Biased Minimax Probability Machine for Medical Diagnosis

Kaizhu Huang, Haiqin Yang, Irwin King, Michael R. Lyu and Laiwan Chan
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, New Territories, Hong Kong
{kzhuang, hqyang, king, lyu, lwchan}@cse.cuhk.edu.hk

Abstract

The Minimax Probability Machine (MPM) constructs a classifier, which provides a worst-case bound on the probability of misclassification of future data points based on reliable estimates of means and covariance matrices of the classes from the training data points, and achieves the comparative performance with a state-of-the-art classifier, the Support Vector Machine. In this paper, we eliminate the assumption of the unbiased weight for each class in the MPM and develop a critical extension, named Biased Minimax Probability Machine (BMPM), to deal with biased classification tasks, especially in the medical diagnostic applications. We outline the theoretical derivatives of the BMPM. Moreover, we demonstrate that this model can be transformed into a concave-convex Fractional Programming (FP) problem or a pseudoconcave problem. After illustrating our model with a synthetic dataset and applying it to the real-world medical diagnosis datasets, we obtain encouraging and promising experimental results.

1. Introduction

Biased classifiers have many applications, including the medical diagnostic applications. The goal of constructing a two-category biased classifier is to make the accuracy of the important class, instead of the overall accuracy, as high as possible, while maintaining the accuracy of the less important class at an acceptable level. For some biased classifiers, e.g., the weighted Support Vector Machine [8], it is often hard to quantitatively evaluate how the weight will affect the classification. Recently, a novel classification model, Minimax Probability Machine (MPM) [4], provides a worst-case bound on the probability of misclassification of future data points based on reliable estimates of means and covariance matrices of the classes from the training data points and achieves the comparative performance with a state-of-the-art classifier, the Support Vector Machine [12].

In this paper, by eliminating the assumption of the unbiased weight for each class in the MPM, we develop a critical extension, Biased Minimax Probability Machine (BMPM), to deal with biased classification tasks. This model is transformed into a concave-convex Fractional Programming (FP) [10] problem or a pseudoconcave problem with every local maximum being global maximum. Moreover, as far as we know, this model is the first quantitative method to control how the decision hyperplane moves in favor of the classification of the more important class.

The paper is organized as follows. In Section 2, we present the linear Biased Minimax Probability Machine while reviewing the original MPM model. In Section 3, we kernelize the linear Biased Minimax Probability Machine and propose a feasible solving method to extend its application into the non-linear classification tasks. In Section 4, we illustrate our model with a synthetic dataset and apply it to real-world medical diagnosis datasets. Finally, we conclude the paper in Section 5.

2. The Linear Optimal Biased Probabilistic Decision Hyperplane

In this section, we present the linear biased minimax framework while reviewing the original MPM.

Suppose two random vectors $\mathbf{x}$ and $\mathbf{y}$ represent two classes of data with means and covariance matrices as $\{\bar{\mathbf{x}}, \Sigma_{\mathbf{x}}\}$, $\{\bar{\mathbf{y}}, \Sigma_{\mathbf{y}}\}$ respectively in a two-category classification task, where $\mathbf{x}$, $\mathbf{y}$, $\bar{\mathbf{x}}$, $\bar{\mathbf{y}} \in \mathbb{R}^n$, and $\Sigma_{\mathbf{x}}$, $\Sigma_{\mathbf{y}} \in \mathbb{R}^{n\times n}$. For convenience, we also use $\mathbf{x}$ and $\mathbf{y}$ to represent the corresponding class of the $\mathbf{x}$ data and the $\mathbf{y}$ data respectively.

With reliable estimations of $\{\bar{\mathbf{x}}, \Sigma_{\mathbf{x}}\}$, $\{\bar{\mathbf{y}}, \Sigma_{\mathbf{y}}\}$ for two classes of data, Minimax Probability Machine attempts to determine the hyperplane $\mathbf{a}^T\mathbf{z} = b$ ($\mathbf{a} \neq \mathbf{0}$, $\mathbf{z} \in \mathbb{R}^n$, $b \in \mathbb{R}$, superscript T denotes the transpose) which can separate two classes data with a maximal probability. The formulation for the original model is written as follows:

$$\max_{\alpha,b,\mathbf{a}\neq\mathbf{0}} \quad \alpha \quad \text{s.t.} \qquad \inf P_r\{\mathbf{a}^T\mathbf{x} \geq b\} \geq \alpha \ ,$$
$$\inf P_r\{\mathbf{a}^T\mathbf{y} \leq b\} \geq \alpha \ ,$$

where α represents the lower bound of the accuracy for the future data, or the worst-case accuracy. Future points, $\mathbf{z}$ for which $\mathbf{a}^T\mathbf{z} > b$ are then classified as the class $\mathbf{x}$; otherwise, they are judged as the class $\mathbf{y}$. This derived decision hyperplane is claimed to minimize the worst-case (maximum) probability of misclassification, or the error rate, for the classification of future data points. Furthermore, the MPM problem can be transformed into a convex optimization problem, more specifically, a Second Order Cone Programming problem [6] [7].

