Journal of the Operations Research
Society of Japan
2007, Vol. 50, No. 2, 123-136

AN EXTENSION OF A MINIMAX APPROACH TO MULTIPLE CLASSIFICATION

Tomonari Kitahara Shinji Mizuno Kazuhide Nakata
Tokyo Institute of Technology

(Received June 14, 2006; Revised November 24, 2006)

Abstract When mean vectors and covariance matrices of two classes are available in a binary classification problem, Lanckriet et al. [6] propose a minimax approach for finding a linear classifier which minimizes the worst-case (maximum) misclassification probability. In this paper, we extend the minimax approach to a multiple classification problem, where the number m of classes could be more than two.

Assume that mean vectors and covariance matrices of all the classes are available, but no further assumptions are made with respect to class-conditional distributions. Then we define a problem for finding linear classifiers which minimize the worst-case misclassification probability $\bar{\alpha}$. Unfortunately, no efficient algorithms for solving the problem are known. So we introduce the maximum pairwise misclassification probability $\bar{\beta}$ instead of $\bar{\alpha}$. It is shown that $\bar{\beta}$ is a lower bound of $\bar{\alpha}$ and a good approximation of $\bar{\alpha}$ when m or $\bar{\alpha}$ are small. We define a problem for finding linear classifiers which minimize the probability $\bar{\beta}$ and show some basic properties of the problem. Then the problem is transformed to a parametric Second Order Cone Programming problem (SOCP). We propose an algorithm for solving the problem by using nice properties of it. We conduct preliminary numerical experiments and confirm that classifiers computed by our method work very well to benchmark problems.

Keywords: Optimization, multiple classification, minimax approach, second order cone programming

1. Introduction

An object of a classification problem is to construct a model which predicts a class of a given sample. Due to its practical importance, many approaches to classification have been studied. These approaches include, for example, classification using the Mahalanobis distance, the Neural Network, and the Support Vector Machine (SVM).

When mean vectors and covariance matrices of two classes are available in a binary classification problem, Lanckriet et al. [6] propose a minimax approach for finding a linear classifier. Their algorithm is called the Minimax Probability Machine (MPM). The correct classification probability depends on (unknown) probability distributions of the classes. This fact means that a classifier may behave well under a particular combination of probability distributions, but quite poorly in other situation. The minimax approach provides a new scope against the difficulty. It finds a classifier which minimizes the worst-case (maximum) misclassification probability under all the possible choice of class-conditional densities with given mean vectors and covariance matrices. Lanckriet et al. [6] show that this problem is formulated as a Second Order Cone Programming problem (SOCP), which is easily solved by an interior point method. They demonstrate that a classifier computed by the MPM works effectively to practical problems.

After the presentation by Lanckriet et al. [6], several studies related to the MPM are made. Huang et al. [4] propose the Biased-MPM, which takes into account an importance of

each class. In [5], the same authors with [4] further generalize the Biased-MPM and develop the Minimum Error MPM, which brings tighter worst-case accuracy.

In this paper, we extend the minimax approach by Lanckriet et al. [6] to a multiple classification problem, where the number m of classes is not restricted to two. Unlike other binary classification methods, an extension of the minimax approach to the multiple classification is not studied well.

When we directly use a binary classification method to multiple classification, there are two major ways: one-against-all and one-against-one [3], and we point out that one-against-all MPM is studied by Hoi and Lyu [2]. In both the methods, we compute multiple binary classifiers, and classify a sample from the results of binary classification. Though both the methods have various rules to decide the class, the logic behind them may not be so obvious. Against these methods, in this paper we propose another approach, which treats all classes at once. In our approach, we aim to find classifiers whose worst-case misclassification probability is small.

Assume that mean vectors and covariance matrices of all the m classes are available, but no further assumptions are made with respect to class-conditional distributions. We define a problem for finding m linear classifiers which minimize the worst-case misclassification probability $\bar{\alpha}$ under all the possible choice of class-conditional densities with given mean vectors and covariance matrices. Unfortunately, no efficient algorithms for solving the problem are known. So we introduce the maximum pairwise misclassification probability $\bar{\beta}$ instead of $\bar{\alpha}$. It is shown that the newly defined probability $\bar{\beta}$ is a lower bound of $\bar{\alpha}$ and a good approximation of $\bar{\alpha}$ when m or $\bar{\alpha}$ are small. We define a problem for finding classifiers which minimize the maximum pairwise misclassification probability $\bar{\beta}$. The classifiers obtained from this problem give a classification region which consists of m convex polytopes and we show that they correctly classify the m mean vectors, see Theorem 3.1.

The underlying problem is transformed to a parametric SOCP, which has some nice properties. We propose an algorithm for solving it by using the properties. We also conduct preliminary numerical experiments and show that our classifiers work very effectively to benchmark problems and they are competitive to the ones computed by the SVM. So we preserve the effectiveness of the binary MPM to multiple case.

The rest of this paper is organized as follows. In Section 2 we briefly review the minimax approach to the binary classification problem by Lanckriet et al. [6]. In Section 3, we extend the minimax approach to the multiple classification problem and we define two problems for finding classifiers which minimize the worst-case misclassification probability $\bar{\alpha}$ and the pairwise probability $\bar{\beta}$. We show that the second problem is transformed to a parametric SOCP in Section 4. In Section 5, an algorithm for solving the problem is proposed. Some results of preliminary numerical experiments are shown in Section 6. We conclude the paper in Section 7 .

