A Minimax Probability Machine for Nondecomposable Performance Measures

Junru Luo®, Hong Qiao®, Fellow, IEEE, and Bo Zhang®, Member, IEEE

Abstract—Imbalanced classification tasks are widespread in many real-world applications. For such classification tasks, in comparison with the accuracy rate (AR), it is usually much more appropriate to use nondecomposable performance measures such as the area under the receiver operating characteristic curve (AUC) and the Fβ measure as the classification criterion since the label class is imbalanced. On the other hand, the minimax probability machine is a popular method for binary classification problems and aims at learning a linear classifier by maximizing the AR, which makes it unsuitable to deal with imbalanced classification tasks. The purpose of this article is to develop a new minimax probability machine for the Fβ measure, called minimax probability machine for the Fβ-measures (MPMF), which can be used to deal with imbalanced classification tasks. A brief discussion is also given on how to extend the MPMF model for several other nondecomposable performance measures listed in the article. To solve the MPMF model effectively, we derive its equivalent form which can then be solved by an alternating descent method to learn a linear classifier. Further, the kernel trick is employed to derive a nonlinear MPMF model to learn a nonlinear classifier. Several experiments on real-world benchmark datasets demonstrate the effectiveness of our new model.

Index Terms—Imbalanced classification, minimax probability machine, nondecomposable performance measures.

I. INTRODUCTION

BINARY classification may be the most encountered problem in real-world applications and has been extensively studied. The standard binary model seeks to learn a classifier by maximizing the accuracy rate (AR) or minimizing the error rate. However, the AR is not an appropriate metric in the setting where the label class is imbalanced [1]–[4]. A simple majority algorithm can guarantee a high prediction accuracy in such a case. Many algorithms were devised to deal with imbalanced classification problems, based on the cost-sensitive learning and resampling techniques [5]–[10]. A comprehensive survey on related works can be found in [7], [11]–[13]. The classical imbalanced learning algorithms aim at maximizing a modified AR. Several other evaluation metrics have been proposed to measure the learned classifier, including AUC, Fβ-measure, and the geometric mean (GM) of the true positive rate (TPR) and the True Negative Rate (TNR) [14]–[16], which are more appropriate for imbalanced classification tasks. But, different from the AR, these measures cannot be expressed as a sum of independent metrics on individual examples and thus are called nondecomposable performance measures. They give a more holistic evaluation of the entire data, which makes them challenging to be optimized.

In recent years, many methods have been proposed which focus on optimizing nondecomposable performance measures [17]–[21]. By [22], these methods fall into two groups: the empirical utility maximization (EUM) and the Decision-Theoretic Approach (DTA).

EUM is also called the population utility (PU) in [23]. It learns a classifier by maximizing the corresponding empirical measures based on the training data and then predicts an unseen instance by using the learned classifier. Several works have applied the traditional algorithms, such as SVM and logistic regression, to the nondecomposable performance measure maximization by replacing the error rate with a specially designed surrogate loss function [1], [24]–[26]. Recently, studies are focused on the plug-in approach, which learns a class probability function first based on a training example and then makes a decision of the threshold according to another class probability function first based on a training example and then makes a decision of the threshold according to another.

Manuscript received 24 June 2020; revised 30 December 2020 and 22 June 2021; accepted 16 August 2021. Date of publication 2 September 2021; date of current version 3 May 2023. The work of Hong Qiao was supported by the NNSF of China under Grant 91948303 and Grant 61627808. (Corresponding author: Bo Zhang.)

Junru Luo is with the School of Computer Science and Artificial Intelligence and the Aliyun School of Big Data, Changzhou University, Changzhou, Jiangsu 213100, China (e-mail: luojunru@cczu.edu.cn).

Hong Qiao is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: hong.qiao@ia.ac.cn).

Bo Zhang is with the LSEC and Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: b.zhang@amt.ac.cn).

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
in Section II, many nondecomposable performance measures except AM aim at minimizing a polynomial combination of FPR and FNR rather than a linear combination of them. The cost-sensitive learning approach seems not an appropriate method for these metrics.

DTA was first proposed in [29]. It considers set classifiers and predicts the label of a test set by maximizing its expected utility (ETU) in [23].

The minimax probability machine (MPM) method is a competitive algorithm for binary classifier problems which was first proposed in [33]. It aims at maximizing the AR of a random instance in a worst case setting, where only the mean and covariance matrix of each class are assumed to be known [33]. It forces the worst case probability of misclassification of each class to be exactly equal, which is usually not true in real-world applications. The minimum error minimax probability machine (MEMPM) removes this constraint and seeks to learn a linear classifier by minimizing a weighted probability of misclassification of the future data in the same worst case setting [34]. However, MPM and MEMPM are not suitable for imbalanced classification tasks since they are based on the AR which is not an appropriate metric for such tasks, as mentioned above.

In this article, we develop an MPM model based on nondecomposable performance measures for the first time. This model can deal with imbalanced classification tasks where the mean and covariance matrix of data are given or can be estimated, which is different from [33] and [34]. We first present a minimax probability machine for the $F_\beta$-measures (MPMF) and then briefly discuss how to extend it to several other nondecomposable performance measures as listed in Table I. Further, to solve MPMF effectively, its equivalent minimization formulation is derived whose objective function is given in terms of a polynomial combination of FPR and FNR. The equivalent minimization problem is then solved by the alternative descent method. The convergence of the method is also analyzed and verified numerically. Furthermore, we explore the kernel trick in the setting to obtain a nonlinear MPM model, yielding a nonlinear classifier that can effectively deal with nonlinearly separable imbalanced problems. Note that, due to the difficulty caused by the nondecomposability, most of the existing works related to nondecomposable metrics only consider linear models. A main feature of our method is that it only makes use of the mean and covariance matrix of data and thus is independent of the training data size, making our method appropriate for large-scale problems. Another feature of our method is that it has no hyper-parameters to choose. Several experiments on real-world benchmark datasets were presented in the article and compared with the plug-in method (a state-of-the-art method) illustrate that our MPMF method is effective for imbalanced classification problems.

The article is organized as follows. In Section II, we introduce several nondecomposable performance measures for binary classification and derive their equivalent minimization objective functions, which can be expressed as a polynomial combination of FPR and FNR. In Section III, the minimax probability machine and its variants for the AR maximization are presented. In Section IV, we present the MPM model with the $F_\beta$ metric which is then solved by using the alternative descent method. In Section V, we propose a nonlinear $F_\beta$ maximization based on the kernel trick and the minimax probability machine. Several experiments are presented in Section VI, and conclusions are given in Section VII.

II. PROBLEM SETTING

In this section, we present several nondecomposable performance measures in binary classification.

A. Binary Classification

Let $\mathcal{X} \subset \mathbb{R}^d$ be the instance space and let $\mathcal{Y} = \{+1, -1\}$ be the label set with the distribution $D$ over $\mathcal{X} \times \mathcal{Y}$. We aim to learn a classifier $f \in \mathcal{F}$, which predicts a label based on the seeing feature and makes a minimal prediction error rate. Given a classifier $f : \mathcal{X} \rightarrow \mathbb{R}$ and an instance $x \in \mathcal{X}$, the label of $x$ is assigned to be $+1$ if $f(x) > 0$, and $-1$ if otherwise. Then the binary classification problem can be expressed as

$$\arg_{f \in \mathcal{F}} \max_{(x, y) \sim D} \Pr(yf(x) > 0).$$

Note that the prior knowledge of the instance space plays a key role in machine learning problems. Suppose the true distribution $D$ is explicitly given. Then the Bayesian estimator can be computed by solving (1). If the distribution class is given, we can first estimate the distribution $D$ with the maximum likelihood method and then learn a classifier. In general, the true distribution $D$ or its distribution class is not known in practice. But the moment statistics of the input space can be estimated, and thus the minimax probability machine can be developed. In the worst case, the sample data can be used to learn a classifier by minimizing the empiric risk.

