the better calibrating function obtained from RPR, this sigmoid-based calibrating function has a steeper slope in the middle and flatter slope in two tails. The exponential form of the sigmoid function makes it fail to capture fat tails in applications. This mis-specification error cannot be reduced by increasing the training size.

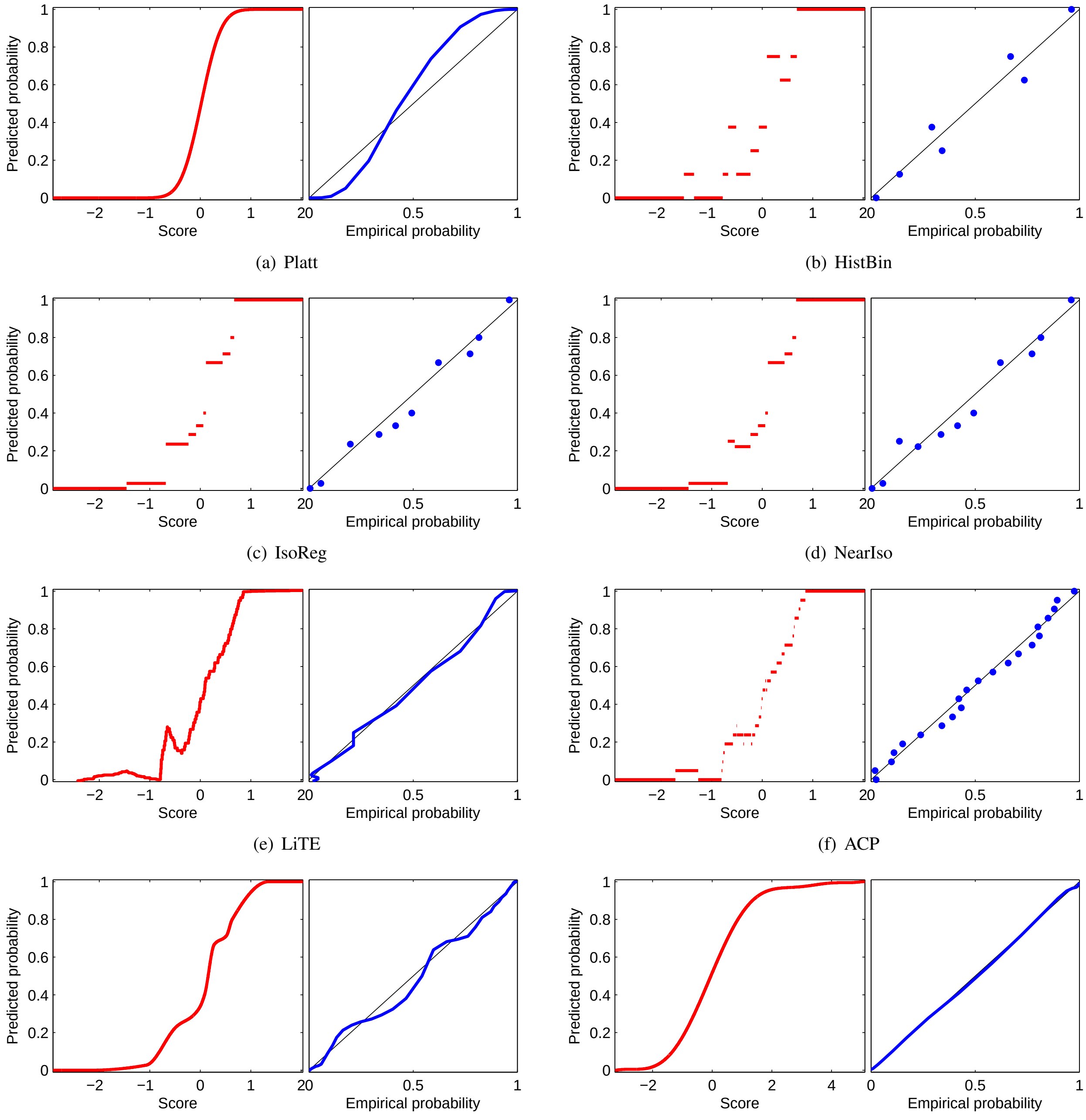
B. Experiment 2: on hyperparameters k and λ

This experiment uses the Adult data from the UCI machine learning repository [46], which has 14 features and 45,222 samples (after removing samples with unknown features). The calibrating objective is to predict the probability of earning more than $50k. The percentage of the positive class in the first data is 24.78%. 80 samples are randomly drawn for SVM training (RBF kernel with $\sigma^2 = 1$) and probability calibration. The other samples are used to construct the empirical calibration function, since its size is very large.