2. A Minimax Approach to Binary Classification

In this section, we review the minimax approach by Lanckriet et al. [6].

Let x be a random n-dimensional vector, which belongs to one of two classes, which are called Class 1 and Class 2. Suppose that we know the mean vector $\mu_i \in \Re^n$ and the covariance matrix $\Sigma_i \in \mathrm{S}^{\mathrm{n}}_{++}$ of each class $i \in \{1, 2\}$, where $\Re^n$ and $\mathrm{S}^{\mathrm{n}}_{++}$ denote the n-dimensional Euclidean space and the set of positive definite $n \times n$ symmetric matrices respectively. In this paper, we assume that the covariance matrices are positive definite for simplicity. This assumption is valid in actual when we add a regularization term to the

covariance matrices. No further assumptions are made with respect to the class-conditional distributions.

We determine a linear classifier $f(z) = a^T z + b$, where $a \in \Re^n \setminus \{0\}$, $b \in \Re$, and $z \in \Re^n$. For any given sample x, if $a^T x + b < 0$ then it is classified as Class 1, if $a^T x + b > 0$ then it is as Class 2, and if $a^T x + b = 0$, then it is classified as either Class 1 or Class 2.

For the classifier $f(z) = a^T z + b$, its worst-case misclassifying probability of Class 1 sample as Class 2 is expressed as

$$\sup_{x \sim (\mu_1, \Sigma_1)} \Pr\{a^T x + b \geq 0\},$$

where the supremum is taken over all probability distributions having the mean vector μ_1 and the covariance matrix Σ_1. Considering the other case similarly, the worst-case misclassification probability $\bar{\alpha}$ of the classifier $a^T z + b$ is given by

$$\bar{\alpha} = \max\{\sup_{x \sim (\mu_1, \Sigma_1)} \Pr\{a^T x + b \geq 0\}, \sup_{x \sim (\mu_2, \Sigma_2)} \Pr\{a^T x + b \leq 0\}\}.$$

Exploiting the complementary event property of probability, the worst-case correct classification probability is

$$\alpha = 1 - \bar{\alpha} = \min\{\inf_{x \sim (\mu_1, \Sigma_1)} \Pr\{a^T x + b < 0\}, \inf_{x \sim (\mu 2, \Sigma_2)} \Pr\{a^T x + b > 0\}\}.$$

The minimax approach seeks the classifier $f(z) = a^T z + b$, which minimizes the worst-case (maximum) misclassification probability $\bar{\alpha}$. The problem is written as

$$\min \bar{\alpha},$$

which is expressed as

$$\begin{array}{ll} \min & \bar{\alpha} \\ \text{subject to} & \sup_{x \sim (\mu_1, \Sigma_1)} \Pr\{a^T x + b \geq 0\} \leq \bar{\alpha}, \\ & \sup_{x \sim (\mu_2, \Sigma_2)} \Pr\{a^T x + b \leq 0\} \leq \bar{\alpha}, \end{array}$$

where $\bar{\alpha} \in [0, 1]$, $a \in \Re^n$, and $b \in \Re$ are variables. This problem is equivalent to

$$\begin{array}{ll} \max & \alpha \\ \text{subject to} & \inf_{x \sim (\mu_1, \Sigma_1)} \Pr\{a^T x + b < 0\} \geq \alpha, \\ & \inf_{x \sim (\mu_2, \Sigma_2)} \Pr\{a^T x + b > 0\} \geq \alpha. \end{array} \tag{2.1}$$

This expression looks for a classifier which maximizes the worst-case (minimum) correct classification probability. Lanckriet et al. [6] show that if $\mu_1 \neq \mu_2$ then $\alpha > 0$ and $a \neq 0$ at any optimal solution (α, a, b) of (2.1). We introduce an important result in [6] to handle the constraints in (2.1).

Lemma 2.1 *(Lanckriet et al. [6]) Let $(\mu_1, \Sigma_1) \in \Re^n \times \mathrm{S}^n_{++}$ be given. Then for any $(a, b) \in \Re^n \times \Re$ with $a \neq 0$, it holds that*

$$\inf_{x \sim (\mu_1, \Sigma_1)} \Pr\{a^T x + b < 0\} = \frac{s^2}{a^T \Sigma_1 a + s^2},$$

where $s \equiv \max\{-a^T \mu_1 - b, 0\}$.

We are interested in classifiers satisfying $a^T\mu_1 + b \leq 0$, since otherwise α in (2.1) is zero. Using this inequality and Lemma 2.1, the first constraint in (2.1) becomes

$$\begin{aligned} \inf_{x\sim(\mu_1,\Sigma_1)} \Pr\{a^T x + b < 0\} \geq \alpha \quad &\Leftrightarrow \quad \frac{s^2}{a^T\Sigma_1 a + s^2} \geq \alpha \\ &\Leftrightarrow \quad s \geq \eta(\alpha)\sqrt{a^T\Sigma_1 a} \\ &\Leftrightarrow \quad -a^T\mu_1 - b \geq \eta(\alpha)\|\Sigma_1^{1/2} a\|, \end{aligned}$$

where $\eta(\alpha) \equiv \sqrt{\frac{\alpha}{1-\alpha}}$. Similarly the second constraints in (2.1) is equivalent to

$$a^T\mu_2 + b \geq \eta(\alpha)\|\Sigma_2^{1/2} a\|.$$

Substitute these relations to (2.1), we have

$$\begin{array}{ll} \max & \alpha \\ \text{subject to} & -a^T\mu_1 - b \geq \eta(\alpha)\|\Sigma_1^{1/2} a\|, \\ & a^T\mu_2 + b \geq \eta(\alpha)\|\Sigma_2^{1/2} a\|. \end{array}$$

