Second Order Cone Programming Formulations for Robust Multi-class Classification[1]

Ping Zhong[2] and Masao Fukushima[3]

Abstract Multi-class classification is an important and on-going research subject in machine learning. Current support vector methods for multi-class classification implicitly assume that the parameters in the optimization problems to be known exactly. However, in practice, the parameters have perturbations since they are estimated from the training data which are usually subject to measurement noise. In this paper, we propose linear and nonlinear robust formulations for multi-class classification based on M-SVM method. The preliminary numerical experiments confirm the robustness of the proposed method.

Keywords: Multi-class classification; Support vector machine; Second-order cone program; Robust classifier

AMS Subject classification: 65K05, 68T10, 68Q32

1 Introduction

Given L labeled examples known to come from $K(>2)$ classes

$$\mathcal{T} = \{(\boldsymbol{x}_p, \theta_p)\}_{p=1}^{L} \subset \mathcal{X} \times \mathcal{Y},$$

where $\mathcal{X} \subset \mathcal{R}^N$ and $\mathcal{Y} = \{\Theta_1, \cdots, \Theta_K\}$, multi-class classification refers to the construction of a discriminate function from the input space $\mathcal{X}$ onto the unordered set of classes $\mathcal{Y}$.

Support vector machines (SVMs) serve as a useful and popular tool for classification. Recent developments in the study on SVMs show that there are roughly two types of

[1] This work was supported in part by the Scientific Research Grant-in-Aid from Japan Society for the Promotion of Science and the National Science Foundation of China Grant No. 10371131.

[2] Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan./Faculty of Science, China Agricultural University, Beijing, 100083, China. Email: zping@amp.i.kyoto-u.ac.jp

[3] Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan. Email: fuku@amp.i.kyoto-u.ac.jp

approaches to tackle multi-class classification problem. One is to construct and fuse several binary classifiers, such as 'one-against-all' [7, 20], 'one-against-one' [12, 14], directed acyclic graph SVM (DAGSVM) [17], 'error-correcting output code (ECOC)' [2, 9], K-SVCR method [3] and ν-K-SVCR method [23] and so on. The other, called 'all-together', is to consider all data in one optimization formulation [4, 5, 11, 20, 21, 22]. In this paper, we focus on the second approach.

There are several all-together methods. The method independently proposed in [20] and [21] is similar to 'one-against-all'. It constructs K two-class discriminants where each discriminant separates a single class from all the others. Hence there are K decision functions but all are obtained by solving one optimization problem. In [4] a piecewise-linear discriminant for the K-class classification is constructed by a single linear program. The method called M-SVM [5] extends the method in [4] to generate a kernel based nonlinear K-class discriminant by solving a convex quadratic program. Although the original forms proposed in [20, 21] and [5] are different, it is pointed out in [11] that they are not only equivalent to each other, but also equivalent to that in [11]. Based on M-SVM, the linear programming formulations are proposed in a low dimensional feature subspace [22].

In the above mentioned methods, the parameters in the optimization problems are implicitly assumed to be known exactly. However, in practice, these parameters have perturbations since they are estimated from the training data which are usually corrupted by measurement noise. As pointed out in [10], the solutions to the optimization problems are sensitive to parameter perturbations. Errors in the input space tend to get amplified in the decision function, which often results in misclassification. So it will be useful to explore formulations that can yield discriminants robust to such estimation errors. In this paper we propose a robust formulation of M-SVM, which is represented as a second-order cone program (SOCP). The second-order cone (SOC) in $\mathcal{R}^n$ ($n \geq 1$), also called the Lorentz cone, is the convex cone defined by

$$\mathcal{K}^n = \left\{ \begin{bmatrix} z_0 \\ \bar{\boldsymbol{z}} \end{bmatrix} : \, z_0 \in \mathcal{R}, \, \bar{\boldsymbol{z}} \in \mathcal{R}^{n-1}, \, \|\bar{\boldsymbol{z}}\| \leq z_0 \right\},$$

where $\|\cdot\|$ denotes the Euclidean norm. The SOCP is a special class of convex optimization problems involving SOC constraints, which can be efficiently solved by interior point methods. The work related to SOCP can be seen, for example, in [1, 8, 13, 15] and the references therein.

The paper is organized as follows. We first propose a robust formulation for piecewise-linear M-SVM in Section 2, and then construct a robust classifier based on the dual SOCP formulation in Section 3. In Section 4, we extend the robust classifier to the piecewise-nonlinear M-SVM case. Section 5 gives numerical results. The last section concludes the paper.