B. Nondecomposable Performance Measures

Let $p = \Pr(y = +1)$ be the probability of the positive examples. We now introduce some basic notations

$$\text{TPR} := 1 - \text{FNR} = \Pr(\hat{y} = +1 | y = +1)$$
$$\text{TNR} := 1 - \text{FPR} = \Pr(\hat{y} = -1 | y = -1)$$
$$\text{Precision} := \Pr(y = 1 | \hat{y} = 1) = \frac{p \cdot \text{TPR}}{p \cdot \text{TPR} + (1 - p) \cdot \text{FPR}}$$

A better classifier should have a greater TPR, TNR, and precision values, but there is a tradeoff among these measures. It is unable to achieve the optimal values simultaneously for TPR, TNR, and precision since each of these quantities can be maximized at the cost of other measures. So, usually,
we consider the criterion that adjusts the basic quantities, such as several combinations of TPR and TNR. Table I presents some usually used performance measures. In this article, we focus on the criterion that adjusts the basic quantities, such as the monic mean (HM) of TPR and TNR, the GM of TPR and TNR, the quadratic mean (QM) of TPR and TNR, the harmonic mean (HM) of TPR and TNR, the AM, and the Jaccard coefficient (JAC). Maximizing these measures is also equivalent to minimizing 1 − F0, where F0 is a polynomial combination of FPR and FNR defined by

\[
F_0 = \frac{(\beta^2 + 1) \cdot \text{TPR} \cdot \text{Precision} + \beta^2 \cdot \text{Precision} + \text{TPR}}{(1 + \beta^2) \cdot p \cdot \text{TPR} + (1 - p) \cdot \text{FPR} + \beta^2 p \cdot \text{FNR}}.
\]

A larger F0 means a better corresponding classifier.

**Lemma 1:** The classifier \( f \) maximizes \( F_0 \) if and only if it minimizes \( Q_F \), where \( Q_F \) is a polynomial combination function of FPR and FNR defined by

\[
Q_F = \sum_{i=0}^{\infty} \left[ \frac{(1 - p) \cdot \text{FPR} + \beta^2 p \cdot \text{FNR}}{1 - \text{FNR}} \right] \cdot (\text{FNR})^i.
\]

**Proof:** Since \( F_0 > 0 \), then maximizing \( F_0 \) is equivalent to minimizing \( 1/F_0 \). We have

\[
\frac{1}{F_0} = 1 + \frac{(1 - p) \cdot \text{FPR} + \beta^2 p \cdot \text{FNR}}{p(1 + \beta^2)(1 - \text{FNR})} = 1 + \sum_{i=0}^{\infty} \frac{[(1 - p) \cdot \text{FPR} + \beta^2 p \cdot \text{FNR}] \cdot (\text{FNR})^i}{p(1 + \beta^2)}.
\]

The proof is thus complete. \( \square \)

Note that the above method can be extended to other nondecomposable performance measures, such as the AM, the GM, the quadratic mean (QM) of TPR and TNR, the harmonic mean (HM) of TPR and TNR, the GM of TPR and TNR, the Jaccard coefficient (JAC). Maximizing these measures is also equivalent to minimizing \( Q_{\alpha d} \), which can also be expressed as a polynomial combination of FPR and FNR, that is, \( Q_{\alpha d} = \sum_{i,j=0}^{\infty} \tau_{ij}(\text{FPR})^j(\text{FNR})^i \) (see the third column in Table I).

### III. MPM WITH THE ACCURACY RATE MEASURE

Different from the traditional classification algorithms, such as SVMs and neural networks, which seek to learn a real-valued function by minimizing the error rate on a given training dataset, MPM tries to separate the two classes of the data samples with the maximal probability in a worst case setting where only the mean and covariance matrix of each class are given in advance [33]. Let the notation \( x \sim (\mu, \Sigma) \) denote that \( x \) belongs to the class of distributions having the mean \( \mu \) and the covariance matrix \( \Sigma \). Assume that the mean and covariance matrix of the positive and negative samples, denoted, respectively, by \( \{\mu_F, \Sigma_F\} \) and \( \{\mu_N, \Sigma_N\} \), are all reliable. Then MPM can be expressed as the optimization problem

\[
\begin{align*}
\max_{0 < a < 1, w \neq 0, b} & \quad a \\
\text{s.t.} & \quad \inf_{x \sim (\mu_F, \Sigma_F)} \Pr\{w^T x_F \geq b\} \geq a \\
& \quad \inf_{x \sim (\mu_N, \Sigma_N)} \Pr\{w^T x_N \leq b\} \geq a
\end{align*}
\]

where the inf is taking over all distributions with mean \( \mu \) and covariance matrix \( \Sigma \). Let \( (\alpha^*, \omega^*, b^*) \) be the optimal solution of (2). It is guaranteed that the misclassification probability is less than \( 1 - \alpha^* \) for any future data sample with the learned classifier \( f(x) = (\omega^*)^T x - b^* \).

To solve the optimization problem (2), we need to remove the unknown probability in the constraint by using the following result established in [35] (see also [36, Th. 6.1]).

**Lemma 2:** Suppose \( x \in \mathbb{R}^p \) is a random vector with \( x \sim (\mu, \Sigma) \) and \( S \subset \mathbb{R}^p \) is a given convex set. Then the supremum of the probability of \( x \in S \) is given as \( \sup_{x \sim (\mu, \Sigma)} \Pr\{x \in S\} = 1/(1 + d^2) \), where \( d^2 = \inf_{x \in S} (x - \mu)^T \Sigma^{-1} (x - \mu) \).

If the convex set \( S \) is a half-space, then \( d \) can be calculated explicitly, leading to the following lemma which was proven in [33].

**Lemma 3:** Given \( w \in \mathbb{R}^p \) with \( w \neq 0 \) and \( b \in \mathbb{R} \), the condition

\[
\inf_{x \sim (\mu, \Sigma)} \Pr\{w^T x \leq b\} \geq a
\]

holds if and only if

\[
b - w^T x \geq \kappa(a) \sqrt{w^T \Sigma w}
\]

where \( \kappa(a) = \sqrt{a/(1 - a)} \).

---

**Table I**

| Measure | Definition \((P_{\alpha d}(\text{TPR}, \text{TNR}))\) | Minimization Objective \((Q_{\alpha d}(\text{FNR}, \text{FPR}))\) |
|---------|-------------------------------------------------|-------------------------------------------------|
| AR      | \( p \cdot \text{TPR} + (1 - p) \cdot \text{TNR} \) | \( p \cdot \text{FNR} + (1 - p) \cdot \text{FPR} \) |
| AM      | \((\text{TPR} + \text{TNR})/2\)                  | \((\text{FNR} + \text{FPR})/2\)                  |
| QM      | \( 1 - [(1 - \text{TPR})^2 + (1 - \text{TNR})^2]/2 \) | \( [(\text{FNR})^2 + (\text{FPR})^2]/2 \) |
| \( F_0 \) | \((1 + \beta^2) \cdot \text{TPR} + \beta^2 \cdot \text{TPR} + (1 - \beta^2) \cdot \text{FPR} + \beta^2 p \cdot \text{FNR} \) | \( \sum_{i=0}^{\infty} \beta^2 p \cdot \text{FNR} + (1 - p) \cdot \text{FPR} \cdot (\text{FNR})^i \) |
| HM      | \( 2 \cdot \text{TPR} \cdot \text{TNR} / (\text{TPR} + \text{TNR}) \) | \( \sum_{i=0}^{\infty} (\text{FNR})^i \cdot (\text{FPR})^i \) |
| GM      | \( \sqrt{\text{TPR} \cdot \text{TNR}} \)         | \( \sum_{i=0}^{\infty} (\text{FNR})^i \cdot (\text{FPR})^i \) |
| G-TP/PR | \( \sqrt{\text{TPR} \cdot \text{FPR}} \)         | \( \sum_{i=0}^{\infty} (\text{FNR})^i \cdot (\text{FPR})^i \) |
| JAC     | \( p \cdot \text{TPR} / \{p \cdot \text{TPR} + p \cdot \text{FNR} + (1 - p) \cdot \text{FPR}\} \) | \( p \cdot \text{FNR} + (1 - p) \cdot \text{FPR} \cdot \sum_{i=0}^{\infty} (\text{FNR} \cdot (2 - \text{FNR}))^i \) |
It is easy to see that \( \kappa(\alpha) \) is a monotonically increasing function of \( \alpha \). By Lemma 3, the MPM problem (2) becomes

\[
\max_{0 < \alpha_1, 0 < \alpha_2, 1, w \neq 0, b} \kappa(\alpha)
\]

s.t. 
\[
- b + w^T \mu_P \geq \kappa(\alpha) \sqrt{w^T \Sigma_P w}
\]

\[
- b - w^T \mu_N \geq \kappa(\alpha) \sqrt{w^T \Sigma_N w}.
\]