We estimate the calibrating function by RPR with different polynomial degrees: $k = 4, 6, 8, 40$. In these four models, the tradeoff parameter λ is fixed at 100. Fig. 3(a)-3(d) show four estimated calibrating functions and four corresponding reliability curves. It is clear that the estimated calibrating function adapts better with the data as the polynomial degree increases. When k is small, the calibrating function achieves poor out-of-sample performance for the sake of deficient flexibility and under-fitting. However, when k is very large, the calibrating function also achieves undesirable out-of-sample performance for the sake of excessive flexibility and overfitting.

We also study the effect of λ with four values: 1, 10, 100, 1000. In these four models, k is fixed at 16. As shown in Fig. 3(e)-3(h), the estimated calibrating function can achieve more flexibility as the polynomial degree increases. When λ

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2895794, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Platt

(b) HistBin

(c) IsoReg

(d) NearIso

(e) LiTE

(f) ACP

(g) SmoIsoReg

(h) Restricted polynomial regression

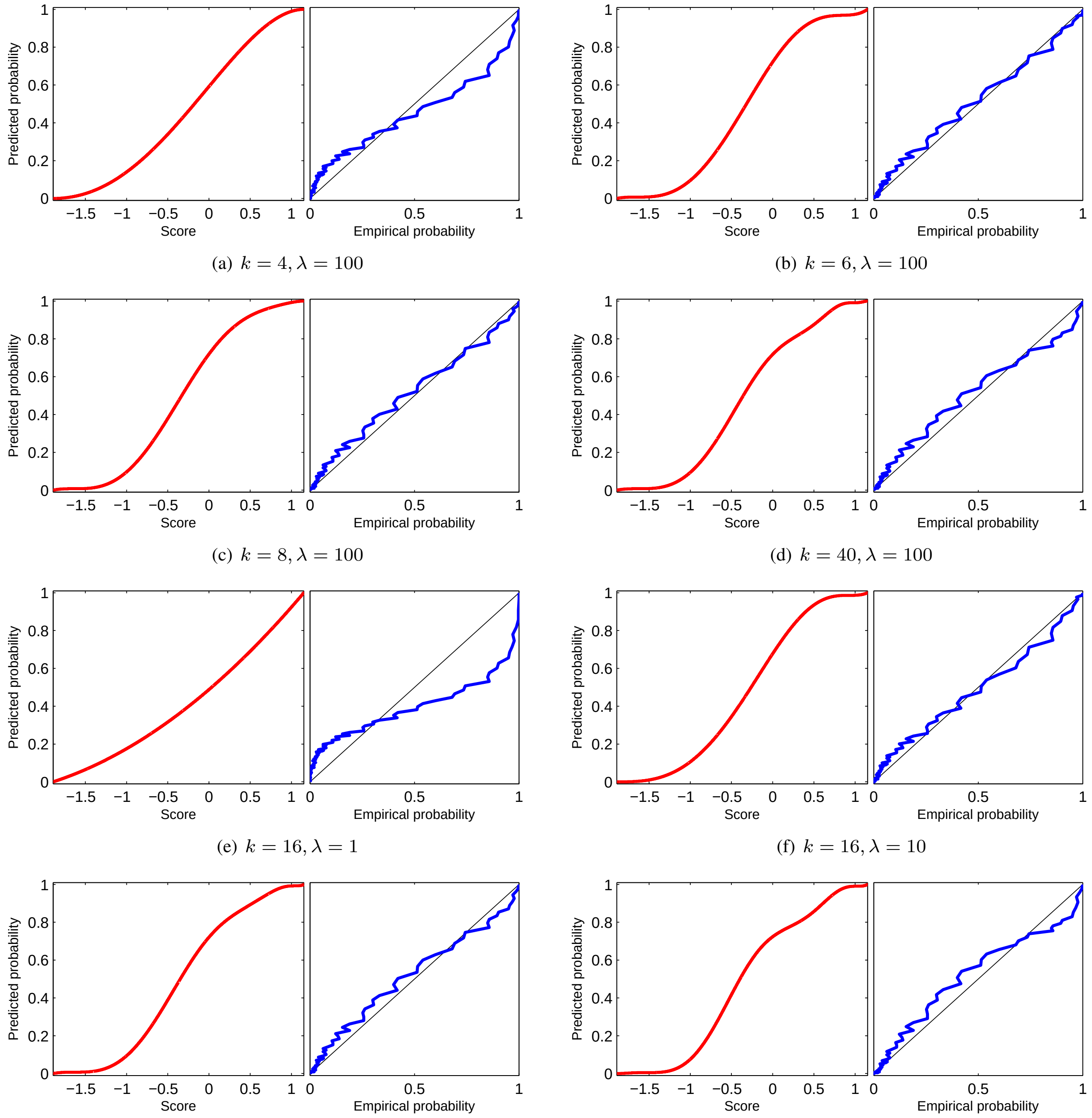
Fig. 2. Experiment 1: an illustrative comparison between various calibration models on a simulated data. In each subfigure, Left panel: the estimated calibrating function; right panel: reliability gram.