In this model, it assumes an unbiased weight for two classes, i.e., it forces the probabilities for the class $\mathbf{x}$ and the class $\mathbf{y}$ to be an equal value α. However, in real-world applications, the importance for two classes is not always the same, which implies that the corresponding two probabilities are not necessarily equal. Motivated by this point, we propose the following Biased Minimax Probability Machine (BMPM) formulation:

$$\max_{\alpha,\beta,b,\mathbf{a}\neq\mathbf{0}} \quad \alpha \quad \text{s.t.} \qquad \inf P_r\{\mathbf{a}^T\mathbf{x} \geq b\} \geq \alpha \ , \tag{1}$$
$$\inf P_r\{\mathbf{a}^T\mathbf{y} \leq b\} \geq \beta \ , \tag{2}$$
$$\beta \geq \gamma \ , \tag{3}$$

where γ is a pre-specified positive constant, which represents an acceptable accuracy level for the less important class.

This optimization will maximize the accuracy (the probability α) for the biased class $\mathbf{x}$ while maintaining the other class $\mathbf{y}$'s accuracy at an acceptable level by setting a lower bound as (3). The hyperplane $\mathbf{a}^{*T}\mathbf{z} = b^*$ given by the solution of this optimization will favor the classification of the important class $\mathbf{x}$ over the less important class $\mathbf{y}$ and will be more suitable in handling biased classification tasks.

In the following, we propose to solve this optimization problem. First, we borrow Lemma 1 from [5].

Lemma 1: Given $\mathbf{a} \neq \mathbf{0}$, b such that $\mathbf{a}^T\mathbf{y} \leq b$ and $\beta \in [0, 1)$, the condition

$$\inf P_r\{\mathbf{a}^T\mathbf{y} \leq b\} \geq \beta,$$

holds if and only if $\quad b - \mathbf{a}^T\bar{\mathbf{y}} \geq \kappa(\beta)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}} \quad$ with $\quad \kappa(\beta) = \sqrt{\frac{\beta}{1-\beta}}$.

This lemma can be proved by using the Lagrange multiplier method and the following theory developed in [9]:

$$\sup_{\mathbf{y}\in\{\bar{\mathbf{y}},\Sigma_{\mathbf{y}}\}} Pr\{\mathbf{a}^T\mathbf{y} \geq b\} = \frac{1}{1+d^2}, \quad \text{with} \quad d^2 = \inf_{\mathbf{a}^T\mathbf{y}\geq\mathbf{b}} (\mathbf{y}-\bar{\mathbf{y}})^T\Sigma_{\mathbf{y}}^{-1}(\mathbf{y}-\bar{\mathbf{y}}).$$

Details about the proof can be seen in [5].

By using Lemma 1, we obtain the following transformed optimization problem:

$$\max_{\alpha,\beta,b,\mathbf{a}\neq\mathbf{0}} \quad \alpha \quad \text{s.t.} \qquad -b + \mathbf{a}^T\bar{\mathbf{x}} \geq \kappa(\alpha)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{x}}\mathbf{a}}, \tag{4}$$
$$b - \mathbf{a}^T\bar{\mathbf{y}} \geq \kappa(\beta)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}}, \tag{5}$$
$$\beta \geq \gamma, \tag{6}$$

where $\kappa(\alpha) = \sqrt{\frac{\alpha}{1-\alpha}}$, $\kappa(\beta) = \sqrt{\frac{\beta}{1-\beta}}$. Equation (5) is directly obtained from (2) by using Lemma 1. Similarly, by changing $\mathbf{a}^T\mathbf{x} \geq b$ to $\mathbf{a}^T(-\mathbf{x}) \leq -b$, (4) is obtained from (1). From (4) and (5), we get:

$$\mathbf{a}^T\bar{\mathbf{y}} + \kappa(\beta)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}} \leq b \leq \mathbf{a}^T\bar{\mathbf{x}} - \kappa(\alpha)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{x}}\mathbf{a}}\ . \tag{7}$$

If we eliminate b from this inequality, we obtain

$$\mathbf{a}^T(\bar{\mathbf{x}} - \bar{\mathbf{y}}) \geq \kappa(\alpha)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{x}}\mathbf{a}} + \kappa(\beta)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}}. \tag{8}$$

We observe the magnitude of $\mathbf{a}$ will not influence the solution of (8). Without loss of generality, we can set $\mathbf{a}^T(\bar{\mathbf{x}} - \bar{\mathbf{y}}) = 1$. In addition, since $\kappa(\alpha)$ increases monotonously with α, maximizing α is equivalent to maximizing $\kappa(\alpha)$. Thus the problem can be further modified to

$$\max_{\alpha,\beta,\mathbf{a}\neq\mathbf{0}} \quad \kappa(\alpha) \quad \text{s.t.} \qquad 1 \geq \kappa(\alpha)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{x}}\mathbf{a}} + \kappa(\beta)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}}, \tag{9}$$

$$\mathbf{a}^T(\bar{\mathbf{x}} - \bar{\mathbf{y}}) = 1, \tag{10}$$

$$\kappa(\beta) \geq \kappa(\gamma), \tag{11}$$

where (11) is equivalent to (6) due to the monotone property of κ function.