2 Robust Piecewise-Linear M-SVM Formulation

For each i, let $\mathcal{A}^i$ be a set of examples in the N-dimensional real space $\mathcal{R}^N$ with cardinality l_i. Let A^i be an $l_i \times N$ matrix whose rows are the examples in $\mathcal{A}^i$. The pth example in $\mathcal{A}^i$ and the pth row of A^i are both denoted A^i_p. Let $\boldsymbol{e}^i$ denote the vector of ones of dimension l_i. For each i, let $\boldsymbol{w}^i$ be a vector in $\mathcal{R}^N$ and b^i be a real number. The sets $\mathcal{A}^i$, $i = 1, \cdots, K$, are called piecewise-linearly separable [5] if there exist $\boldsymbol{w}^i$ and b^i, $i = 1, \cdots, K$, such that

$$A^i\boldsymbol{w}^i - b^i\boldsymbol{e}^i > A^i\boldsymbol{w}^j - b^j\boldsymbol{e}^i, \quad i, j = 1, \cdots, K, \quad i \neq j.$$

Piecewise-linear M-SVM can be formulated as follows [5]:

$$\begin{array}{ll}\min\limits_{\boldsymbol{w},\,\boldsymbol{b},\,\boldsymbol{y}} & \nu\left(\frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{i-1}\|\boldsymbol{w}^i - \boldsymbol{w}^j\|^2 + \frac{1}{2}\sum_{i=1}^{K}\|\boldsymbol{w}^i\|^2\right) + (1-\nu)\sum_{i=1}^{K}\sum_{j=1, j\neq i}^{K}(\boldsymbol{e}^i)^T\boldsymbol{y}^{ij} \\ \text{s.t.} & A^i(\boldsymbol{w}^i - \boldsymbol{w}^j) - (b^i - b^j)\boldsymbol{e}^i + \boldsymbol{y}^{ij} \geq \boldsymbol{e}^i, \\ & \boldsymbol{y}^{ij} \geq \boldsymbol{0}, \qquad i, j = 1, \cdots, K, \quad i \neq j,\end{array} \tag{1}$$

where $\nu \in (0, 1]$,

$$\boldsymbol{w} = \left[(\boldsymbol{w}^1)^T, (\boldsymbol{w}^2)^T, \cdots, (\boldsymbol{w}^K)^T\right]^T \in \mathcal{R}^{KN}, \tag{2}$$

$$\boldsymbol{b} = \left[b^1, b^2, \cdots, b^K\right]^T \in \mathcal{R}^K, \tag{3}$$

$$\boldsymbol{y} = \left[(\boldsymbol{y}^{12})^T, \cdots, (\boldsymbol{y}^{1K})^T, \cdots, (\boldsymbol{y}^{K1})^T, \cdots, (\boldsymbol{y}^{K(K-1)})^T\right]^T \in \mathcal{R}^{L(K-1)}, \tag{4}$$

and $L = \sum_{i=1}^{K} l_i$. When $\nu = 1$, (1) is the formulation for the piecewise-linearly separable case. Otherwise, it is the formulation for the piecewise-linearly inseparable case. Figure 1 shows an example of a piecewise-linearly separable M-SVM for three classes in two dimensions.

The training data A^i, $i = 1, \cdots, K$, used in problem (1) are implicitly assumed to be known exactly. However, in practice, training data are often corrupted by measurement noises. Errors in the input space tend to get amplified in the decision function, which

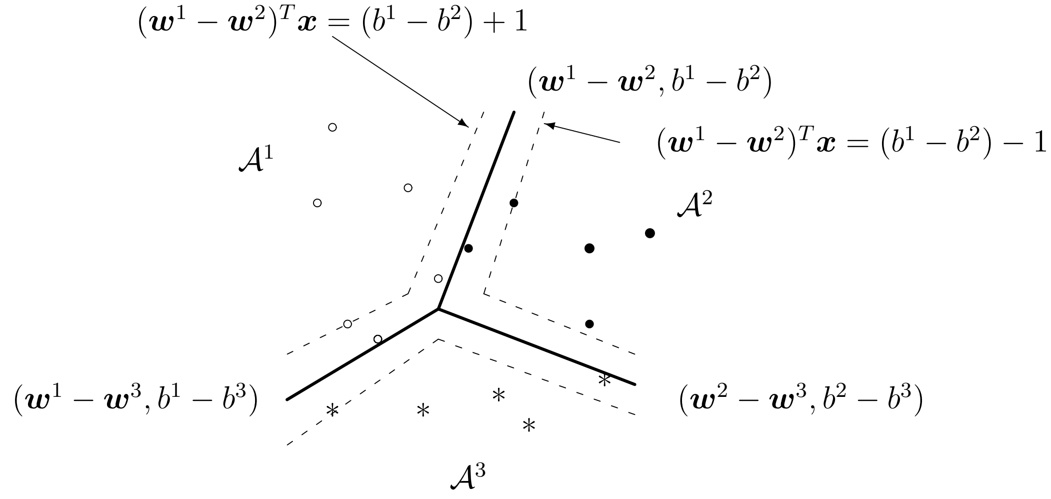
$(\boldsymbol{w}^1 - \boldsymbol{w}^2)^T\boldsymbol{x} = (b^1 - b^2) + 1$
$(\boldsymbol{w}^1 - \boldsymbol{w}^2, b^1 - b^2)$
$(\boldsymbol{w}^1 - \boldsymbol{w}^2)^T\boldsymbol{x} = (b^1 - b^2) - 1$
$\mathcal{A}^1$
$\mathcal{A}^2$
$(\boldsymbol{w}^1 - \boldsymbol{w}^3, b^1 - b^3)$
$(\boldsymbol{w}^2 - \boldsymbol{w}^3, b^2 - b^3)$
$\mathcal{A}^3$

Figure 1: Three classes separated by piecewise-linear M-SVM discriminants.