is small, this polynomial is too restricted to achieve nice in-the-sample and nice out-of-sample performance. When λ is very large, however, this polynomial seems to be too free and overfit the training data.

Compared with Fig. 2(h), RPR in this experiment provides worse calibrating performance. The calibrating performance depends on two terms: the approximation error (the first term in Eq. (50)) and the estimation error (the second term in Eq. (50)). In this experiment, the small training size (N=80) makes this estimator subject to large estimator error, though it is capable of achieving nice approximation for its great flexibility.

Compared with all reliability grams in Figure 2, these reliability curves in this experiment are rougher. This difference is caused by different test sizes in the test step. When the number of samples in each bin is smaller, the fraction of positive class is subject to larger deviation. In experiment 1 the number of samples in each bin is 4,000, while in experiment 2 the number of samples in each bin is 400.

At the end of this subsection, we must emphasize that the requirement of monotonicity itself plays a similar role in terms of reducing overfitting. Polynomial regression with a high degree is widely criticized for its overfitting. Its estimated curve frequently displays wide fluctuation. The monotone



(a) $k = 4, \lambda = 100$

(b) $k = 6, \lambda = 100$

(c) $k = 8, \lambda = 100$

(d) $k = 40, \lambda = 100$

(e) $k = 16, \lambda = 1$

(f) $k = 16, \lambda = 10$

(g) $k = 16, \lambda = 100$

(h) $k = 16, \lambda = 1000$

Fig. 3. Experiment 2: experimental results of RPR with various k and λ on the Adult data. In each subfigure, Left panel: the estimated calibrating function; right panel: reliability gram.

shape-restriction is a very strong restriction on this polynomial, because the estimated polynomial cannot be wiggly at least.

C. Experiment 3: performance comparison

This subsection compares eleven calibration models listed in Table I with two real data sets from UCI machine learning repository [46]. The first data set is the Adult data. The second data set is Bank Marketing, which has 20 features and 45,211 samples. The calibrating objective is to predict the probability of subscribing term deposits.