Since $\eta(\alpha)$ is an increasing function with respect to α, the problem is equivalent to

$$\begin{array}{ll} \max & \eta \\ \text{subject to} & -a^T\mu_1 - b \geq \eta\|\Sigma_1^{1/2} a\|, \\ & a^T\mu_2 + b \geq \eta\|\Sigma_2^{1/2} a\|. \end{array}$$

We can eliminate η in this problem (see [6] in detail) and get the next problem

$$\begin{array}{ll} \min & \|\Sigma_1^{1/2} a\| + \|\Sigma_2^{1/2} a\| \\ \text{subject to} & (\mu_2 - \mu_1)^T a = 1. \end{array} \tag{2.2}$$

This is a second order cone programming problem (SOCP). The problem (2.2) can be solved efficiently by an interior point method. In [6], Lancriet et al. call their algorithm the Minimax Probability Machine (MPM) and they demonstrate that the MPM is competitive to the Support Vector Machine (SVM).

3. A Minimax Approach to Multiple Classification

In practical applications, the number of classes of a classification problem is not always restricted to two. So we extend the minimax approach presented in Section 2 to multiple classification. In Section 3.1, we introduce linear classifiers whose values determine a class of each sample. In Section 3.2, we define a problem for determining the classifiers so that the worst-case misclassification probability is minimized. Since the problem defined in Section 3.2 is not easy to solve, we propose a problem which minimizes an approximation of the worst-case misclassification probability in Section 3.3.

3.1. Multiple classification of a random vector

We define an index set $M = \{1, 2, \ldots, m\}$ for an integer $m \geq 2$. Suppose that we have m classes of n-dimensional random vectors. The i-th class is called Class i for each $i \in M$. Assume that the mean vector $\mu_i \in \Re^n$ and the covariance matrix $\Sigma_i \in \mathrm{S}^n_{++}$ of each Class $i \in M$ are known. No further assumptions are made with respect to the class-conditional distributions.

In the binary classification stated in Section 2, we use only one classifier $f(z) = a^T z + b$ and classify a sample x according to the sign of $f(x)$. When we have two functions $f_1(z)$ and $f_2(z)$ such that $f(z) = f_1(z) - f_2(z)$, the condition $f(z) > 0$ is equivalent to $f_1(z) > f_2(z)$. So we can equivalently classify the sample x as Class 1 if $f_1(x) < f_2(x)$ and as Class 2 if $f_1(x) > f_2(x)$. In multiple classification, we prepare a linear classifier

$$f_i(z) = a_i^T z + b_i \text{ for each } i \in M, \tag{3.1}$$

where $a_i \in \Re^n$ and $b_i \in \Re$. Then a sample x is classified as a class which gives the minimum value of $f_i(x)$. To express formally, the sample x is classified as Class l when

$$f_l(x) = \min_{i \in M} f_i(x).$$

If two or more functions have the same minimum value, we can choose any one of them. In this section, we adopt this classification rule. The problem is how to choose m linear classifiers $f_i(z) = a_i^T z + b_i$, $i \in M$.

3.2. A minimization problem of the worst-case misclassification probability

For any $i \in M$, we define the set

$$R_i \equiv \{z \in \Re^n | a_i^T z + b_i < a_j^T z + b_j \text{ for any } j \in M, j \neq i\}.$$

It is easy to see that any sample x belongs to one of such sets R_i (or the closure of R_i), $i \in M$. In our rule, the sample x is judged as Class i when $x \in R_i$. As we do in Section 2, we want to find classifiers $f_i(z) = a_i^T z + b_i$ $(i \in M)$ which minimize the worst-case misclassification probability. Since the mean vector $\mu_i \in \Re^n$ and the covariance matrix $\Sigma_i \in \mathrm{S}_{++}^n$ are known, the worst-case misclassification probability of a sample x in Class i, $i \in M$ is expressed as

$$\bar{\alpha}_i = \sup_{x \sim (\mu_i, \Sigma_i)} \Pr\{x \notin R_i\}. \tag{3.2}$$

Then the correct classification probability is

$$\alpha_i = 1 - \bar{\alpha}_i = \inf_{x \sim (\mu_i, \Sigma_i)} \Pr\{x \in R_i\}.$$

Our problem is to compute classifiers $f_i(z) = a_i^T z + b_i$ $(i \in M)$ which minimize

$$\bar{\alpha} = \max_{i \in M} \bar{\alpha}_i \tag{3.3}$$

or equivalently maximize

$$\alpha = 1 - \bar{\alpha} = \min_{i \in M} \alpha_i.$$

This problem is expressed as

$$\begin{array}{ll} \max & \alpha \\ \text{subject to} & \inf_{x \sim (\mu_i, \Sigma_i)} \Pr\{x \in R_i\} \geq \alpha, \quad i \in M, \end{array} \tag{3.4}$$

where $\alpha \in [0, 1]$, $a_i \in \Re^n$ $(i \in M)$, and $b_i \in \Re$ $(i \in M)$ are variables. Unfortunately, the problem (3.4) is not easy to solve, and we do not know any efficient method for solving it.