Lemma 2: The maximum value of $\kappa(\alpha)$ under the constraints of (9) (10) (11) is achieved when the right hand side of (9) is strictly equal to 1.

Proof: Assume the maximum is achieved when $1 > \kappa(\beta)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}} + \kappa(\alpha)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{x}}\mathbf{a}}$. A new solution constructed by increasing $\kappa(\alpha)$ with a small positive amount and maintaining $\kappa(\beta)$, $\mathbf{a}$ unchanged will satisfy the constraints and will be a better solution. ■

Moreover, $\mathbf{\Sigma_x}$ and $\mathbf{\Sigma_y}$ can be considered as positive definite matrices[1]. Therefore, we obtain $\kappa(\alpha) = \frac{1-\kappa(\beta)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}}}{\sqrt{\mathbf{a}^T\Sigma_{\mathbf{x}}\mathbf{a}}}$ according to Lemma 2. Obviously, this optimization function is a linear function with respect to $\kappa(\beta)$ and $\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}}$ is a positive term; therefore, this optimization function is maximized when $\kappa(\beta)$ is set to its lower bound $\kappa(\gamma)$. Thus, the BMPM optimization problem can be changed and written into the so-called Fractional Programming (FP) problem [10] as:

$$\max_{\mathbf{a}\neq\mathbf{0}} \quad \frac{f(\mathbf{a})}{g(\mathbf{a})}\ , \quad \text{s.t.} \quad \mathbf{a} \in A = \{\mathbf{a}|\mathbf{a}^T(\bar{\mathbf{x}} - \bar{\mathbf{y}}) = 1\}\ , \tag{12}$$

where $f(\mathbf{a}) = 1 - \kappa(\gamma)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}}$, $\quad g(\mathbf{a}) = \sqrt{\mathbf{a}^T\Sigma_{\mathbf{x}}\mathbf{a}}$. In the following, we propose Lemma 3 to show that this FP problem is solvable.

Lemma 3: The Fractional Programming problem (12) is a strictly quasiconcave problem and is thus solvable.

Proof: It is easy to see that the domain A is a convex set on $\mathbb{R}^n$, $f(\mathbf{a})$ and $g(\mathbf{a})$ are differentiable on A. Moreover, since $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{y}}$ can be both considered as positive definite matrices, $f(\mathbf{a})$ is a concave function on A and $g(\mathbf{a})$ is a convex function on A. Then $\frac{f(\mathbf{a})}{g(\mathbf{a})}$ is a concave-convex FP or a pseudoconcave problem. Hence it is strictly quasiconcave on A according to [10]. Therefore, every local maximum is a global maximum [10]. In other words, this Fractional Programming problem is solvable. ■

Many methods can be used to solve this problem. For example, a conjugate gradient method can solve this problem in n (the dimension of the data points) steps if the initial point is suitably assigned [1]. In each step, the computational cost to calculate the conjugate gradient is $O(n^2)$. Thus this method will have a worst-case $O(n^3)$ time complexity. Adding the time cost to estimate $\bar{\mathbf{x}}$, $\bar{\mathbf{y}}$,$\Sigma_{\mathbf{x}}$,$\Sigma_{\mathbf{y}}$, the total cost is

[1]In practice, we can always add a small positive amount to the diagonal elements of these two matrices and make them positive definite.

$O(n^3 + Nn^2)$, where N is the number of the data points. This computational cost is the same order to the Minimax Probability Machine [4] and the linear support vector machine [11].

In this paper, we use Rosen gradient projection method [1] to find the solution of this concave-convex FP problem, which is proved to converge to a local maximum with a worse-case linear convergence rate. Moreover, the local maximum will be exactly the global maximum in this problem.

From Lemma 2, we can see that the inequalities in (7) will become equalities at the maximum point. The optimal b, denoted by b^*, will thus be obtained by

$$b^* = \mathbf{a}^{*T}\bar{\mathbf{y}} + \kappa(\beta^*)\sqrt{\mathbf{a}^{*T}\Sigma_{\mathbf{y}}\mathbf{a}^*} = \mathbf{a}^{*T}\bar{\mathbf{x}} - \kappa(\alpha^*)\sqrt{\mathbf{a}^{*T}\Sigma_{\mathbf{x}}\mathbf{a}^*}. \tag{13}$$

3. Kernelization

In this section, we first use the kernel trick to find a linear classifier in the feature space, $\mathbb{R}^f$, via mapping the n-dimensional data points into a high-dimensional feature space, where the linear classifier in $\mathbb{R}^f$ corresponds to a nonlinear hyperplane in the original space. Next, we propose a feasible algorithm to solve the kernelized optimization problem.