Performance comparison is based on two intuitive statistics that measure calibration relative to the ideal reliability diagram: Expected Calibration Error (ECE) and Maximum Calibration Error (MCE). In computing these measures, the predictions are sorted and partitioned into $K = 100$ fixed number of equal-frequency bins. For each bin, the predicted probability and the empirical probability are computed in the way same as that in Subsection V-A. The ECE calculates average calibration error over these bins, and MCE calculates the maximum calibration error among these bins, i.e.

$$\mathrm{ECE} = \sum_{i=1}^{K} |p_i - e_i|/K, \quad \mathrm{MCE} = \max_{i=1,\cdots,K} |p_i - e_i| \qquad (70)$$

where e_i is the empirical (or observed) probability, which is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2895794, IEEE Transactions on Pattern Analysis and Machine Intelligence

regarded as the true fraction of positive instances in i-th bin), and p_i is the predicted probability (the mean of the post-calibrated probabilities for the instances in i-th bin). Lower values of ECE and MCE mean better calibrating performance.

Two foregoing algorithms are used to obtain the scores for estimating calibrating functions: logistic regression and SVM with RBF kernel. The kernel parameter σ^2 is determined by 4-fold cross-validation with choices $\{10^i\}_{i=-10}^{10}$. All ensemble-based methods, i.e. BBQ, ENIR and ELiTE, use the default settings of the toolbox that is available at https://github.com/pakdaman/calibration.git. The number of bins for Histogram Binning, and the tradeoff parameter λ for both NearIso and LiTE are based on the best Bayesian score derived from the BDeu [26]. When logistic regression is applied as the foregoing classifier, predictions of ACP use 95% confidence interval. When SVM is applied, ACP is simplified as K-nearest neighbor prediction with $K = \text{round}(N \times 5\%)$.

The determination of hyperparameters for other methods is based on 2-fold cross validation. For RPR, the choices for the polynomial degree k are $\{4, 5, \cdots, 20\}$, and the choices for λ are $\{5^i\}_{i=-10}^{10}$. For SmoIsoReg, the choices for λ, the trade-off parameter of the smoothness term, are $\{2^i\}_{i=-10}^{10}$. Model evaluation is based on the MCE measure with bin size $K = 10$. After obtaining the best hyperparameters by cross validation, we train these models with the whole training data and these optimal hyperparameters.

Other commonly used performance measures, such as area under ROC curve (AUC) and classification accuracy, will not be used, because these measures evaluate how well the methods discriminate class-0 and class-1 instances. A strictly monotone calibrating function will not alter the ROC curve. Since $\mathbb{P}[Y = 1|\mathbf{X} = \mathbf{x}]$ is hard to estimate, model comparison based on performance measurement associated with the overall probability prediction problem (1) is almost infeasible. Therefore, model comparison in this paper is based on only two reliability-curve related measures.

The whole data are randomly partitioned into training data and test data. To show calibrating performance of these models under different training sizes, we randomly draw 200 or 500 samples as training data and the residual samples as test data. The test size is very large (more than 40,000), which can make the empirical probability in each bin very close to the true probability. We first train foregoing classification models and train calibration models using the same training data, second evaluate calibrating performance with the test data. To reduce statistical variability, the above partition and evaluation are executed 50 rounds. The final model comparison is based on the average of these 50 rounds.

Calibration results with foregoing classifier logistic regression are shown in Table II. Calibration results with foregoing classifier SVM are shown in Table III. The main features are summarized as follows.

1) The proposed RPR achieves dominated advantage over other ten calibration methods. In the experiment with LR scores, RPR achieves 7 best performances, and in the experiment with SVM scores, RPR achieves 5 best performances and 3 second best performances. We believe that this advantage results from the exploit of domain knowledge on calibration functions, i.e. monotonicity and continuousness.
2) On the whole, in most methods calibration performance increases with training size N. Calibration methods that exploit less domain knowledge can achieve greater marginal improvement. This is clearly verified by going through that of HistBin, which exploit no domain knowledge at all.
3) Ensemble-based models can achieve better calibrating performance than their corresponding component models in general.

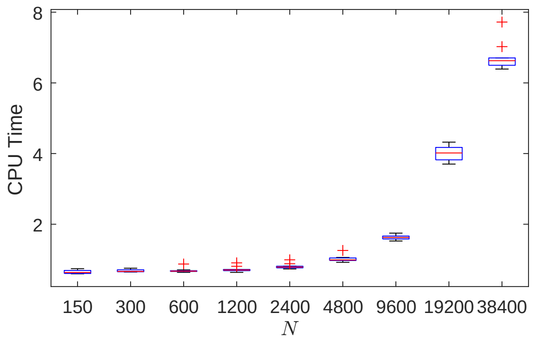


Fig. 4. Experiment 4: Boxplot of computational time of RPR (in seconds).