Further, by eliminating \( b \) the above problem reduces to

\[
\max_{\gamma, w \neq 0} \gamma
\]

s.t. 
\[
w^T (\mu_P - \mu_N) \geq \gamma \left( \sqrt{w^T \Sigma_P w} + \sqrt{w^T \Sigma_N w} \right).
\]

Without loss of generality, we may set \( w^T (\mu_P - \mu_N) = 1 \) (see [33]). As a result, MPM is finally equivalent to the following constrained second-order cone programming (SOCP):

\[
\min_{w \neq 0} \| \Sigma_P^{1/2} w \|_2 + \| \Sigma_N^{1/2} w \|_2
\]

s.t. 
\[
w^T (\mu_P - \mu_N) = 1.
\]

The above SOCP problem can be solved by the interior point method [37]. Now, write \( w = w_0 + Fu \) with \( w_0 = (\mu_P - \mu_N) \| \mu_P - \mu_N \|_2^{-2} \) and \( F \in \mathbb{R}^{p \times (p-1)} \) the orthogonal matrix whose columns span the subspace of the vectors orthogonal to \( (\mu_P - \mu_N) \). Then the constraint can be eliminated to get the unconstrained SOCP [33]

\[
\min_{w \neq 0} \| \Sigma_P^{1/2} (w_0 + Fu) \|_2 + \| \Sigma_N^{1/2} (w_0 + Fu) \|_2
\]

A block coordinate descent method is then used to solve the above equivalent unconstrained SOCP [33].

In many real-world applications, the tolerance of the misclassification probability may be different for the positive and negative classes, as seen in the disease diagnosis problem. Thus it is worth increasing the TPR measure at the expense of a lower TNR measure, leading to the biased minimax probability machine (BMPM) and the minimum error minimax probability machine (MEMPM) [34], [38]. MEMPM can be expressed as

\[
\max_{0 < \alpha_1, 1, \alpha_2 < 1, w \neq 0, b} \alpha_1 \left( 1 - p \right) \alpha_2
\]

s.t. 
\[
\inf_{x \sim (\mu_P, \Sigma_P)} \Pr(w^T x_P \geq b) \geq \alpha_1
\]

\[
\inf_{x \sim (\mu_N, \Sigma_N)} \Pr(w^T x_N \leq b) \geq \alpha_2.
\]

If \( \alpha_2 \in (0, 1) \) is a predefined constant, then (3) becomes the BMPM problem. The objective function in (3) is a weighted misclassification probability. Lemma 3 can also be applied to simplify the optimization problem (3). Let \( (\alpha_1^*, \alpha_2^*, w^*, b^*) \) be the optimal solution of (3) and let \( f(x) = w^T x - b^* \) be the learned classifier. Then it is guaranteed that the obtained TPR and TNR are at least equal to \( \alpha_1^* \) and \( \alpha_2^* \), respectively. In MEMPM, \( \alpha_1^* \) and \( \alpha_2^* \) are not necessarily the same, which is different from the case in MPM. Further, the experimental results obtained in [34] demonstrate the effectiveness of MEMPM. Setting \( p = 1/2 \) in the optimization problem (3), a biased minimax probability machine was derived in [39] for imbalanced classification problems. This machine tries to maximize the AM performance measure in a worst case setting where only the mean and covariance matrix of each class are given. It can be regarded as a special case of our method, as will be illustrated in the next section.

IV. MPM WITH NONDECOMPOSABLE PERFORMANCE MEASURES

In this section, we develop an MPM method with nondecomposable performance measures to deal with imbalanced classification tasks. We first consider the \( F_B \) measure.

For an imbalanced classification problem, suppose the probability of an instance belonging to the positive class \( p \) is small enough. Then the MEMPM problem (3) is equivalent to maximizing \( \alpha_2 \) or TNR, which is not appropriate for this case. To address this issue, we replace the objective function in (3) with the global performance metric \( F_B \), which is a function of TPR (or \( \alpha_1 \)) and TNR (or \( \alpha_2 \)). Thus we get the problem

\[
\max_{0 < \alpha_1, 1, \alpha_2 < 1, w \neq 0, b} F_B
\]

s.t. 
\[
\inf_{x \sim (\mu_P, \Sigma_P)} \Pr(w^T x_P \geq b) \geq \alpha_1
\]

\[
\inf_{x \sim (\mu_N, \Sigma_N)} \Pr(w^T x_N \leq b) \geq \alpha_2.
\]

By Lemma 1, maximizing \( F_B \) is equivalent to minimizing \( Q_F \). Set \( \alpha_P = \text{FNR}, \alpha_N = \text{FPR} \). Then (4) becomes

\[
\min_{\alpha_P, \alpha_N, w \neq 0, b} \frac{1}{1 - \alpha_P}
\]

s.t. 
\[
\inf_{x \sim (\mu_P, \Sigma_P)} \Pr(w^T x_P \geq b) \geq 1 - \alpha_P
\]

\[
\inf_{x \sim (\mu_N, \Sigma_N)} \Pr(w^T x_N \leq b) \geq 1 - \alpha_N
\]

\[
0 < \alpha_P < 1, \ 0 < \alpha_N < 1.
\]

We call this model the minimax probability machine for the \( F_B \)-measure (MPMF). For simplicity, define \( Q_F(\alpha_P, \alpha_N; w) := Q_F = \{(1 - p)\alpha_N + \beta^2 p \alpha_P (1 - \alpha_P)^{-1}\} \) and omit the conditions \( 0 < \alpha_P < 1, 0 < \alpha_N < 1 \) in what follows. Note that these two constrained conditions are guaranteed to be satisfied by the optimal value of \( \alpha_P, \alpha_N \) obtained below (see (11) and the sentences following (12) below). By Lemma 1, \( F_B = (1 + \beta^2) P\{Q_F + (1 + \beta^2) p\}^{-1} \). Suppose \( (\alpha_P^*, \alpha_N^*, w^*, b^*) \) is the optimal solution of the MPMF problem (5) and \( f(x) = (w^*)^T x - b^* \) is the learned classifier. Then \( Q_F \geq Q_F^* := Q_F(\alpha_P^*, \alpha_N^*; w^*) \) for any future data sample, so the \( F_B \) metric is not bigger than \( F_B^* := (1 + \beta^2) P\{Q_F^* + (1 + \beta^2) p\}^{-1} \) for any future data sample.

We now propose an algorithm to solve the optimization problem (5). By Lemma 3, we can remove the probability terms without any distribution assumption and obtain the following optimization problem which is equivalent to the problem (5):

\[
\min_{\alpha_P, \alpha_N, w \neq 0, b} \frac{(1 - p)\alpha_N + \beta^2 p \alpha_P}{1 - \alpha_P}
\]

s.t. 
\[
w^T \mu_P - b \geq \pi(\alpha_P) \sqrt{w^T \Sigma_P w}
\]

\[
w^T \mu_N \geq \pi(\alpha_N) \sqrt{w^T \Sigma_N w}
\]

where \( \pi(\alpha) = 1/\kappa(\alpha) = \sqrt{1 - \alpha}/\alpha \) is monotonically decreasing with \( \alpha \).
Lemma 4: The minimal value of (6) is achieved when the two inequalities in the constraints become equalities.

Proof: Assume that the optimal solution of (6) is $(w^*, b^*, a_P, a'_N, \alpha'_{N,t})$ and the two inequalities in the constraints hold strictly, that is,

$$(w'^T_0 \mu - b > \pi (a'_P) \sqrt{(w'^T_0 \Sigma_P w'} - \alpha'_{N,t})$$

$$b^* - (w'^T_0 \mu_N > \pi (a'_N) \sqrt{(w'^T_0 \Sigma_N w')} - \alpha_{N,t}).$$

Then the objective value is getting smaller with $a'_P$ and $\alpha'_{N,t}$ decreasing while the constraints remain to hold. This is a contradiction to the fact that $(w^*, b^*, a'_P, a'_N, \alpha'_{N,t})$ is the optimal solution of (6). The proof is thus complete.