Let $\{\mathbf{x}_i\}_{i=1}^{N_\mathbf{x}}$ and $\{\mathbf{y}_j\}_{j=1}^{N_\mathbf{y}}$ represent the training data for the class $\mathbf{x}$ and the class $\mathbf{y}$ respectively and be mapped as $\mathbf{x} \to \varphi(\mathbf{x}) \sim (\overline{\varphi(\mathbf{x})}, \mathbf{\Sigma}_{\varphi(\mathbf{x})})$, and $\mathbf{y} \to \varphi(\mathbf{y}) \sim (\overline{\varphi(\mathbf{y})}, \mathbf{\Sigma}_{\varphi(\mathbf{y})})$, where $\varphi : \mathbb{R}^n \to \mathbb{R}^f$ is a mapping function[2]. The corresponding linear classifier in $\mathbb{R}^f$ is $\mathbf{a}^T\varphi(\mathbf{z}) = b$, where $\mathbf{a}$, $\varphi(\mathbf{z}) \in \mathbb{R}^f$ and $b \in \mathbb{R}$. Similarly, the transformed FP optimization in BMPM can be written as:

$$\max_{\mathbf{a}\neq\mathbf{0}} \frac{1-\kappa(\gamma)\sqrt{\mathbf{a}^T\Sigma_{\varphi(\mathbf{y})}\mathbf{a}}}{\sqrt{\mathbf{a}^T\Sigma_{\varphi(\mathbf{x})}\mathbf{a}}} \quad \text{s.t.} \quad \mathbf{a}^T(\overline{\varphi(\mathbf{x})} - \overline{\varphi(\mathbf{y})}) = 1 \ . \tag{14}$$

To make the kernel work, we need to represent the final decision hyperplane and the optimization into a kernel form, $K(\mathbf{z}_1, \mathbf{z}_2) = \varphi(\mathbf{z}_1)^T\varphi(\mathbf{z}_2)$, namely an inner product form of the mapping data points.

We reformulate the optimization and the decision hyperplane as the kernel form in the following.

Let $\mathbf{a} = \mathbf{a_p} + \mathbf{a_v}$, where $\mathbf{a_p}$ is the projection of $\mathbf{a}$ in the space spanned by all the training data, i.e., $\{\varphi(\mathbf{x}_i)\}_{i=1}^{N_\mathbf{x}}$ and $\{\varphi(\mathbf{y}_j)\}_{j=1}^{N_\mathbf{y}}$ and $\mathbf{a_v}$ is the orthogonal component of $\mathbf{a}$ in this span space, the component $\mathbf{a_v}$ will be observed to vanish in the optimization (14) by using ${\mathbf{a_v}}^T\varphi(\mathbf{x}_i) = 0$ and ${\mathbf{a_v}}^T\varphi(\mathbf{y}_j) = 0$. This implies that the optimal $\mathbf{a}$ is in the space spanned by all the training data and thus can be written as a linear combination form of the training data, i.e.,

$$\mathbf{a} = \sum_{i=1}^{N_\mathbf{x}} \mu_i\varphi(\mathbf{x}_i) + \sum_{j=1}^{N_\mathbf{y}} \upsilon_j\varphi(\mathbf{y}_j) \ , \tag{15}$$

where the coefficients μ_i, $\upsilon_j \in \mathbb{R}, i = 1, \ldots, N_\mathbf{x}$ and $j = 1, \ldots, N_\mathbf{y}$.

Substituting (15) and the following four plug-in estimated parameters $\overline{\varphi(\mathbf{x})} = \frac{1}{N_\mathbf{x}}\sum_{i=1}^{N_\mathbf{x}} \varphi(\mathbf{x}_i)$, $\overline{\varphi(\mathbf{y})} = \frac{1}{N_\mathbf{y}}\sum_{j=1}^{N_\mathbf{y}} \varphi(\mathbf{y}_j)$, $\Sigma_{\varphi(\mathbf{x})} = \frac{1}{N_\mathbf{x}}\sum_{i=1}^{N_\mathbf{x}}(\varphi(\mathbf{x}_i) - \overline{\varphi(\mathbf{x})})(\varphi(\mathbf{x}_i) - \overline{\varphi(\mathbf{x})})^T$, $\Sigma_{\varphi(\mathbf{y})} = \frac{1}{N_\mathbf{y}}\sum_{j=1}^{N_\mathbf{y}}(\varphi(\mathbf{y}_j) - \overline{\varphi(\mathbf{y})})(\varphi(\mathbf{y}_j) - \overline{\varphi(\mathbf{x})})^T$ into the optimization problem (14), we can obtain a kernelized version:

$$\max_{\mathbf{w}\neq\mathbf{0}} \frac{1-\kappa(\gamma)\sqrt{\frac{1}{N_\mathbf{y}}\mathbf{w}^T\tilde{\mathbf{K}}_\mathbf{y}^T\tilde{\mathbf{K}}_\mathbf{y}\mathbf{w}}}{\sqrt{\frac{1}{N_\mathbf{x}}\mathbf{w}^T\tilde{\mathbf{K}}_\mathbf{x}^T\tilde{\mathbf{K}}_\mathbf{x}\mathbf{w}}} \quad \text{s.t.} \quad \mathbf{w}^T(\tilde{\mathbf{k}}_\mathbf{x} - \tilde{\mathbf{k}}_\mathbf{y}) = 1 \ . \tag{16}$$