D. *Experiment 4: on computational time*

This subsection shows the computational time of the proposed method with the Adult data. This experiment is run on a common PC with Intel Core i5-2400 CPU @3.10GHz and 4GB RAM. We consider 9 training sizes $\{150 \times 2^i\}_{i=0}^{8}$. For each N, we draw randomly N samples and record CPU time of RPR. The above procedure was repeated 10 times and their statistics are shown with boxplot in Figure 4. The computational time is within 10 seconds, even when N=38 400, which is sufficiently large for most practical applications. Moreover, in all SDP optimization the numbers of iteration are between 19 and 23. It seems that the number of iterations is independent of the training size N.

VI. Conclusions and future work

This paper proposes a shape-restricted polynomial regression for probability calibration problem. This method has a great advantage over existing methods, because it satisfies all four criteria for calibration methods. Experimental results on both toy and real data sets clearly demonstrate that the proposed method can achieve better calibration performance than other methods. Its advantage is apparent because it exploits domain knowledge on calibrating functions.

Further work includes two directions. One is an extension to multiple classifiers [32], [47], in which probability calibration is based on multiple scoring functions. The question is how to ensemble multiple classification algorithms to achieve better membership probability prediction. The other is an extension from binary-class to multi-class or structured classification problems [48].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2895794, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE II
CALIBRATION RESULTS BASED ON LOGISTIC REGRESSION SCORES

	Adult				Bank Marketing			
	$N = 200$		$N = 500$		$N = 200$		$N = 500$	
	ECE	MCE	ECE	MCE	ECE	MCE	ECE	MCE
Platt	8.918±2.337	3.734±0.989	7.953±3.158	4.179±1.185	9.107±3.143	3.175±1.588	8.576±3.545	4.130±1.271
Hist	11.038±2.981	5.931±1.690	7.041±2.498	2.504±0.834	13.922±4.083	4.858±0.832	8.354±2.575	3.882±0.921
Isoreg	7.500±2.318	4.074±1.309	7.091±2.625	2.483±0.643	9.064±2.076	3.139±1.195	7.130±2.800	3.005±0.470
NearIso	10.561±3.057	3.385±1.542	8.071±3.265	3.202±0.893	11.267±3.045	6.463±1.502	8.677±1.519	3.110±0.914
LiTE	6.978±3.273	3.073±1.032	5.453±2.480	2.789±0.979	7.902±2.775	2.629±1.465	6.287±2.295	2.899±0.844
ACP	9.875±2.917	3.639±1.304	8.025±3.165	3.164±1.524	9.277±3.486	4.034±1.146	8.173±1.732	4.705±1.644
SmoIsoReg	6.872±2.001	2.901±0.562	5.071±2.321	2.720±0.786	6.532±1.521	2.612±1.213	4.569±2.329	2.044±0.768
RPR	**4.291**±1.212	**1.677**±0.734	**3.615**±1.881	2.613±0.751	**4.817**±1.388	**2.539**±0.727	**4.351**±2.289	**1.952**±0.901
BBQ	5.752±2.981	3.643±0.735	5.468±3.264	2.557±1.266	10.820±2.405	2.938±1.179	6.641±3.510	2.329±1.069
ENIR	6.687±2.079	2.691±1.517	7.660±3.616	2.909±1.140	6.816±1.404	2.631±1.416	6.985±2.489	3.152±1.212
ELiTE	6.731±1.885	2.492±1.093	4.110±2.109	**2.244**±0.718	6.300±2.303	3.590±1.288	5.836±1.992	2.143±0.666

In each cell $a \pm b$: a is the average and b is the standard deviation of 50 performances. The best performance in each column is in bold. The second best performance in each column is underlined.