By Lemma 4, (6) can be rewritten as

$$\min_{a_P, a_N, w \neq 0, b} \frac{(1 - p)a_N + \beta^2 p a_P}{1 - a_P}$$

$$\text{s.t. } w'^T_0 \mu - b = \pi (a'_P) \sqrt{w'^T_0 \Sigma_P w'}$$

$$b - w'^T_0 \mu_N = \pi (a'_N) \sqrt{w'^T_0 \Sigma_N w'}. \quad (7)$$

By eliminating $b$ from (7), the two equality constraints become

$$\pi (a) \sqrt{w'^T_0 \Sigma_N w} + \pi (a) \sqrt{w'^T_0 \Sigma_P w} = w'^T_0 \mu_{pn} \quad (8)$$

where $\mu_{pn} = \mu - \mu_N$. Note that (8) is positively homogeneous in $w$. Therefore we need to give an additional constraint on $w$. Following [33] and [34] we may set $w'^T_0 (\mu - \mu_N) = 1$ without loss of generality, leading to the constrained SCOP subproblem. Following [40], we set $\|w\| = 1$, leading to a constrained concave (or convex) optimization problem which is easier to solve (see (14) below). The problem (7) is then transformed into the problem

$$\min_{a_P, a_N, w} \frac{(1 - p)a_N + \beta^2 p a_P}{1 - a_P}$$

$$\text{s.t. } \|w\| = 1$$

$$\pi (a) \sqrt{w'^T_0 \Sigma_N w} + \pi (a) \sqrt{w'^T_0 \Sigma_P w} = w'^T_0 \mu_{pn} \quad (9)$$

We apply the alternative descent method to solve the non-convex problem (9). Details are presented in Algorithm 1. We initialize the classifier $w_1 = (\mu - \mu_N)/\|\mu - \mu_N\|$, so $\|w_1\| = 1$. Assume that at the $t$th round, we have obtained the classifier $w_t$ with $\|w_t\| = 1$. Let $A_t = (w'^T_0 \Sigma_P w_t)/2$, $B_t = (w'^T_0 \Sigma_N w_t)/2$ and $C_t = w'^T_0 (\mu - \mu_N)$. Then we seek $a_P$ and $a_{N,t}$ minimizing $Q_F(a_P, a_{N,t}; w_t)$ with the fixed classifier $w_t$, that is,

$$a_{P,t}, a_{N,t} = \arg \min_{a_P, a_N} \frac{(1 - p)a_N + \beta^2 p a_P}{1 - a_P}$$

$$\text{s.t. } \pi (a) B_t + \pi (a) A_t = C_t. \quad (10)$$

From the inequality constraint in (10) it follows that

$$a_N = \frac{B_t^2}{B_t^2 + \pi (a) A_t}. \quad (11)$$

Substituting $a_N$ into (10) gives

$$a_{P,t} = \arg \min_{a_P} \frac{(1 - p)B_t^2}{[B_t^2 + (C_t - \pi (a) A_t)^2](1 - a_P)} + \frac{\beta^2 p}{\pi^2 (a)} \quad (12)$$

Algorithm 1 MPM for the $F_\beta$-Measure

1: **Input:** Mean and covariance matrix of positive and negative samples are $(\mu_P, \Sigma_P)$ and $(\mu_N, \Sigma_N)$, respectively.
2: Initial classifier $w_1 = (\mu_P - \mu_N)/\|\mu_P - \mu_N\|$.
3: for $t = 1, 2, \ldots, \text{do}$
4: Given $w_t$, find $a_{P,t}$ by solving problem (12) with a grid-based search method. The corresponding $a_{N,t}$ is calculated from (11).
5: Calculate the objective function $Q = Q_F(a_{P,t}, a_{N,t}; w_t)$.
6: Set $w_1 = w_t$ and solve (14).
7: for $k = 1, 2, \ldots, \text{do}$
8: $u_k = u_k + \gamma_k \nabla f_k(u_k)$
9: $v_{k+1} = u_k/\|u_k\|$
10: Terminate if the conditions (15) are satisfied.
11: end for
12: Set $w_{t+1} = v_{k+1}$ and the corresponding $a_{N,t}$ is given as $a_{N,t} = 1/[1 + \lambda_t^2 (w_{t+1})]$.
13: Calculate the objective function $Q' = Q_F(a_{P,t}, a_{N,t}; w_{t+1})$.
14: if $Q - Q' \leq 0.0001$, terminate.
15: end for
16: Compute $b$ by using (16).

From the equality constraint in (10) again we have that $C_t - \pi (a_P) A_t \geq 0$, implying that $a_P \geq A_t^2/(A_t^2 + C_t^2)$. Make use of a grid-based search method to find $a_{P,t} \in [A_t^2/(A_t^2 + C_t^2), 1)$. The corresponding $a_{N,t}$ can then be computed from (11).

We now update the classifier $w_t$. Note that the objective $Q_F(a_{P,t}, a_{N,t}; w_t)$ will not decrease if both $a_{P,t}, a_{N,t}$ are fixed. Therefore we fix $a_{P,t}$ and update both $a_{N,t}$ and the classifier simultaneously, which gives a smaller misclassification probability ($a_{N,t} < a_{N,t}$). This can be achieved by solving the problem (9) at $a_P = a_{P,t}$. Let $\tau_t = \pi (a_{P,t})$. The problem (9) can then be transformed into the problem

$$a_{N,t} = \arg \min_{a_N} \frac{(1 - p)a_N + \beta^2 p a_P}{1 - a_P}$$

$$\text{s.t. } \pi (a) A_t = C_t. \quad (10)$$

Note that $\pi (a)$ is monotonically decreasing with $a$, so the above optimization problem is equivalent to the fractional programming (FP) problem

$$\max_w \frac{w'^T_0 \mu_{pn} - \tau_t \sqrt{w'^T_0 \Sigma_P w}}{\sqrt{w'^T_0 \Sigma_N w}}$$

$$\text{s.t. } \|w\| = 1. \quad (13)$$

Write $\lambda_t(w) = [w'^T_0 \mu_{pn} - \tau_t \sqrt{w'^T_0 \Sigma_P w}]/(w'^T_0 \Sigma_N w)^{1/2}$. It is not necessary to find an exact solution $w_{t+1}$ such that $\lambda_t(w_{t+1}) > \lambda_t(w_t)$ and $\|w_{t+1}\| = 1$. The corresponding misclassification probability is then given as $a_{N,t} = 1/[1 + \lambda_t^2 (w_{t+1})]$. Let $\eta_t = \lambda_t(w_t) > 0$ and let us define $f : \mathbb{R}^p \rightarrow \mathbb{R}$ by

$$f_k(w) := w'^T_0 \mu_{pn} - \tau_t \sqrt{w'^T_0 \Sigma_P w} - \eta_t \sqrt{w'^T_0 \Sigma_N w}.$$
Then $f_t$ has the following properties.

**Lemma 5:** 1) $f_t(w)$ is a concave function; 2) the condition $f_t(w) > 0$ is positively homogeneous in $w$; 3) $f_t(w_0) = 0$; 4) if $w_0$ satisfies that $f_t(\hat{\omega}_t) > 0$, then we have $\lambda_t(\hat{\omega}_t) > \eta_t$.

**Proof:** 1) Since $f_t(w) = w^T \mu_p - \tau_s(\Sigma_p w)^{1/2} - \eta_t(\Sigma_w w)^{1/2}$ and $\tau_s(\Sigma_p w)^{1/2}$ and $\eta_t(\Sigma_w w)^{1/2}$ are convex function, then it follows that $f_t(w)$ is concave.

2) Suppose there is a $\hat{\omega} \in \mathbb{R}^p$ such that $f_t(\hat{\omega}) > 0$. Then, for any $s > 0$ we have $f_t(s\hat{\omega}) = sf_t(\hat{\omega}) > 0$, that is, the condition $f_t(w) > 0$ is positively homogeneous in $w$.

3) Since $\eta_t = \lambda_t(\hat{\omega}_t)$, it is easy to see that $f_t(w_0) = 0$.

4) The condition $f_t(\hat{\omega}_t) > 0$ implies that $\hat{\omega}_t^T \Sigma_p \hat{\omega}_t > 0$. Then, by the definition of $\lambda_t(\hat{\omega}_t)$ and $\eta_t$ we have $\lambda_t(\hat{\omega}) > \eta_t$.