In (16), $\mathbf{w} = [\mu_1, \ldots, \mu_{N_\mathbf{x}}, \upsilon_1, \ldots, \upsilon_{N_\mathbf{y}}]^T$ and $\tilde{\mathbf{k}}_\mathbf{x}, \tilde{\mathbf{k}}_\mathbf{y} \in \mathbb{R}^{N_\mathbf{x}+N_\mathbf{y}}$ with

$$[\tilde{\mathbf{k}}_\mathbf{x}]_i = \frac{1}{N_\mathbf{x}}\sum_{j=1}^{N_\mathbf{x}} \mathbf{K}(\mathbf{x}_j, \mathbf{z}_i), \qquad [\tilde{\mathbf{k}}_\mathbf{y}]_i = \frac{1}{N_\mathbf{y}}\sum_{j=1}^{N_\mathbf{y}} \mathbf{K}(\mathbf{y}_j, \mathbf{z}_i),$$

[2]The notation presented in this section largely follows that of [5].

where $\mathbf{z}_i = \mathbf{x}_i$ for $i = 1, 2, \ldots, N_\mathbf{x}$ and $\mathbf{z}_i = \mathbf{y}_{i-N_\mathbf{x}}$ for $i = N_\mathbf{x}+1, N_\mathbf{x}+2, \ldots, N_\mathbf{x}+N_\mathbf{y}$. $\tilde{\mathbf{K}}$ is given by

$$\tilde{\mathbf{K}} = \begin{pmatrix} \tilde{\mathbf{K}}_\mathbf{x} \\ \tilde{\mathbf{K}}_\mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{K}_\mathbf{x} - \mathbf{1}_{N_\mathbf{x}} \tilde{\mathbf{k}}_\mathbf{x}^T \\ \mathbf{K}_\mathbf{y} - \mathbf{1}_{N_\mathbf{y}} \tilde{\mathbf{k}}_\mathbf{y}^T \end{pmatrix} ,$$

where $\mathbf{1}_{N_\mathbf{x}}$ is an $N_\mathbf{x}$-dimension column vector with the values of all elements equal to one and $\mathbf{1}_{N_\mathbf{y}}$ is similarly defined. $N_\mathbf{x}$ and $N_\mathbf{y}$ are the number of the data points for the class $\mathbf{x}$ and $\mathbf{y}$ respectively. $\mathbf{K}_\mathbf{x}$ and $\mathbf{K}_\mathbf{y}$ are the matrices formed by the first $N_\mathbf{x}$ rows and the last $N_\mathbf{y}$ rows of the Gram matrix $\mathbf{K}$, which is defined as $\mathbf{K}_{ij} = \varphi(\mathbf{z}_i)^T \varphi(\mathbf{z}_j)$.

Similarly, the optimal b in the kernelized version, represented by b^*, can be obtained as

$$b^* = \mathbf{w}^{*T} \tilde{\mathbf{k}}_\mathbf{y} + \kappa(\beta^*) \sqrt{\frac{1}{N_\mathbf{y}} \mathbf{w}^{*T} \tilde{\mathbf{K}}_\mathbf{y}^T \tilde{\mathbf{K}}_\mathbf{y} \mathbf{w}^*} = \mathbf{w}^{*T} \tilde{\mathbf{k}}_\mathbf{x} - \kappa(\alpha^*) \sqrt{\frac{1}{N_\mathbf{x}} \mathbf{w}^{*T} \tilde{\mathbf{K}}_\mathbf{x}^T \tilde{\mathbf{K}}_\mathbf{x} \mathbf{w}^*},$$

where $\mathbf{w}^*$, α^*, and β^* are the optimum values given by the above optimization procedure. The kernelized decision hyperplane can be written as

$$f(\mathbf{z}) = \sum_{i=1}^{N_\mathbf{x}} \mathbf{w}_i^* K(\mathbf{z}, \mathbf{x}_i) + \sum_{i=1}^{N_\mathbf{y}} \mathbf{w}_{N_\mathbf{x}+i}^* K(\mathbf{z}, \mathbf{y}_i) - b^* \ .$$

After kernelization, the dimension of the covariance matrices will be the same as the number of the data points, the Rosen Gradient method is not suitable to solve this large-scale optimization problem. We adopt the parametric method [10] to solve the kernelized Fractional Programming problem. Moreover, we still use the unkernelized version to present the algorithm since (16) has a form similar to the unkernelized version of (12). According to the parametric method, the fractional function, $f(\mathbf{a})/g(\mathbf{a})$ can be iteratively optimized in two steps:

Step 1: Find $\mathbf{a}$ by maximizing $f(\mathbf{a}) - \lambda g(\mathbf{a})$ in the domain A, where $\lambda \in \mathbb{R}$ is the newly introduced parameter.