TABLE III
CALIBRATION RESULTS BASED ON SVM SCORES

	Adult				Bank Marketing			
	$N = 200$		$N = 500$		$N = 200$		$N = 500$	
	ECE	MCE	ECE	MCE	ECE	MCE	ECE	MCE
Platt	8.684±2.538	4.971±1.956	8.442±3.866	4.043±3.795	7.061±2.990	5.500±1.438	6.536±4.277	4.594±2.825
HistBin	11.785±3.924	7.227±4.389	6.947±3.594	4.640±2.725	12.289±5.759	6.443±3.495	8.353±3.222	4.407±2.797
IsoReg	9.732±4.062	5.301±3.809	8.875±3.319	4.487±3.126	9.353±4.495	5.456±2.019	8.113±3.909	5.268±1.995
NearIso	13.381±4.558	8.299±2.772	10.543±3.948	6.259±3.864	11.901±4.470	4.275±2.357	5.262±3.821	3.962±2.548
LiTE	8.722±4.628	5.695±3.552	6.835±5.725	3.817±3.456	8.512±3.017	4.785±2.371	6.419±5.468	5.290±3.137
ACP	8.023±2.898	3.017±1.154	7.033±2.796	2.628±1.069	7.425±2.578	3.417±0.997	7.170±2.775	3.314±1.120
SmoIsoReg	5.005±2.385	3.543±1.129	5.295±2.122	2.820±1.127	5.541±2.718	**2.473**±1.663	5.968±2.715	3.071±1.236
RPR	**4.440**±1.601	**2.239**±0.932	**4.362**±1.965	**2.002**±1.219	**4.704**±2.761	2.615±1.038	4.549±1.828	2.549±1.250
BBQ	8.007±4.298	4.936±2.052	7.897±4.405	4.609±2.965	6.199±4.648	4.075±1.787	6.256±3.897	3.580±1.798
ENIR	9.314±4.981	4.303±1.768	8.115±3.399	4.373±2.841	7.574±2.351	4.041±2.037	6.154±2.738	3.036±1.642
ELITE	8.292±2.113	4.539±2.069	6.583±2.605	2.851±1.175	6.589±3.563	3.648±0.958	**4.138**±2.030	**2.354**±1.161

In each cell $a \pm b$: a is the average and b is the standard deviation of 50 performances. The best performance in each column is in bold. The second best performance in each column is underlined.

REFERENCES

[1] A. K. Menon, X. J. Jiang, S. Vembu, C. Elkan, and L. Ohno-Machado, "Predicting accurate probabilities with a ranking loss," in *Proceedings of the 29-th International Conference on Machine Learning.*, vol. 2012, 2012, p. 703.

[2] V. Vovk, I. Petej, and V. Fedorova, "Large-scale probabilistic predictors with and without guarantees of validity," in *Advances in Neural Information Processing Systems*, 2015, pp. 892–900.

[3] T. Leathart, E. Frank, G. Holmes, and B. Pfahringer, "Probability calibration trees," in *Proceedings of the 9-th Asian Conference on Machine Learning*, 2017, pp. 145–160.

[4] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34-th International Conference on Machine Learning*, 2017, pp. 1321–1330.

[5] G. M. Cordeiro and P. McCullagh, "Bias correction in generalized linear models," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 53, no. 3, pp. 629–643, 1991.

[6] G. King and L. Zeng, "Logistic regression in rare events data," *Political Analysis*, vol. 9, no. 2, pp. 137–163, 2001.

[7] X. Jiang, M. Osl, J. Kim, and L. Ohno-Machado, "Smooth isotonic regression: A new method to calibrate predictive models," *AMIA Summits on Translational Science Proceedings*, vol. 2011, p. 16, 2011.

[8] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[9] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers," in *Proceedings of the 18-th International Conference on Machine Learning*, 2001, pp. 609–616.

[10] ——, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the 8-th International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 694–699.

[11] M. P. Naeini and G. F. Cooper, "Binary classifier calibration using an ensemble of near isotonic regression models," in *Proceedings of the 16-th International Conference on Data Mining*, 2016, pp. 360–369.

[12] ——, "Binary classifier calibration using an ensemble of piecewise linear regression models," *Knowledge and Information Systems*, vol. 54, no. 1, pp. 151–170, 2018.