3.3. An approximation of the worst-case misclassification probability

Instead of the misclassification probability $\bar{\alpha}_i$, we introduce the maximum worst-case pairwise misclassification probability which is defined as

$$\bar{\beta}_i = \max_{j \neq i} \sup_{x \sim (\mu_i, \Sigma_i)} \Pr\{a_i^T x + b_i \geq a_j^T x + b_j\}. \tag{3.5}$$

Then the minimum pairwise correct classification probability is

$$\beta_i = 1 - \bar{\beta}_i = \min_{j \neq i} \inf_{x \sim (\mu_i, \Sigma_i)} \Pr\{a_i^T x + b_i < a_j^T x + b_j\}.$$

In the next lemma, we show that $\bar{\beta}_i$ is a lower bound of $\bar{\alpha}_i$.

Lemma 3.1 *When the misclassification probability $\bar{\alpha}_i$ and the maximum pairwise misclassification probability $\bar{\beta}_i$ for each $i \in M$ are defined by (3.2) and (3.5) respectively, we have*

$$\bar{\beta}_i \leq \bar{\alpha}_i \leq (m-1)\bar{\beta}_i. \tag{3.6}$$

Proof: Since the set R_i is a subset of $\{z \in \Re^n | a_i^T z + b_i < a_j^T z + b_j\}$ for any $j \neq i$, we have

$$\Pr\{x \in R_i\} \leq \min_{j \neq i} \Pr\{a_i^T x + b_i < a_j^T x + b_j\}$$

for any random vector x. The first inequality in (3.6) follows from this inequality and the complementary event property. We can get the second inequality in (3.6) from

$$\begin{aligned}
\Pr\{x \in R_i\} &= \Pr\{a_i^T x + b_i < a_j^T x + b_j \text{ for any } j \neq i\} \\
&= 1 - \Pr\{a_i^T x + b_i \geq a_j^T x + b_j \text{ for some } j \neq i\} \\
&\geq 1 - \sum_{j \neq i} \Pr\{a_i^T x + b_i \geq a_j^T x + b_j\} \\
&\geq 1 - (m-1) \max_{j \neq i} \Pr\{a_i^T x + b_i \geq a_j^T x + b_j\}.
\end{aligned}$$

□

From the above lemma, the maximum pairwise misclassification probability $\bar{\beta}_i$ is a good approximation of the misclassification probability $\bar{\alpha}_i$ for each $i \in M$, when m or $\bar{\alpha}_i$ are small. Especially when $m = 2$, $\bar{\beta}_i$ is equal to $\bar{\alpha}_i$. So we consider the problem for finding classifiers $f_i(z) = a_i^T z + b_i$ $(i \in M)$ which minimize

$$\bar{\beta} = \max_{i \in M} \bar{\beta}_i \tag{3.7}$$

or equivalently maximize

$$\beta = 1 - \bar{\beta} = \min_{i \in M} \beta_i.$$

Such a β is the solution of the problem

$$\begin{array}{ll} \max & \beta \\ \text{subject to} & \inf_{x \sim (\mu_i, \Sigma_i)} \Pr\{a_i^T x + b_i < a_j^T x + b_j\} \geq \beta, \ i, j \in M, \ j \neq i. \end{array} \tag{3.8}$$

We get the next results for this problem.

Theorem 3.1 *If mean vectors of two classes are same ($\mu_i = \mu_j$ for some $i, j \in M$, $i \neq j$), the optimal value β of the problem (3.8) is 0. Otherwise $\beta > 0$ and $\mu_i \in R_i$ for each $i \in M$ at any optimal solution of the problem (3.8).*

See Appendix for the proof of this theorem.

Since the problem (3.8) does not have a meaningful solution if mean vectors of two classes are same, we assume that all the mean vectors are distinct in the remainder of this paper.

4. A Parametric SOCP

In this section, we show that the problem (3.8) is transformed to a parametric second order cone programming problem.

As Theorem 3.1 suggests, we can add constraints $\mu_i \in R_i$ $(i \in M)$, or equivalently

$$a_i^T \mu_i + b_i < a_j^T \mu_i + b_j \tag{4.1}$$

for any $i, j \in M$, $i \neq j$ to the problem (3.8). Note that two constraints

$$a_i^T \mu_i + b_i < a_j^T \mu_i + b_j$$

and

$$a_j^T \mu_j + b_j < a_i^T \mu_j + b_i$$

lead to

$$(a_j - a_i)^T (\mu_i - \mu_j) > 0,$$

which means $a_i \neq a_j$ for any i and j with $i \neq j$. Then, from Lemma 2.1, we have that

$$\begin{aligned} \inf_{x \sim (\mu_i, \Sigma_i)} \Pr\{a_i^T x + b_i < a_j^T x + b_j\} \geq \beta \quad &\Leftrightarrow \quad \frac{s'^2}{(a_i - a_j)^T \Sigma_i (a_i - a_j) + s'^2} \geq \beta \\ &\Leftrightarrow \quad s' \geq \eta(\beta) \|\Sigma_i^{1/2} (a_i - a_j)\|, \end{aligned}$$

where $s' = \max\{-(a_i - a_j)^T \mu_i - (b_i - b_j), 0\}$ and $\eta(\beta) = \sqrt{\frac{\beta}{1-\beta}}$. Notice that

$$-(a_i - a_j)^T \mu_i - (b_i - b_j) > 0$$

holds for any $i, j \in M$, $i \neq j$ from (4.1). Then $s' = -(a_i - a_j)^T \mu_i - (b_i - b_j)$ and the constraint in (3.8) becomes

$$-(a_i - a_j)^T \mu_i - (b_i - b_j) \geq \eta(\beta) \|\Sigma_i^{1/2} (a_i - a_j)\|.$$