We now use Lemma 5 to find an inexact solution of (13). If $\nabla f_t(w_0) = 0$, then set $w_{t+1} = w_t$. The algorithm terminates. Otherwise, $f_t(w_0)$ does not attain its maximum at $w_t$. Hence there is a $w_{t+1}$ such that $f_t(w_{t+1}) > 0$. Choose $w_{t+1}$ as the solution of the optimization problem

$$\max_{w_t} f_t(w) \quad \text{s.t.} \quad \|w\| \leq 1. \quad (14)$$

We may use a gradient projection method to solve the problem (14). Take the initial value $w_1 = w_t$ and the step size $\eta_t = 1/k$, $k = 1, 2, \ldots$. At the $k$th sub-step, $v_k$ is updated as follows:

$$u_k = v_k + \eta_t \nabla f_t(v_k), \quad v_k = \frac{u_k}{\|u_k\|} \quad \text{where} \quad \nabla f_t(v_k) = (\mu_p - \mu_N) - \tau_s\Sigma_p v_k - \eta_t\Sigma_N v_k.$$  

The algorithm terminates at the $k$th step if $f_t(v_k)$ and $f_t(v_{k+1})$ satisfy the conditions

$$f_t(v_{k+1}) \geq 0, \quad f_t(v_k) > 0, \quad \|\nabla f_t(v_{k+1})\| \leq 0.0001. \quad (15)$$

Then we update the classifier with $w_{t+1} = v_k$. After running $T$ rounds, $b$ is given as

$$b_T = w_T^T \mu_p - \nu(\mu_p, \nu) \sqrt{w_T^T \Sigma_p w_T}. \quad (16)$$

**Theorem 1:** After running Algorithm 1, the objective value $Q(a_{p,i}, a_{N,i}; w_t)$ converges.

**Proof:** Note first that Algorithm 1 uses the alternative descent method to solve the MPMF problem (9). At the $t$th round, we obtained the misclassification probability $Q_F(a_p, a_N; w_t)$. We then fix $a_{p,i}$ and seek a classifier $w_{t+1}$ which makes $a_{N,i}$ smaller. By the definition of $Q_F$ we have $Q_F(a_p, a_{N,i}; w_t) > Q_F(a_p, a_{N,i}; w_{t+1})$. Repeat this process. Then, at the $(t + 1)$th round, update $(a_{p,i+1}, a_{N,i+1}) = \min Q_F(a_p, a_{N,i}; w_{t+1})$. Thus, $Q_F(a_p, a_{N,i}; w_t) > Q_F(a_p, a_{N,i}; w_{t+1})$. This means that the objective $Q_F(a_p, a_{N,i}; w_t)$ is monotonically decreasing with $t$, implying that $Q_F(a_p, a_{N,i}; w_t)$ converges as $t \to \infty$. The proof is thus complete.

Suppose $(a_p, a_N)$ converges to $(a^*_p, a^*_N)$. Then the classifier $w$ also converges to the optimal solution of (14) in which $\tau_p = \pi(a^*_p)$ and $\eta_t = \pi(a^*_N)$. In the experiments, we will see that both $a_p, a_N$ are convergent for several real datasets (see Fig. 2).

We have the following remarks on Algorithm 1.

1) For imbalanced data, the mean of the positive and negative samples is different, that is, $\mu_p \neq \mu_N$. Thus, when $d = \|\mu_p - \mu_N\|$ becomes larger, $a_N$ and $a_p$ will get smaller.

2) Given $w_t$, we have $C_t = \pi(a_p)A_t$. Then

$$\alpha_p \geq \frac{1}{1 + (\frac{C_t}{C})^2} \geq \frac{1}{1 + \frac{d}{w_t \Sigma_p w_t}} \geq \frac{1}{1 + \frac{d}{\lambda^2}} \quad \text{where} \quad \lambda^2 = \min_{\Sigma_p} \frac{1}{w_t \Sigma_p w_t}$$

and

$$\alpha_N \geq \frac{1}{1 + (\frac{C_t}{C})^2} \geq \frac{1}{1 + \frac{d^2}{w_t \Sigma_N w_t}} \geq \frac{1}{1 + \frac{d^2}{\lambda^2}} \quad \text{where} \quad \lambda^2 = \min_{\Sigma_N} \frac{1}{w_t \Sigma_N w_t}.$$  

3) We have the implicit constraint $w_t^T \Sigma_p w_t > 0$ for the classifier $w_t$. This is also satisfied by the initial classifier $w_1 = (\mu_p - \mu_N)/\|\mu_p - \mu_N\|$.

4) Suppose $\mu_p, \mu_N, \Sigma_p, \Sigma_N$ are fixed. Then the second constrained condition in the optimization problem (9) is equality for the unknown variables $a_{p,i}, a_{N,i}, w$. The objective function can be rewritten as $a_{p,i} + \pi(\beta, \beta) (a_p) / (1 - \alpha_p)$ with $\tau(\beta, \beta) = \beta^2 p / (1 - p)$. Thus the MPMF model depends on $\tau(\beta, \beta)$.

5) Our MPMF algorithm for $F_{\beta}$ can be naturally extended to other nondecomposable performance measures listed in Table I. In fact, replacing the objective function $Q_F$ with the general objective function $Q_{nd}(a_p, a_N)$ in the problem (5), we have an MPM with a nondecomposable performance measure (MPMND), leading to the following optimization problem similar to (9):

$$\min_{a_p, a_N, w} \sum_{i,j=0}^{\infty} \tau_{ij} a^i_N a^j_p \quad \text{s.t.} \quad \|w_0\| = 1 \quad \pi(a_N) \sqrt{w^T \Sigma_N w} + \pi(a_p) \sqrt{w^T \Sigma_p w} = w^T \mu_p. \quad (17)$$

This optimization problem can also be solved by the alternative descent method, leading to an algorithm similar to Algorithm 1. The only difference is step 3, where the problem (12) is replaced with the problem

$$a_{p,i} = \arg\min_{a_{p,i}} Q_{nd}(a_p, a_{N,i}, w_t).$$

6) If the mean and the covariance matrix are given in advance for a dataset, then the constant $b$ can be directly calculated from (16). However, in practice, the mean and the covariance matrix are usually not given in advance for many real-world datasets, and thus it is necessary to give an estimated mean and covariance matrix for the datasets which will be used to replace the true statistics in the MPMF and MPMND models (see Section VI on experiments below).

In this case, the value of $b$ calculated directly from (16) with the estimated mean and covariance matrix may not be the optimal one, and so it is reasonable to fine-tune the value of $b$ with a validated dataset to get a better result.
V. KERNELIZATION

Due to the nondecomposability of the $F_{\beta}$-measure, most algorithms for the $F_{\beta}$-measure maximization focus on linear classifiers which may not be always effective for some real-world problems. In this article, we apply the kernel trick to the minimax probability machine for the $F_{\beta}$-measure to derive the Kernel Minimax Probability Machine for $F_{\beta}$ (KMPMF) which yields a nonlinear classifier.

By [33], the kernel trick works in the MPM model. Let $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be a positive kernel function. Given the positive samples $\{x_P^i\}_{i=1}^{N_P}$ and the negative samples $\{x_N^i\}_{i=1}^{N_N}$, define the Gram matrix $K = (K_{ij})$ with

$$K_{ij} = \begin{cases} K(x_P^i, x_P^j), & \text{if } i \leq N_P, \ j \leq N_P \\ K(x_P^i, x_N^{j-N_P}), & \text{if } i \leq N_P, \ j > N_P \\ K(x_N^{i-N_P}, x_P^j), & \text{if } i > N_P, \ j \leq N_P \\ K(x_N^{i-N_P}, x_N^{j-N_P}), & \text{if } i > N_P, \ j > N_P. \end{cases}$$

The first $N_P$ rows and the last $N_N$ rows of $K$ are named the positive Gram matrix $K_P$ and the negative Gram matrix $K_N$, respectively. Denote by $I_P$ and $I_N$ their corresponding column average, which are both $N_P + N_N$ dimensional vectors. Define

$$L_P = \frac{1}{\sqrt{N_P}}(K_P - I_P I_P^T), \quad L_N = \frac{1}{\sqrt{N_N}}(K_N - I_N I_N^T)$$

where $I_P$ and $I_N$ are column vectors of ones of dimension $N_P$ and $N_N$, respectively. We have the following theorem which can be shown similarly as in the proof of [33, Th. 6].