[13] ——, "Binary classifier calibration using an ensemble of linear trend estimation," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, 2016, pp. 261–269.

[14] X. Jiang, M. Osl, J. Kim, and L. Ohno-Machado, "Calibrating predictive model estimates to support personalized medicine," *Journal of the American Medical Informatics Association*, vol. 19, no. 2, pp. 263–274, 2011.

[15] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using Bayesian binning." in *Proceedings of the 29-th AAAI Conference on Artificial Intelligence*, 2015, pp. 2901–2907.

[16] ——, "Binary classifier calibration using a Bayesian non-parametric approach," in *Proceedings of the 2015 SIAM International Conference on Data Mining*, 2015, pp. 208–216.

[17] R. Hettich and K. O. Kortanek, "Semi-infinite programming: theory, methods, and applications," *SIAM Review*, vol. 35, no. 3, pp. 380–429, 1993.

[18] R. Reemtsen and J.-J. Rückmann, *Semi-infinite Programming*. Springer Science & Business Media, 1998, vol. 25.

[19] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[20] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," Stanford University and University of Toronto, pp. 1–28, 1996.

[21] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22-nd International Conference on Machine Learning*, 2005, pp. 625–632.

[22] M. Ayer, H. Brunk, G. Ewing, W. Reid, E. Silverman *et al.*, "An empirical distribution function for sampling with incomplete information," *Annals of Mathematical Statistics*, vol. 26, no. 4, pp. 641–647, 1955.

[23] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "ℓ_1 trend filtering," *SIAM Review*, vol. 51, no. 2, pp. 339–360, 2009.

[24] X. Wang and F. Li, "Isotonic smoothing spline regression," *Journal of Computational and Graphical Statistics*, vol. 17, no. 1, pp. 21–37, 2008.

[25] F. N. Fritsch and R. E. Carlson, "Monotone piecewise cubic interpolation," *SIAM Journal on Numerical Analysis*, vol. 17, no. 2, pp. 238–246, 1980.

[26] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, vol. 20, no. 3, pp. 197–243, 1995.

[27] J. Friedman and R. Tibshirani, "The monotone smoothing of scatterplots," *Technometrics*, vol. 26, no. 3, pp. 243–250, 1984.

[28] P. Hall and L.-S. Huang, "Nonparametric kernel regression subject to monotonicity constraints," *Annals of Statistics*, vol. 29, no. 3, pp. 624–647, 2001.

[29] X. He and P. Ng, "COBS: Qualitatively constrained smoothing via linear programming," *Computational Statistics*, vol. 14, no. 3, pp. 315–338, 1999.

[30] J. Wang and S. K. Ghosh, "Shape restricted nonparametric regression with Bernstein polynomials," *Computational Statistics & Data Analysis*, vol. 56, no. 9, pp. 2729–2741, 2012.

[31] S. R. Cosslett, "Distribution-free maximum likelihood estimator of the binary choice model," *Econometrica*, vol. 51, no. 3, p. 765, 1983.

[32] L. W. Zhong and J. T. Kwok, "Accurate probability calibration for multiple classifiers," in *Proceedings of the 23-rd International Joint Conference on Artificial Intelligence*, 2013, pp. 1939–1945.

[33] H. Brunk, "Maximum likelihood estimates of monotone parameters," *Annals of Mathematical Statistics*, vol. 26, no. 4, pp. 607–616, 1955.

[34] C. Hildreth, "Point estimates of ordinates of concave functions," *Journal of the American Statistical Association*, vol. 49, no. 267, pp. 598–619, 1954.

[35] A. Painsky and S. Rosset, "Isotonic modeling with non-differentiable loss functions with application to lasso regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 308–321, 2016.

[36] Y. Wang and H. Ni, "Multivariate convex support vector regression with semidefinite programming," *Knowledge-Based Systems*, vol. 30, pp. 87 – 94, 2012.

[37] R. Mazumder, A. Choudhury, G. Iyengar, and B. Sen, "A computational framework for multivariate convex regression and its variants," *Journal of the American Statistical Association*, vol. 0, no. 0, p. In press, 2018.