Hence the problem (3.8) is equivalent to

$$\begin{array}{ll} \max & \beta \\ \text{subject to} & -(a_i - a_j)^T \mu_i - (b_i - b_j) \geq \eta(\beta) \|\Sigma_i^{1/2} (a_i - a_j)\|, \ i, j \in M, \ j \neq i, \\ & -(a_i - a_j)^T \mu_i - (b_i - b_j) > 0, \ i, j \in M, \ j \neq i. \end{array} \tag{4.2}$$

Since the constraints of (4.2) are positive homogeneous in a_i, b_i, $i \in M$, that is, if $(\beta, a_1, b_1, \ldots, a_m, b_m)$ is a feasible solution, then $(\beta, sa_1, sb_1, \ldots, sa_m, sb_m)$ is also feasible for any $s > 0$. This means that the constraints

$$-(a_i - a_j)^T \mu_i - (b_i - b_j) > 0, \ i, j \in M, \ j \neq i$$

in (4.2) can be replaced with

$$-(a_i - a_j)^T \mu_i - (b_i - b_j) \geq u, \ i, j \in M, \ j \neq i,$$

where $u > 0$ is a constant. Consequently, we get the following parametric second order cone programming problem

$$\begin{array}{ll} \max & \beta \\ \text{subject to} & -(a_i - a_j)^T \mu_i - (b_i - b_j) \geq \eta(\beta) \|\Sigma_i^{1/2} (a_i - a_j)\|, \ i, j \in M, \ j \neq i, \\ & -(a_i - a_j)^T \mu_i - (b_i - b_j) \geq u, \ i, j \in M, \ j \neq i. \end{array} \tag{4.3}$$

Problem (4.3) seems to be difficult to solve due to nonlinear constraints in it, but it can be solved easily by using special properties of the problem.

5. An Algorithm for the Parametric SOCP

In this section, we explain two important properties of the parametric SOCP (4.3) and we propose an algorithm which exploits the properties.

First and most obviously, if we fix β in (4.3), then the constraints in (4.3) become second order cone (SOC) constraints. General SOC constraint on a variable $x \in \Re^n$ has the form

$$c^T x + d \geq \|Ax + b\|, \tag{5.1}$$

where $A \in \Re^{m \times n}$, $b \in \Re^m$, $c \in \Re^n$ and $d \in \Re$ are given data. We can easily check whether (5.1) has a feasible solution or not by solving the next problem

$$\begin{array}{ll} \min_{x,t} & t \\ \text{subject to} & c^T x + d + t \geq \|Ax + b\|, \\ & t \geq 0, \end{array} \tag{5.2}$$

where $t \in \Re$ is a new variable. This is an SOCP which can be solved efficiently by an interior point method. The inequality (5.1) is feasible if and only if the optimal objective value of (5.2) is zero. Then we obtain a feasible solution by solving (5.2).

Second property is that (4.3) has a monotonicity, as shown in the following lemma.

Lemma 5.1 *If the problem (4.3) is infeasible for $\beta = \beta' \in (0, 1)$, then it is infeasible for any $\beta \in [\beta', 1)$.*

Proof: Suppose in contrast that the problem (4.3) is infeasible for $\beta' \in (0, 1)$, but it is feasible for some $\beta'' \in [\beta', 1)$. Let (a_i, b_i) $(i \in M)$ be a feasible solution for β''. Since $\eta(\beta) = \sqrt{\frac{\beta}{1-\beta}}$ increases monotonically with respect to β, we have $\eta(\beta'') \geq \eta(\beta')$. Then for all i, j satisfying $i \neq j$, it holds that

$$-(a_i - a_j)^T \mu_i - (b_i - b_j) \geq \eta(\beta'') \|\Sigma_i^{1/2}(a_i - a_j)\| \geq \eta(\beta') \|\Sigma_i^{1/2}(a_i - a_j)\|.$$

This means that (a_i, b_i) $(i \in M)$ constitute a feasible solution of the problem (4.3) for β', which is contradiction. □

Thanks to the properties mentioned in Lemma 5.1, we can use the so-called bisection method to solve (4.3). In the following, we first explain basic ideas of the algorithm, then we give the formal representation.