**Theorem 2**: If $I_P = I_N$, then the minimal probability decision problem has no solution. Otherwise, the optimal decision boundary is determined by the solution $(a_P^*, a_N^*, w^*)$ of the optimization problem

$$\min_{a_P, a_N, w} \frac{(1 - p)(a_P) + \beta^2 p(a_P)}{1 - a_P}$$

s.t. $\|w\| = 1$

$$\pi(a_P) = \sqrt{w^T L_P^T L_P w} + \pi(a_N) = \sqrt{w^T L_N^T L_N w} = w^T (I_P - I_N)$$

and $b^* = (w^*)^T I_P - \pi(a_P)((w^*)^T L_P I_P w^*)^{1/2}$.

By Theorem 2, a new data point $x$ is predicted by $\hat{y} = \text{sign}(\sum_{i=1}^{N_P} w_P^i K(x_P^i, x) + \sum_{j=N_P+1}^{N_P+N_N} w_N^j K(x_N^j-N_P, x) - b^*)$.

VI. EXPERIMENTS

In this section, we present experiments to verify our MPMF model for the $F_{\beta}$ measure.

We first consider the case when the mean and covariance matrix of the data are given in advance. Let $\mu_P = (3, 1)^T$, $\mu_N = (-1, -2)^T$, $\Sigma_P = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}$ and $\Sigma_N = \begin{bmatrix} 1 & 1/3 \\ 1/3 & 1 \end{bmatrix}$. Then we calculate the maximal $\alpha_P$ (FNR) and $\alpha_N$ (FPR) with different combinations of $\beta$ and the proportion $p$ of the positive samples. Set $p = 0.5, 0.4, 0.3, 0.2, 0.1, 0.05, 0.01$ and $\beta = 1, 3$. Table II presents the FPR and FNR values in the worst case. Given the proportion $p$ of the positive samples, a larger $\beta$ helps reduce FNR in the cost of FPR, which is very important in imbalanced classification problems.

In the second experiment, we apply our method MPMND to some real-world datasets, which can be downloaded from LIBSVM website and UCI machine learning repository. Table III presents details of these datasets. We also define $p$ as the proportion of the positive samples, which is the same as in the training and test datasets. In the case of multiclass datasets, we report results (using one-vs-all classifiers) averaged over the classes. For each dataset, the reported values of the nondecomposable measures are obtained by averaging over 20 independent runs. To deal with the high-dimensional dataset MNIST which has 784 features, we first use a principal component analysis (PCA) method to reduce its feature dimension by keeping 80% variance, and then apply our MPMF/MPMND method to the dataset MNIST with reduced low-dimensional features.

Let $\{x_P^i\}_{i=1}^{N_P}$ and $\{x_N^i\}_{i=1}^{N_N}$ be the training data points with positive and negative class labels, respectively. Then the mean and covariance matrix of the dataset can be estimated as

$$\hat{\mu}_P = \frac{\sum_{i=1}^{N_P} x_P^i}{N_P}, \quad \hat{\Sigma}_P = \frac{\sum_{i=1}^{N_P} (x_P^i - \hat{\mu}_P)(x_P^i - \hat{\mu}_P)^T}{N_P - 1}$$

$$\hat{\mu}_N = \frac{\sum_{i=1}^{N_N} x_N^i}{N_N}, \quad \hat{\Sigma}_N = \frac{\sum_{i=1}^{N_N} (x_N^i - \hat{\mu}_N)(x_N^i - \hat{\mu}_N)^T}{N_N - 1}.$$
The plug-in method, which first learns a classification probability function by training a logistical regression model on a training dataset and then decides a threshold based on another dataset, is consistent with several nondecomposable performance measures. Table IV gives the running time of MPMND and the plug-in method on real datasets for the AR, AM, and F measures. Note that the plug-in method learns the same logistical model on the training dataset for all performance measures, so we average its running time. The results show that our MPMND method is faster than the plug-in method on most of the datasets. Table V presents the corresponding reported performance measures which show that our method achieves a comparable performance with the state-of-the-art plug-in method (see Table V).

We consider the F1-measure metric. Fig. 1 presents the value of \( Q_F \) obtained for several different imbalanced datasets. The horizontal axis represents the number of iterations. From Fig. 1 it is seen that the objective \( Q_F \) converges quickly, which is consistent with Theorem 1. Fig. 2 shows the reported value of \( \alpha_P \) and \( \alpha_N \) as a function of iterations. It is shown that both \( \alpha_P \) and \( \alpha_N \) also converge. After fixing \( \alpha_P \), we update the classifier by solving a concave optimization problem (14). As \( \alpha_P \) converges, the classifier is also convergent.

We now compare our MPMF method with several state-of-the-art imbalanced learning algorithms, including the over-sampling methods: random over-sampling (ROS) [7], synthetic minority over-sampling technique (SMOTE) [5], ADAptive SYNthetic sampling (ADASYN) [41]), the under-sampling methods: random under-sampling (RUS) [7], cluster centroids (CC) [42], and the ensemble learning methods: balanced random forest classifier (BRF) [43] and RUSBoost [44]. These compared algorithms are implemented with the help of an open-source toolbox Imbalanced-learn [42] and the package Scikit-learn [45]. For the resampling methods, we train a

---

**Table IV**

| DATASET   | AR       | AM       | QM       | HM       | GM       | GTP       | JAC/F1  | F2       | PLUG-IN  |
|-----------|----------|----------|----------|----------|----------|-----------|---------|----------|----------|
| LETTER    | 0.0222   | 0.1393   | 0.1578   | 0.1952   | 0.2008   | 0.2702    | 0.1878  | 0.1923   | 0.1419   |
| BREAST    | 0.0254   | 0.0341   | 0.0290   | 0.0614   | 0.0676   | 0.0980    | 0.0550  | 0.0981   | 0.0021   |
| SEGMENT   | 0.1081   | 0.2321   | 0.2178   | 0.3425   | 0.3489   | 0.3576    | 0.2984  | 0.3236   | 0.0280   |
| USPS      | 0.8704   | 1.3123   | 1.2332   | 1.4355   | 1.4179   | 1.5656    | 1.4405  | 1.4754   | 24.0888  |
| CVTYPE    | 0.0225   | 0.4994   | 0.2762   | 0.7239   | 1.1183   | 0.4069    | 0.3504  | 0.3756   | 9.4688   |
| IJCNN     | 0.0121   | 0.0770   | 0.0541   | 0.0810   | 0.0851   | 0.1186    | 0.1082  | 0.0866   | 0.3404   |
| SKIN      | 0.2357   | 0.1867   | 0.1489   | 0.2044   | 0.2330   | 0.3842    | 0.2455  | 0.2717   | 0.6936   |
| SENSORLESS| 0.0668   | 0.4258   | 0.4552   | 1.0971   | 0.9714   | 0.5558    | 0.5073  | 0.5847   | 7.9259   |
| MNIST     | 0.1681   | 0.1704   | 0.1740   | 0.1866   | 0.1823   | 0.1966    | 0.1700  | 0.1776   | 0.7285   |