Step 2: Update λ by $\frac{f(\mathbf{a})}{g(\mathbf{a})}$.

According to [10], the maximum of λ, namely, the maximum solution of the FP problem, is guaranteed to converge via a series of the above iterations.

In the following, we adopt a method to solve the maximization problem in Step 1. Replacing $f(\mathbf{a})$ and $g(\mathbf{a})$, we expand the optimization problem as:

$$\max_{\mathbf{a} \neq \mathbf{0}} \quad 1 - \kappa(\gamma) \sqrt{\mathbf{a}^T \Sigma_\mathbf{y} \mathbf{a}} - \lambda \sqrt{\mathbf{a}^T \Sigma_\mathbf{x} \mathbf{a}} \quad \text{s.t.} \quad \mathbf{a}^T (\bar{\mathbf{x}} - \bar{\mathbf{y}}) = 1 \ . \tag{17}$$

Equation (17) is equivalent to $\min_\mathbf{a} \kappa(\gamma) \sqrt{\mathbf{a}^T \Sigma_\mathbf{y} \mathbf{a}} + \lambda \sqrt{\mathbf{a}^T \Sigma_\mathbf{x} \mathbf{a}}$ under the same constraint. By writing $\mathbf{a} = \mathbf{a}_0 + \mathbf{F}\mathbf{u}$, where $\mathbf{a}_0 = (\bar{\mathbf{x}} - \bar{\mathbf{y}}) / \parallel \bar{\mathbf{x}} - \bar{\mathbf{y}} \parallel_2^2$ and $\mathbf{F} \in \mathbb{R}^{n \times (n-1)}$ is an orthogonal matrix whose columns span the subspace of vectors orthogonal to $\bar{\mathbf{x}} - \bar{\mathbf{y}}$, an equivalent form (a factor $\frac{1}{2}$ over each term has been dropped) to remove the constraint can be obtained:

$$\min_{\mathbf{u}, \eta > 0, \xi > 0} \eta + \frac{\lambda^2}{\eta} \parallel \Sigma_\mathbf{x}^{1/2} (\mathbf{a}_0 + \mathbf{F}\mathbf{u}) \parallel_2^2 + \xi + \frac{\kappa(\gamma)^2}{\xi} \parallel \Sigma_\mathbf{y}^{1/2} (\mathbf{a}_0 + \mathbf{F}\mathbf{u}) \parallel_2^2 \ . \tag{18}$$

This optimization form is very similar to the one in Minimax Probability Machine [4] and can also be solved by using an iterative least-squares approach [1] [4].

4. Experiments

In this section, we first illustrate our model with a synthetic dataset. Then we apply it to two real-world medical diagnosis datasets, the breast-cancer dataset and the heart disease dataset.

4.1. A Synthetic Dataset

A two-variable synthetic dataset is generated by the two-dimensional gamma distribution. Two classes of data are generated under the same gamma distribution with the shape and scale parameter $\Gamma(5, 4)$ for the first dimension and $\Gamma(6, 3)$ for the second dimension. To illustrate the algorithm clearly, we transform the data by some displacement and rotation to distinguish the two classes as illustrated in Fig. 1. We assume that the class $\mathbf{x}$ (the more important class) is represented by filled squares (training points) and o's (test points). The other class $\mathbf{y}$ (the less important class) is represented by black +'s (training points) and blue $\times$'s (test points). The acceptance level is assumed to set to 90%. It is clearly observed that the solid line/curve (BMPM for linear/Gaussian kernel) is pushed away from the biased class $\mathbf{x}$ when compared with the corresponding dashed line/curve. This is consistent with the lower bounds in Table I, the corresponding lower bounds for class $\mathbf{x}$ in BMPM are higher than those in MPM. In addition, the test-set accuracies for class $\mathbf{x}$, $\mathrm{TSA}_{\mathbf{x}}$, are significantly increased in BMPM than those in MPM for both the linear and the Gaussian kernel settings. On the other hand, the test-set accuracies for the less important class $\mathbf{y}$, $\mathrm{TSA}_{\mathbf{y}}$, maintain at an acceptable level, i.e., 91.1% and 93.3%, for linear and Gaussian kernel respectively by setting the lower bound to 90.0%. Moreover, the worst-case accuracies given by $\alpha_{\mathbf{x}}$, $\alpha_{\mathbf{y}}$, or α are all smaller than the real test-set accuracies. This clearly demonstrates how the worst-case probability can serve as the quantitative indicator of the classification accuracy of future data points. From Table I, we also observe that the overall test-set accuracies, i.e., TSA, of BMPM are not necessarily lower than those of MPM. An interesting interpretation can be seen in [3].

TABLE I. LOWER BOUND α AND TEST-SET ACCURACY WITH BMPM AND MPM ON THE SYNTHETIC DATASET.