[38] Y. Wang, S. Wang, C. Dang, and W. Ge, "Nonparametric quantile frontier estimation under shape restriction," *European Journal of Operational Research*, vol. 232, pp. 671–678, 2014.

[39] A. Lanza and L. Di Stefano, "Statistical change detection by the pool adjacent violators algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1894–1910, 2011.

[40] Y. Nesterov, "Squared functional systems and optimization problems," in *High Performance Optimization*. Springer-Verlag, 2000, pp. 405–440.

[41] Y. Nesterov and A. Nemirovsky, "Interior point polynomial algorithms in convex programming," *Studies in Applied Mathematics Philadelphia SIAM*, vol. 13, 1994.

[42] C. C. Gonzaga and M. J. Todd, "An $o(\sqrt{n}l)$ iteration large-step primal-dual affine algorithm for linear programming," *SIAM Journal on Optimization*, vol. 2, no. 3, pp. 349–359, 2006.

[43] R. A. DeVore and G. G. Lorentz, *Constructive Approximation*. Springer-Verlag, 1993.

[44] G. G. Lorentz, M. v. Golitschek, and Y. Makovoz, *Constructive Approximation: Advanced Problems*. Springer-Verlag, 1996.

[45] L. Györfi, M. Kohler, A. Krzyzak, and H. Walk, *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

[46] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[47] O. Ozdemir, T. Allen, S. Choi, T. Wimalajeewa, and P. Varshney, "Copula based classifier fusion under statistical dependence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 0, no. 0, p. In Press, 2018.

[48] V. Kuleshov and P. S. Liang, "Calibrated structured prediction," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 3474–3482.

Yong-qiao Wang received the Ph.D. degree in Management Sciences and Engineering from Academy of Mathematics and Systems Science, Chinese Academy of Sciences (CAS), Beijing in 2005. He is currently a full professor in School of Finance, Zhejiang Gongshang University. His research interests include machine learning, data mining and financial risk management. He has co-authered more than 10 papers in journals including IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Fuzzy Systems, and European Journal of Operational Research.

Li-shuai Li received a Ph.D. and a M.Sc. in Air Transportation Systems from the Department of Aeronautics and Astronautics at Massachusetts Institute of Technology (MIT). She received a B.Eng. in Aircraft Design and Engineering from Fudan University. Before joining CityU, she was a consultant at McKinsey & Company in San Francisco. Currently Li is an Assistant Professor in the Department of Systems Engineering and Engineering Management at City University of Hong Kong. She is interested in innovative methods for the design, management, operation of air transportation systems, drawing expertise in Big Data and Information Technology. She develops data analysis and decision support tools to improve aircraft operations, airspace efficiency, and system safety. Her current research focuses on the application of data mining techniques in airline safety management.

Chuang-yin Dang received his PhD degree (Cum Laude) in Operations Research/Mathematical Economics from Tilburg University of the Netherlands in 1991. He is currently full Professor of Systems Engineering and Engineering Management at City University of Hong Kong. Prior to this, Prof. Dang held faculty positions at the University of California at Davis, Delft University of Technology and Auckland University, and was research fellow at the Cowles Foundation for Research in Economics of Yale University. Prof. Dang's research focuses on systems modeling, analysis and optimization. Due to his significant contributions, Prof. Dang received the award of outstanding research achievements of the year from Tilburg University of The Netherlands in 1990 and invited to give talks at such universities as Cornell University, Stanford University, the University of Michigan at Ann Arbor, the University of Minnesota at Minneapolis, and to work as Research Fellow at Yale University in 1994. Prof. Dang has published over 130 papers in journals including Mathematical Programming, Mathematics of Operations Research, SIAM Journal on Optimization, Computational Optimization and Applications, European Journal of Operational Research, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Automatic Control, IEEE Transactions on Computers, IEEE Transactions on Cybernetics, IEEE Transactions on Evolutionary Computation, IEEE Transactions on Fuzzy Systems, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Neural Networks, Artificial Intelligence, Neural Computation.