**Table V**

| Dataset   | AR       | AM       | QM       | HM       | GM       | GTP       | JAC/F1  | F2       | PLUG-IN  | G-TP/PR  | JAC      | F1       | F2       |
|-----------|----------|----------|----------|----------|----------|-----------|---------|----------|----------|----------|----------|----------|----------|----------|
| LETTER    | 0.9705   | 0.9750   | 0.8994   | 0.8845   | 0.9878   | 0.9830    | 0.9007  | 0.8829   | 0.9025   | 0.8943   | 0.9528   | 0.8795   | 0.8565   |
| BREAST    | 0.9781   | 0.9639   | 0.9813   | 0.9664   | 0.9996   | 0.9985    | 0.9798  | 0.9663   | 0.9820   | 0.9443   | 0.9526   | 0.8795   | 0.8565   |
| SEGMENT   | 0.9570   | 0.9684   | 0.9461   | 0.9532   | 0.9902   | 0.9939    | 0.9438  | 0.9528   | 0.9443   | 0.9526   | 0.8795   | 0.8565   |
| USPS      | 0.9701   | 0.9479   | 0.9452   | 0.9277   | 0.9962   | 0.9911    | 0.9449  | 0.9242   | 0.9455   | 0.9260   | 0.8795   | 0.8565   |
| CVTYPE    | 0.7404   | 0.7722   | 0.7600   | 0.7695   | 0.9465   | 0.9463    | 0.7689  | 0.7683   | 0.7694   | 0.7689   | 0.8271   | 0.8339   |
| IJCNN     | 0.9103   | 0.9238   | 0.8248   | 0.8363   | 0.9699   | 0.9712    | 0.9571  | 0.9840   | 0.9645   | 0.9492   | 0.9840   | 0.9840   |
| SKIN      | 0.9437   | 0.9259   | 0.9650   | 0.9504   | 0.9976   | 0.9954    | 0.9633  | 0.9840   | 0.9645   | 0.9492   | 0.9840   | 0.9840   |
| SENSORLESS| 0.9305   | 0.9356   | 0.8969   | 0.8628   | 0.9753   | 0.9687    | 0.8771  | 0.8573   | 0.8841   | 0.8594   | 0.8841   | 0.8594   |
| MNIST     | 0.9835   | 0.9881   | 0.9799   | 0.9809   | 0.9996   | 0.9996    | 0.9798  | 0.9809   | 0.9799   | 0.9809   | 0.9799   | 0.9809   |

The plug-in method, which first learns a classification probability function by training a logistical regression model on a training dataset and then decides a threshold based on another dataset, is consistent with several nondecomposable performance measures. Table IV gives the running time of MPMND and the plug-in method on real datasets for the AR, AM, and F measures. Note that the plug-in method learns the same logistical model on the training dataset for all performance measures, so we average its running time. The results show that our MPMND method is faster than the plug-in method on most of the datasets. Table V presents the corresponding reported performance measures which show that our method achieves a comparable performance with the state-of-the-art plug-in method (see Table V).

We consider the F1-measure metric. Fig. 1 presents the value of \( Q_F \) obtained for several different imbalanced datasets. The horizontal axis represents the number of iterations. From Fig. 1 it is seen that the objective \( Q_F \) converges quickly, which is consistent with Theorem 1. Fig. 2 shows the reported value of \( \alpha_P \) and \( \alpha_N \) as a function of iterations. It is shown that both \( \alpha_P \) and \( \alpha_N \) also converge. After fixing \( \alpha_P \), we update the classifier by solving a concave optimization problem (14). As \( \alpha_P \) converges, the classifier is also convergent.

We now compare our MPMF method with several state-of-the-art imbalanced learning algorithms, including the over-sampling methods: random over-sampling (ROS) [7], synthetic minority over-sampling technique (SMOTE) [5], ADAptive SYNthetic sampling (ADASYN) [41]), the under-sampling methods: random under-sampling (RUS) [7], cluster centroids (CC) [42], and the ensemble learning methods: balanced random forest classifier (BRF) [43] and RUSBoost [44]. These compared algorithms are implemented with the help of an open-source toolbox Imbalanced-learn [42] and the package Scikit-learn [45]. For the resampling methods, we train a
linear SVM on the new datasets. Noting that the ensemble methods learn a nonlinear classifier, we also train a KMPMF model. For computational effectiveness, we randomly choose 200 positive samples and 200 negative samples as the kernel support vectors, respectively (for the dataset Breast, we choose 100 positive samples and 100 negative samples). Table VI

Fig. 1. Value of $Q_F$ as a function of iterations. As the number of iterations increases, $Q_F$ monotonically decreases and converges. (a) LETTER. (b) BREAST. (c) SEGMENT. (d) USPS. (e) COVTYPE. (f) IJCNN. (g) SKIN. (h) SENSORLESS.
Fig. 2. Value of $\alpha_P$ (red line) and $\alpha_N$ (blue line) obtained as a function of iterations. Both of them are convergent for each dataset. (a) LETTER. (b) BREAST. (c) SEGMENT. (d) USPS. (e) COVTYPE. (f) IJCNN. (g) SKIN. (h) SENSORLESS.

gives the running time (in seconds) of the algorithms used for imbalanced classification problems. In most cases, MPMF is faster than the compared imbalanced learning algorithms, especially for large-scale datasets. This means that our MPMF method scales well to deal with large-scale imbalanced classification problems. Table VII presents the obtained values of
the F1-measure. From the experimental results, it can be seen that our linear MPMF model has a better performance than the compared resampling methods and our nonlinear model KMPMF achieves a comparable result for the F1-measure with the compared ensemble methods.

VII. Conclusion

In many real-world problems, only the mean and covariance matrix but not the true distribution of data are known in advance. To address this issue, [33] proposed the minimax probability machine (MPM) based on the mean and covariance matrix of data and the AR. However, AR is not an appropriate metric for imbalanced classification problems. In this article, we extended MPM to deal with imbalance classification problems based on some nondecomposable performance measures including the \( F_\beta \) measure for the first time. To solve the new model effectively, we derived its equivalent minimization formulation in terms of a polynomial combination of FPR and FNR, which is then solved by using the alternating descent method. The kernel trick is also used to obtain a nonlinear \( F_\beta \) maximization. The advantage of our method is that it makes use of only the mean and covariance matrix of data and thus is independent of the training data size, so our method is also appropriate for large-scale problems. In addition, our method has no hyper-parameters to choose from. Experiments on both the synthetic dataset and real-world benchmark datasets have been presented to illustrate the effectiveness of our method.

It is noted that certain online learning algorithms have also been studied recently for the \( F_\beta \)-measure [46]–[48] to handle large-scale problems. As an ongoing project, we are currently developing certain online learning algorithms based on nondecomposable performance measures including the \( F_\beta \)-measure which can deal with imbalanced large-scale problems. In addition, deep networks have recently been applied to nondecomposable measure maximization problems [18], [49]. Note that our method is not very efficient in dealing with very high-dimensional imbalanced classification problems due to the large storage requirement to store the estimator of the true covariance matrix. One way to overcome this difficulty is to use a dimension reduction method such as PCA to reduce the feature dimensions of the data.

REFERENCES

[1] D. Musicant, V. Kumar, and A. Ozgur, “Optimizing F-measure with support vector machines,” in Proc. 16th Int. Florida Artif. Intell. Soc. Conf., 2003, pp. 356–360.
[2] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier, “An exact algorithm for F-measure maximization,” in Proc. Adv. Neural Inf. Process. Syst., Granada, Spain, Dec. 2011, pp. 1404–1412.
[3] A. Menon, H. Narasimhan, S. Agrawal, and S. Chawla, “On the statistical consistency of algorithms for binary classification under class imbalance,” in Proc. Int. Conf. Mach. Learn., Atlanta, CA, USA, Jun. 2013, pp. 603–611.
[4] S. Parambath, N. Usunier, and Y. Grandvalet, “Optimizing F-measures by cost-sensitive classification,” in Proc. Adv. Neural Inf. Process. Syst., Montreal, QC, Canada, Dec. 2014, pp. 2123–2131.
[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” J. Artif. Intell. Res., vol. 16, no. 1, pp. 321–357, 2002.
[6] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, “Cost-sensitive learning of deep feature representations from imbalanced data,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 9, pp. 4152–4165, Sep. 2018.
[9] B. Krawczyk, M. Kozierski, and M. Wozniak, “Radial-based oversampling for multiclass imbalanced data classification,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2818–2831, Aug. 2020.

[10] J. Mathew, C. K. Pang, M. Luo, and W. H. Leong, “Classification of imbalanced data by oversampling in kernel space of support vector machines,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4065–4076, Sep. 2018.

[11] J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance,” *J. Big Data*, vol. 6, no. 1, pp. 4152–4165, Dec. 2019.

[12] G. Haixiang, “A maximum expected utility framework for binary classification,” in *Adv. Neural Inf. Process. Syst.*, vol. 31, pp. 4066–4076, 2018.

[13] G. Haixiang, “A maximum expected utility framework for binary classification,” in *Adv. Neural Inf. Process. Syst.*, vol. 31, pp. 4066–4076, 2018.

[14] J. Hu, H. Yang, M. Lyu, I. King, and A. So, “Online nonlinear AUC maximization for imbalanced data sets,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1253–1266, Apr. 2018.