Let β_* be the optimal value of (4.3). Initially we only know that β_* lies in the interval $[0, 1)$. At first, we take the middle value of the interval, say $\beta = 0.5$, and ask whether the problem (4.3) is feasible or not for this value. In this paper, we represent this procedure as **bisection**$([0, 1))$. More precisely, for $[\beta_l, \beta_u) \subset [0, 1)$,

$$\textbf{bisection}([\beta_l, \beta_u)) = \begin{cases} 1, & \text{if the problem (4.3) is feasible for } \beta = \frac{\beta_l + \beta_u}{2}, \\ 0, & \text{otherwise.} \end{cases}$$

The procedure **bisection**$([0, 1))$ can be done by solving an SOCP as explained above. If **bisection**$([0, 1)) = 1$, then we conclude that β_* lies in the interval $[0.5, 1)$, otherwise we can say β_* is in the interval $[0, 0.5)$ from Lemma 5.1. Note that we can make the initial interval $[0, 1)$ half in any case. Next we execute **bisection**$([0, 0.5))$ or **bisection**$([0.5, 1))$, according to the situation, and repeat the procedure. Obviously, we can make the interval as small as we want. The formal representation of the algorithm is given in Table 1.

Table 1: The formal description of the algorithm

Input:
mean vectors and covariance matrices (μ_i, Σ_i) $(i = 1, \ldots, m)$;
accuracy parameter $\epsilon > 0$;
constant $u > 0$;
initial interval $[\beta_l, \beta_u) = [0, 1)$;

while $(\beta_u - \beta_l \geq \epsilon)\{$
  if (**bisection**$([\beta_l, \beta_u)) = 1)\{$
    $[\beta_l, \beta_u) \leftarrow [\frac{\beta_l+\beta_u}{2}, \beta_u)$;
    update (a_i, b_i) $(i = 1, \ldots, m)$;
  $\}$
  else $\{$
    $[\beta_l, \beta_u) \leftarrow [\beta_l, \frac{\beta_l+\beta_u}{2})$;
  $\}$
$\}$

Output:
approximate optimal value β_l ;
solution (a_i, b_i) $(i = 1, \ldots, m)$;

6. Preliminary Numerical Experiments

In this section, we conduct preliminary numerical experiments to see the actual accuracy of the proposed classification method, although it takes into account the worst-case performance.

Though in this paper we address a classification problem where the mean vector and the covariance matrix of each class are known, we use benchmark problems where all individual data are observed. For each problem, we first estimate the mean vectors and the covariance matrices of all the classes from the data. For regularization, if there is a class whose covariance matrix is not positive definite, we add ρI, where $\rho = 10^{-8}$ and I is the n-dimensional unit matrix, to it. With these estimates, we compute the classifiers for each problem, then classify data using these classifiers.

We collect four problems **iris**, **wine**, **glass**, **vehicle** * from the UCI Repository of machine learning databases [8]. Problem data are summarized in Table 2. We set the accuracy

Table 2: Problem data

Problem	# of data	# of class (m)	# of attributes (n)
iris	150	3	4
wine	178	3	13
glass	214	6	9
vehicle	846	4	18

parameter ϵ in the algorithm to 0.001. We observe that the parameter $u > 0$ in (4.3) somewhat decides the stability of our algorithm. In this paper, we set its value to 0.1. We

*Though **vehicle** is available from the UCI Repository, it is originally from the Statlog collection.

execute our algorithm on MATLAB, and use SeDuMi [9][10] for the SOCP solver. We note here that we use YALMIP [7] for modeling of our problem.

We present results in Table 3, where the second column represents the worst-case pairwise probability, and the third column shows the actual accuracy using the classifiers obtained. We can see from Table 3 that the actual accuracy is considerably greater than the worst-case

Table 3: Results

Problem	β	accuracy	accuracy(SVM, linear kernel)
iris	0. 780	0. 980	0. 973
wine	0. 860	1. 000	0. 994
glass	0. 312	0. 640	0. 664
vehicle	0. 254	0. 780	0. 809

pairwise probability for each problem. Note that as (3.6) suggests, the worst-case accuracy of the classifiers is equal to or less than the worst-case pairwise probability. These two facts indicate that there is a large gap between the worst-case accuracy of the classifiers and the actual accuracy, so that classifiers work very well for practical problems.

To compare the proposed method with other methods, we list the accuracy for each problem by an SVM with the linear kernel in the fourth column of Table 3. These are taken from [3]. We observe that our method is competitive to the SVM, which is one of the most popular and effective classification method. This encourages the proposed method as a promising classification method.

7. Conclusions

In this paper, we have studied an extension of a recently developed minimax approach by Lanckriet et al. [6] to multiple classification. When mean vectors and covariance matrices of two classes are available, the minimax approach finds a linear classifier which minimizes the maximum misclassification probability. Such a classifier can be obtained by solving a relevant Second Order Cone Programming problem (SOCP).

Though it is possible to directly use the minimax approach to multiple classification, we propose another approach, which aims to find classifiers with small worst-case misclassification probability by handling all classes at once.

For a multiple classification problem, we define the problem (3.4) for finding the linear classifiers (3.1) which minimize the worst-case misclassification probability $\bar{\alpha}$ defined by (3.3). Unfortunately, no efficient algorithms for solving the problem are known. So we introduce the maximum pairwise misclassification probability $\bar{\beta}$ by (3.7) instead of $\bar{\alpha}$. It is shown in Lemma 3.1 that the probability $\bar{\beta}$ is a lower bound of $\bar{\alpha}$ and it is a good approximation of $\bar{\alpha}$ when m or $\bar{\alpha}$ are small. We define the problem (3.8) for finding the classifiers which minimize the maximum pairwise misclassification probability $\bar{\beta}$. The classifier obtained from this problem give a classification region consisting of m convex polytopes, and we show that they correctly classify the m mean vectors, see Theorem 3.1.