Kernel	BMPM					MPM			
	α		Accuracy			α	Accuracy		
	$\alpha_{\mathbf{x}}$	$\alpha_{\mathbf{y}}$	$\mathrm{TSA}_{\mathbf{x}}$	$\mathrm{TSA}_{\mathbf{y}}$	TSA	α	$\mathrm{TSA}_{\mathbf{x}}$	$\mathrm{TSA}_{\mathbf{y}}$	TSA
Linear(%)	94.9 $\uparrow$	90.0	97.8 $\uparrow$	91.1	94.4	92.7	93.3	95.6	94.4
Gaussian(%)	96.9 $\uparrow$	90.0	97.8 $\uparrow$	93.3	95.6	93.1	93.3	95.6	94.4

4.2. Medical Datasets

The breaset-cancer dataset and the heart disease dataset are obtained from UCI machine learning repository [2]. The breaset-cancer dataset contains 458 instances of the benign class and 241 instances of the malignant class. Each instance is described by 9 attributes. The heart disease dataset includes 120 instances with heart disease, 150 instances without heart disease and each instance is described by 13 attributes. Since handling the missing attribute values is out of the scope of this paper, we remove the instances with missing attribute values in both datasets. In this experiment, the biased class should be the malignant class for the breast-cancer dataset and the heart disease class for the heart disease dataset respectively since misclassifying a patient with a disease into the opposite one may delay the therapy and lead to the aggravation of the disease. Here, we denote $\mathbf{x}$ and $\mathbf{y}$ as the biased class and the less important class respectively.

We evaluate the BMPM algorithm and the MPM algorithm for both datasts. We perform a 5-fold cross validation (CV-5) in both linear and Gaussian kernel settings for both datasets . The kernel parameter σ for the Gaussian kernel $e^{-\|\mathbf{x}-\mathbf{y}\|^2/\sigma}$ is obtained via the cross validation method. For the BMPM algorithm, we set the lower bound accuracy of classifying the less important class to the "pass-level" 50.0%[3] and try to maximize the accuracy of classifying the biased class. The results are shown in Table II and Table III.

[3]We consider the pass-line 50% as an acceptable level for the common persons. The acceptable level can be controlled by real practitioners according to the specific requirements. And we note that, the setting of the lower bound needs to be suitable, since if it is set too high, the maximum value of α may be smaller than this lower bound or even a zero solution will be obtained.

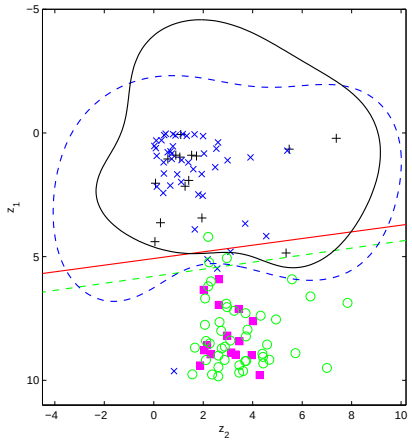

Fig. 1. An example to illustrate Biased Minimax Probability Machine. The solid red line is the decision hyperplane for the linear Biased Minimax Probability Machine while the dashed green line is the decision hyperplane for the linear Minimax Probability Machine. The solid black curve is the decision hyperplane for the Gaussian kernel Biased Minimax Probability Machine, while the dashed blue curve is the decision hyperplane for the Gaussian kernel Minimax Probability Machine. Training points are indicated with magenta filled squares for the class $\mathbf{x}$ and black +'s for the class $\mathbf{y}$. Test points are indicated with green o's for the class $\mathbf{x}$ and blue $\times$'s for the class $\mathbf{y}$. The parameter σ for the Gaussian kernel is searched by the cross validation method. The solid red line and the solid black curve are pushed away from the biased class $\mathbf{x}$ with a qualitative accuracy indicator $\alpha_{\mathbf{x}} = 94.9\%$ and $\alpha_{\mathbf{x}} = 96.9\%$ in Biased Minimax Probability Machine.

TABLE II. LOWER BOUND α AND TEST-SET ACCURACY WITH BMPM AND MPM ON THE BREAST-CANCER DATASET.

Kernel	BMPM					MPM			
	α		Accuracy			α	Accuracy		
	$\alpha_{\mathbf{x}}$	$\alpha_{\mathbf{y}}$	$\text{TSA}_{\mathbf{x}}$	$\text{TSA}_{\mathbf{y}}$	TSA	α	$\text{TSA}_{\mathbf{x}}$	$\text{TSA}_{\mathbf{y}}$	TSA
Linear(%)	$90.0 \pm 0.3\uparrow$	50.0 ± 0.0	$99.9 \pm 0.1\uparrow$	92.0 ± 0.2	94.9 ± 0.2	84.2 ± 0.3	96.9 ± 0.4	97.1 ± 0.5	96.9 ± 0.3
Gaussian(%)	$97.6 \pm 0.3\uparrow$	50.0 ± 0.0	$100.0 \pm 0.0\uparrow$	88.9 ± 0.2	92.8 ± 0.2	90.1 ± 0.3	96.6 ± 0.2	97.1 ± 0.3	96.8 ± 0.2

TABLE III. LOWER BOUND α AND TEST-SET ACCURACY WITH BMPM AND MPM ON THE HEART DISEASE DATASET.