[15] A. Cano, A. Zafría, and S. Ventura, “Weighted data classification methods for imbalanced data,” *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1672–1687, Dec. 2013.

[16] Z. Zhu, Z. Wang, D. Li, Y. Zhu, and W. Du, “Geometric structural ensemble learning for imbalanced problems,” *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1671–1629, Apr. 2020.

[17] K. Bascol, R. Emonet, E. Fromont, A. Habrard, G. Metzler, and M. Sebban, “From cost-sensitive classification to tight F-measure bounds,” in *Proc. Int. Conf. Artif. Intell. Statist.*, Okinawa, Japan, 2019, pp. 1245–1253.

[18] A. Sanyal, P. Kumar, P. Kar, S. Chawla, and F. Sebastiani, “Optimizing non-decomposable measures with deep networks,” *Mach. Learn.*, vol. 107, nos. 8–10, pp. 1597–1620, Jul. 2018.

[19] N. Natarajan, O. Koyejo, P. Ravikumar, and I. S. Dhillon, “Consistent binary classification with generalized performance measures,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2744–2752.

[20] E. Eban, M. Schain, A. Mackey, A. Gordon, R. Saurous, and G. Elidan, “Scalable learning of non-decomposable objectives,” in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, Fort Lauderdale, FL, USA, Apr. 2017, pp. 832–840.

[21] H. Narasimhan, A. Cotter, and M. Gupta, “Optimizing generalized rate metrics with three players,” in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2019, pp. 10747–10758.

[22] N. Ye, K. Chai, W. Lee, and H. Chieu, “Optimizing F-measures: A tale of two approaches,” in *Proc. Int. Conf. Mach. Learn.*, Edinburgh, Scotland, Jun. 2012, pp. 1555–1562.

[23] K. Dembczyński, W. Kotłowski, O. Koyejo, and N. Natarajan, “Consistency analysis for binary classification revisited,” in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, Aug. 2017, pp. 961–969.

[24] M. Jansche, “Maximum expected F-measure training of logistic regression models,” in *Proc. Conf. Human Lang. Technol. Empirical Methods Natural Lang. Process. (HLT)*, 2005, pp. 692–699.

[25] T. Joachims, “A support vector method for multivariate performance measures,” in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, Bonn, Germany, Aug. 2005, pp. 377–384.

[26] P. M. Chinta, P. Balamurugan, S. Shevade, and M. N. Murty, “Optimizing F-measure with non-convex loss and sparse linear classifiers,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Dallas, TX, USA, Aug. 2013, pp. 2858–2865.

[27] W. Kotłowski and K. Dembczyński, “Surrogate regret bounds for generalized classification performance metrics,” *Mach. Learn.*, vol. 106, no. 4, pp. 549–572, Apr. 2017.

[28] H. Narasimhan, R. Vaish, and S. Agarwal, “On the statistical consistency of plug-in classifiers for non-decomposable performance measures,” in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 1493–1501.

[29] D. D. Lewis, “Evaluating and optimizing autonomous text classification systems,” in *Proc. 18th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 1995, pp. 246–254.

[30] M. Jansche, “A maximum expected utility framework for binary sequence labeling,” in *Proc. Joint Meet. Assoc. Comput. Linguistics, Proc. Int. Conf. Déc. Mach. Learn.*, Jun. 2007, pp. 737–743.

[31] W. Waegeman, K. Dembczyński, A. Jachnik, W. Cheng, and E. Hüllermeier, “On the Bayes-optimality of F-measure maximizers,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3333–3388, Jan. 2014.

[32] N. Natarajan, O. Koyejo, P. Ravikumar, and I. S. Dhillon, “Optimal decision-theoretic classification using non-decomposable performance metrics,” 2015, arXiv:1505.01802. [Online]. Available: http://arxiv.org/abs/1505.01802

[33] G. R. Gan Lankriet, L. El Ghaoui, C. Bhattacharya, and M. I. Jordan, “A robust minimax approach to classification,” *J. Mach. Learn. Res.*, vol. 3, pp. 555–582, Dec. 2002.

[34] K. Huang, H. Yang, I. King, M. Lyu, and L. Chan, “The minimum error minimax probability machine,” *J. Mach. Learn. Res.*, vol. 5, pp. 1253–1286, Oct. 2004.

[35] A. W. Marshall and I. Olkin, “Multivariate Chebyshev inequalities,” *Ann. Math. Statist.*, vol. 31, no. 4, pp. 1001–1014, Dec. 1960.

[36] D. Bertsimas and I. Popescu, “Optimal inequalities in probability theory: A convex optimization approach,” *SIAM J. Optim.*, vol. 15, no. 3, pp. 780–804, 2005.

[37] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[38] K. Huang, H. Yang, I. King, M. Lyu, and L. Chan, “Biased minimax probability machine for medical diagnosis,” in *Proc. Int. Symp. Artif. Intell. Math., Tallahassee, FL, USA*, Jan. 2004, pp. 1–8.

[39] K. Huang, H. Yang, I. King, and M. R. Lyu, “Imbalanced learning with a biased minimax probability machine,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 4, pp. 913–923, Aug. 2006.

[40] S. Cousins and J. Shawe-Taylor, “High-probability minimax probability machines,” *Mach. Learn.*, vol. 106, no. 6, pp. 863–886, Jun. 2017.

[41] H. He, Y. Bai, E. A. Garcia, and S. Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intelligence)*, Jun. 2008, pp. 1322–1328.

[42] P. Lopez, E. Almeida, P. L. Neto, and D. L. Costa, “Imbalanced learning: A survey,” *IEEE Trans. Syst., Man, Cybern. C, Appl. Syst.*, vol. 43, no. 6, pp. 780–784, Sep. 2013.

[43] G. Lemaire, F. Nogueira, and C. Aridas, “Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning,” *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 559–563, 2017.
Hong Qiao (Fellow, IEEE) received the B.Eng. degree in hydraulics and control and the M.Eng. degree in robotics from Xian Jiaotong University, Xi’an, China, in 1986 and 1989, respectively, the M.Phil. degree in robotics control from the Industrial Control Center, University of Strathclyde, Glasgow, U.K., in 1993, and the Ph.D. degree in robotics and artificial intelligence from De Montfort University, Leicester, U.K., in 1995. She was a University Research Fellow with De Montfort University from 1995 to 1997, a Research Assistant Professor and an Assistant Professor with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong, from 1997 to 2002, and a Lecturer with School of Informatics, University of Manchester, Manchester, U.K., from 2002 to 2008. She is currently a Professor with the State Key Lab of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

Dr. Qiao is currently a member of Administrative Committee of the IEEE Robotics and Automation Society (RAS), the IEEE Medal for Environmental and Safety Technologies Committee, and Early Career Award Nomination Committee, Most Active Technical Committee Award Nomination Committee, and Industrial Activities Board for RAS. She is currently Associate Editors of the IEEE TRANSACTION ON CYBERNETICS, the IEEE TRANSACTION ON AUTOMATION SCIENCE AND ENGINEERING, and an Editor-in-Chief of Assembly Automation.

Bo Zhang (Member, IEEE) received the B.Sc. degree in mathematics from Shandong University, Jinan, China, in 1983, the M.Sc. degree in mathematics from Xi’an Jiaotong University, Xi’an, China, in 1985, and the Ph.D. degree in applied mathematics from the University of Strathclyde, Glasgow, U.K., in 1992. From 1985 to 1988, he was a Lecturer with the Department of Mathematics, Xi’an Jiaotong University. He was a Post-Doctoral Research Fellow with the Department of Mathematics, Keele University, Keele, U.K., from January 1992 to December 1994, and with the Department of Mathematical Sciences, Brunel University, Uxbridge, U.K., from January 1995 to October 1997. In November 1997, he joined the School of Mathematical and Informational Sciences, Coventry University, Coventry, U.K., as a Senior Lecturer, where he was promoted to Reader in applied mathematics in September 2000 and a Professor of applied mathematics in September 2003. He is currently a Professor with the Institute of Applied Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China. His current research interests include wave propagation and scattering, inverse scattering and imaging, machine learning, and data mining.

Dr. Zhang was an Associate Editor of the IEEE TRANSACTION ON CYBERNETICS. He is currently an Associate Editor of Applicable Analysis.