We show that the problem (3.8) is transformed to the parametric SOCP (4.3) in Section 4. In Section 5, we show that the parametric SOCP has important properties, and we propose the algorithm in Table 1 for solving it by using the properties. We conduct preliminary numerical experiments and confirm that the classifiers of our method work very well to benchmark problems in Section 6. As shown in Table 3, the results are competitive to

the Support Vector Machine (SVM), which supports the proposed method as a promising multiple classification method.

There are some future works to be done. For example, to consider robust version of our method and/or kernelization of our method are interesting topics.

Acknowledgements
The authors are grateful to two anonymous referees. They provided very useful comments to our paper. This research is supported in part by Grant-in-Aid for Scientific Research (A) 16201033 and Young Scientists (B) 17710126 of Japan Society for the Promotion of Science.

A. Proof of Theorem 3.1

We use the following lemma to prove Theorem 3.1. Although it looks that the result of the lemma intuitively holds, we give a mathematical proof for preciseness.

Lemma A.1 *Let $m \geq 2$ and $n \geq 1$ be integers and $M = \{1, 2, \ldots, m\}$. If all the points $\mu_i \in \Re^n$ $(i \in M)$ are distinct, there exist m linear functions $a_i^T x + b_i$ $(i \in M)$ such that*

$$a_i^T \mu_i + b_i < a_j^T \mu_i + b_j \text{ for any } i, j \in M, i \neq j. \tag{A.1}$$

Proof: This fact is proved by induction. In the case $m = 2$, the statement is obviously true. Assume that the statement is true in the case $m = k$. Let μ_i $(i = 1, \ldots, k+1)$ be $k+1$ distinct points. From the assumption, we have k linear functions $a_i^T x + b_i$ $(i = 1, \ldots, k)$ satisfying

$$a_i^T \mu_i + b_i < a_j^T \mu_i + b_j \text{ for any } i, j \in \{1, 2, \ldots, k\}, i \neq j.$$

We define

$$f_{k+1} \equiv \min\{a_i^T \mu_{k+1} + b_i,\ \ i = 1, \ldots, k\}.$$

Let l be an index which attains the minimum, that is,

$$f_{k+1} = a_l^T \mu_{k+1} + b_l. \tag{A.2}$$

We also define

$$f_i \equiv a_i^T \mu_i + b_i,\ \ i = 1, \ldots, k.$$

Consider k half lines defined as

$$L_i \equiv \{(x, x_{n+1}) \in \Re^n \times \Re |\ x = \mu_i, x_{n+1} \leq f_i\},\ \ i = 1, \ldots, k.$$

We claim that

$$(\mu_{k+1}, f_{k+1}) \notin \text{conv}(L_1 \cup \ldots \cup L_k), \tag{A.3}$$

where $\text{conv}(S)$ denotes the convex hull of any set S. Otherwise, there exist finite points

$$(\mu_1, x_{11}), \ldots, (\mu_1, x_{1p(1)}) \in L_1, \ldots, (\mu_k, x_{k1}), \ldots, (\mu_k, x_{kp(k)}) \in L_k$$

and coefficients

$$\lambda_{11}, \ldots, \lambda_{1p(1)}, \ldots, \lambda_{k1}, \ldots, \lambda_{kp(k)}$$

such that

$$\begin{aligned}\sum_{i=1}^{k}\sum_{j=1}^{p(i)}\lambda_{ij}(\mu_i, x_{ij}) &= (\mu_{k+1}, f_{k+1}),\\ \sum_{i=1}^{k}\sum_{j=1}^{p(i)}\lambda_{ij} &= 1,\\ \lambda_{ij} &\geq 0,\ i=1,\ldots,k,\ j=1,\ldots,p(i).\end{aligned}$$

We have that

$$\begin{aligned}a_l^T\mu_{k+1}+b_l-f_{k+1} &= a_l^T\left(\sum_{i=1}^{k}\sum_{j=1}^{p(i)}\lambda_{ij}\mu_i\right)+b_l-\sum_{i=1}^{k}\sum_{j=1}^{p(i)}\lambda_{ij}x_{ij}\\ &\geq \sum_{i=1}^{k}\sum_{j=1}^{p(i)}\lambda_{ij}a_l^T\mu_i+b_l-\sum_{i=1}^{k}\sum_{j=1}^{p(i)}\lambda_{ij}f_i\\ &= \sum_{i=1}^{k}\sum_{j=1}^{p(i)}\lambda_{ij}a_l^T\mu_i+b_l-\sum_{i=1}^{k}\sum_{j=1}^{p(i)}\lambda_{ij}(a_i^T\mu_i+b_i)\\ &\geq \sum_{i=1}^{k}\sum_{j=1}^{p(i)}\lambda_{ij}a_l^T\mu_i+b_l-\sum_{i=1}^{k}\sum_{j=1}^{p(i)}\lambda_{ij}(a_l^T\mu_i+b_l) \qquad \text{(A.4)}\\ &= 0.\end{aligned}$$

This relation and (A.2) imply that the inequality (A.4) holds as equality. So we have $\lambda_{ij}=0$ if $a_i^T\mu_i+b_i < a_l^T\mu_i+b_l$ or equivalently $i\neq l$. Hence $\sum_{j=1}^{p(l)}\lambda_{lj}=1$, which means $\mu_{k+1}=\mu_l$. This contradicts our assumption.