Kernel	BMPM					MPM			
	α		Accuracy			α	Accuracy		
	$\alpha_{\mathbf{x}}$	$\alpha_{\mathbf{y}}$	$\text{TSA}_{\mathbf{x}}$	$\text{TSA}_{\mathbf{y}}$	TSA	α	$\text{TSA}_{\mathbf{x}}$	$\text{TSA}_{\mathbf{y}}$	TSA
Linear(%)	$58.6 \pm 0.2\uparrow$	50.0 ± 0.0	$82.4 \pm 0.3\uparrow$	82.8 ± 0.2	82.2 ± 0.1	56.1 ± 0.3	81.8 ± 0.3	83.7 ± 0.4	82.5 ± 0.3
Gaussian(%)	$61.1 \pm 0.2\uparrow$	50.0 ± 0.0	$83.3 \pm 0.5\uparrow$	85.7 ± 0.4	84.8 ± 0.3	58.4 ± 0.4	81.1 ± 0.4	86.6 ± 0.3	85.2 ± 0.4

From Table II and Table III, we can see that, the accuracies of BMPM for the biased class are increased significantly when compared with those of MPM in both linear and Gaussian kernel settings, which indicate that the corresponding decision boundaries are biased towards the biased class. Meanwhile, we observe that the accuracies of BMPM for the less important class still maintain at an acceptable level by setting the lower bound. We also note that the worst-case bounds are all smaller than the real test-set accuracies. This shows again that the worst-case probability can serve as the quantitative indicator of the medical diagnosis for the future cases. Comparing the results of linear kernel with the results of Gaussian kernel, we also find that both the worst-case bound and test accuracy for the biased class in the Gaussian kernel are greater than those of the linear kernel. This also demonstrates the advantage of Gaussian kernel setting.

5. Conclusion

The Minimax Probability Machine, a recently-proposed novel classifier, provides a worst-case bound on the probability of misclassification of future data points and achieves the comparative performance with a state-of-the-art classifier, the Support Vector Machine. In this paper, by eliminating the assumption of the

unbiased weight for each class in the Minimax Probability Machine, we develop a critical tool, Biased Minimax Probability Machine, which is the first quantitative method to control how the decision hyperplane moves in favor of the classification of the more important class, to deal with biased classification tasks, especially in the medical diagnostic applications. This model is transformed into a concave-convex Fractional Programming problem or a pseudoconcave problem. After illustrating our model with a synthetic dataset and applying it to the real-world medical diagnosis dataset, we obtain encouraging and promising experimental results.

Some important issues need to be checked as our future work. Firstly, our model relies a lot on the good estimates of the means and covariance matrices, how can we estimate them accurately and robustly is one important issue. Secondly, are there other more efficient methods to solve the Fractional Programming optimization problem? Can some decomposition techniques be applied in the Gram matrix and thus speed up the least-squares training? Finally, what we mentioned is in the scope of two-category classification tasks. The scheme to extend to the multi-category tasks is also one of our research topics in the near future.

Acknowledgements

We thank Gert R. G. Lanckriet at U. C. Berkeley for providing the Matlab source code of Minimax Probability Machine on the web. We also want to thank Michael I. Jordan at U. C. Berkeley for his encouragement and comments.

References

[1] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massashusetts, 2nd edition, 1999.

[2] C. L. Blake and C. J. Merz. Repository of machine learning databases, University of California, Irvine, http://www.ics.uci.edu/∼mlearn/mlrepository.html, 1998.

[3] K. Huang, H. Yang, I. King, M. R. Lyu, and L. W. Chan. Minimum error minimax probability machine. *Submitted to Journal of Machine Learning Research*, 2003.

[4] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. Minimax probability machine. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

[5] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2002.

[6] M. Lobo, L. Vandenberghe, , S. Boyd, and H. Lebret. An analysis of bayesian classifiers. *Applications of second order cone programming. Linear Algebra and its Applications*, 284:193–228, 1998.

[7] Y. Nesterov and A. Nemirovsky. Interior point polynomial methods in convex programming: Theory and applications. *SIAM, Philadelphia, PA.*, 1994.

[8] E. Osuna, R. Freund, and F. Girosi. Support Vector Machines: Training and Applications. Technical Report AIM-1602, MIT, 1997.

[9] I. Popescu and D. Bertsimas. Optimal inequalities in probability theory: A convex optimization approach. Technical Report TM62, INSEAD, 2001.

[10] S. Schaible. Fractional programming. In R. Horst and P.M. Pardalos, editors, *Handbook of Global Optimization*, Nonconvex Optimization and Its Applications, pages 495–608. Kluwer Academic Publishers, Dordrecht-Boston-London, 1995.

[11] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[12] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 2nd edition, 1999.