From (A.3) and the fact that $\mathrm{conv}(L_1\cup\ldots\cup L_k)$ is a closed convex set, the separating hyperplane theorem [1] guarantees that there exists a linear function $\beta^Tx+\gamma x_{n+1}+\xi$ $((\beta,\gamma,\xi)\in\Re^n\times\Re\times\Re)$ which satisfies

$$\begin{aligned}\beta^Tx+\gamma x_{n+1}+\xi &< 0 \text{ for any } (x,x_{n+1})\in\mathrm{conv}(L_1\cup\ldots\cup L_k),\\ \beta^T\mu_{k+1}+\gamma f_{k+1}+\xi &> 0.\end{aligned} \qquad \text{(A.5)}$$

Note that we must have $\gamma\geq 0$. Otherwise, if we take $(\mu_1, x_{n+1})\in L_1$ for sufficiently small x_{n+1}, the first strict inequality in (A.5) is violated. As $(\mu_i, f_i)\in L_i$, it holds that

$$\beta^T\mu_i+\gamma f_i+\xi<0\ (i=1,\ldots,k).$$

Putting these strict inequalities together, we have

$$\begin{aligned}\beta^T\mu_i+\gamma f_i+\xi &< 0\ (i=1,\ldots,k),\\ \beta^T\mu_{k+1}+\gamma f_{k+1}+\xi &> 0.\end{aligned}$$

As we can add ϵf_i (with $\epsilon>0$ sufficiently small) to each of them without violating strict inequalities, we assume that $\gamma>0$ in these $k+1$ strict inequalities. Then if we define

$$a_{k+1}\equiv-\frac{1}{\gamma}\beta,\ \ b_{k+1}=-\frac{1}{\gamma}\xi,$$

$k+1$ linear functions $a_i^T x + b_i$ $(i = 1, \ldots, k+1)$ satisfies the statement of the lemma in the case $m = k+1$. □

Proof of Theorem 3.1: Suppose that $\mu_i = \mu_j$ for some i and j. Then, for any $(a_i, b_i), (a_j, b_j) \in \Re^n \times \Re$, we have either $-(a_i - a_j)^T \mu_i - (b_i - b_j) \leq 0$ or $-(a_j - a_i)^T \mu_j - (b_j - b_i) \leq 0$. If $a_i \neq a_j$, these inequalities and Lemma 2.1 mean either $\inf_{x \sim (\mu_i, \Sigma_i)} \Pr\{a_i^T x + b_i < a_j^T x + b_j\} = 0$ or $\inf_{x \sim (\mu_j, \Sigma_j)} \Pr\{a_j^T x + b_j < a_i^T x + b_i\} = 0$. If $a_i = a_j$, obviously one of these equalities holds. Hence the optimal value β of the problem (3.8) is 0.

To show the latter part of the statement, we assume that all of the mean vectors are distinct. Then there are m linear functions $a_i^T x + b_i$ $(i = 1, \ldots, m)$ which satisfy (A.1). With these linear classifiers, it holds, with the help of Lemma 2.1, that

$$\inf_{x \sim (\mu_i, \Sigma_i)} \Pr\{a_i^T x + b_i < a_j^T x + b_j\} > 0, \ i, j \in M, i \neq j.$$

This proves that the optimal value β of the problem (3.8) is positive. Clearly the optimal solution satisfies (A.1). This means $\mu_i \in R_i$ for all $i = 1, \ldots, m$. □

References

[1] S. Boyd and L. Vandenberghe: *Convex Optimization* (Cambridge University Press, 2004).

[2] C.H. Hoi and M.R. Lyu: Robust face recognition using minimax probability machine. *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME 2004)*, (IEEE, 2004), 1175–1178.

[3] C.W. Hsu and C.J. Lin: A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, **13** (2002), 415–425.

[4] K. Huang, H. Yang, I. King, and M.R. Lyu: Learning classifiers from imbalanced data based on biased minimax probability machine. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, (IEEE, 2004), 558–563.

[5] K. Huang, H. Yang, I. King, M.R. Lyu, and L. Chan: The minimum error minimax probability machine. *Journal of Machine Learning Research*, **5** (2004), 1253–1286.

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

[7] J. Lofberg: YALMIP: A toolbox for modeling and optimization in MATLAB. *Proceedings of the 2004 IEEE International Symposium on Computer Aided Control Systems Design*, (IEEE, 2004), 284–289.

[8] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz: *UCI Repository of Machine Learning Databases*. Department of Information and Computer Sciences, University of California (1998). http://www.ics.uci.edu/~mlearn/MLRepository.html

[9] I. Polik: *Addendum to the SeDuMi User Guide Version 1.1*. Advanced Optimization Lab, McMaster University (2005).

[10] J.F. Sturm: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, **11-12** (1999), 625–653.

Tomonari Kitahara
Department of Industrial Engineering and Management
Tokyo Institute of Technology
2-12-1 Ohokayama
Meguro Tokyo 152-8552, Japan
E-mail: kitahara.t.ab@m.titech.ac.jp