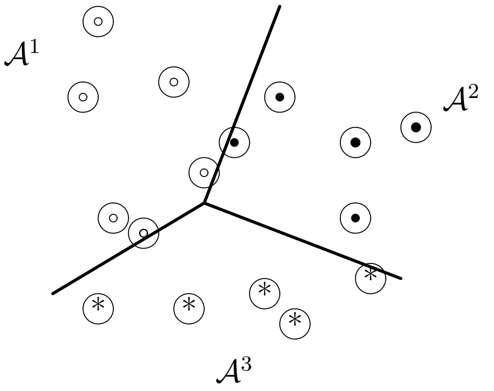

Figure 2: An example of the effect of measurement noises.

often results in misclassification. For example, suppose each example in Figure 1 is allowed to move in a sphere (see Figure 2). The original discriminants cannot separate the training data sets in the worst case. It will be useful to explore formulations which can yield discriminants robust to such estimation errors. In the following, we discuss such a formulation.

We assume

$$\hat{A}_p^i = A_p^i + \rho_p^i (\boldsymbol{a}_p^i)^T, \tag{5}$$

where $\hat{A}_p^i$ is the actual value of the training data, and $\rho_p^i(\boldsymbol{a}_p^i)^T$ is the measurement noise with $\boldsymbol{a}_p^i \in \mathcal{R}^N$, $\|\boldsymbol{a}_p^i\| = 1$ and $\rho_p^i \geq 0$ being a given constant. Denote the unit sphere in $\mathcal{R}^N$ by $\mathcal{U} = \{\boldsymbol{a} \in \mathcal{R}^N : \ \|\boldsymbol{a}\| = 1\}$. The robust version of formulation (1) can be stated as follows:

$$\begin{array}{ll} \min\limits_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{y}} & \nu \left(\frac{1}{2} \sum\limits_{i=1}^{K} \sum\limits_{j=1}^{i-1} \|\boldsymbol{w}^i - \boldsymbol{w}^j\|^2 + \frac{1}{2} \sum\limits_{i=1}^{K} \|\boldsymbol{w}^i\|^2 \right) + (1-\nu) \sum\limits_{i=1}^{K} \sum\limits_{j=1, j\neq i}^{K} (\boldsymbol{e}^i)^T \boldsymbol{y}^{ij} \\ \text{s.t.} & A_p^i(\boldsymbol{w}^i - \boldsymbol{w}^j) + \rho_p^i (\boldsymbol{a}_p^i)^T (\boldsymbol{w}^i - \boldsymbol{w}^j) - (b^i - b^j) + y_p^{ij} \geq 1, \qquad \forall\, \boldsymbol{a}_p^i \in \mathcal{U}, \\ & y_p^{ij} \geq 0, \qquad p = 1, \cdots, l_i, \ i, j = 1, \cdots, K, \ i \neq j. \end{array} \tag{6}$$

Since

$$\min\{\rho_p^i (\boldsymbol{a}_p^i)^T (\boldsymbol{w}^i - \boldsymbol{w}^j) : \ \boldsymbol{a}_p^i \in \mathcal{U}\} = -\rho_p^i \|\boldsymbol{w}^i - \boldsymbol{w}^j\|,$$

problem (6) is equivalent to the following SOCP:

$$\begin{array}{ll} \min\limits_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{y}} & \nu \left(\frac{1}{2} \sum\limits_{i=1}^{K} \sum\limits_{j=1}^{i-1} \|\boldsymbol{w}^i - \boldsymbol{w}^j\|^2 + \frac{1}{2} \sum\limits_{i=1}^{K} \|\boldsymbol{w}^i\|^2 \right) + (1-\nu) \sum\limits_{i=1}^{K} \sum\limits_{j=1, j\neq i}^{K} (\boldsymbol{e}^i)^T \boldsymbol{y}^{ij} \\ \text{s.t.} & A_p^i(\boldsymbol{w}^i - \boldsymbol{w}^j) - \rho_p^i \|\boldsymbol{w}^i - \boldsymbol{w}^j\| - (b^i - b^j) + y_p^{ij} \geq 1, \\ & y_p^{ij} \geq 0, \qquad p = 1, \cdots, l_i, \ i, j = 1, \cdots, K, \ i \neq j. \end{array} \tag{7}$$

Denote

$$Q = (K+1) I_{KN} - \Phi$$

with I_{KN} being the identity matrix of order KN and

$$\Phi = \begin{bmatrix} I_N & I_N & \cdots & I_N \\ I_N & I_N & \cdots & I_N \\ \vdots & \vdots & \ddots & \vdots \\ I_N & I_N & \cdots & I_N \end{bmatrix} \in \mathcal{R}^{KN \times KN.}$$

Denote $\boldsymbol{e} = [\underbrace{(\boldsymbol{e}^1)^T, \cdots, (\boldsymbol{e}^1)^T}_{K-1}, \cdots, \underbrace{(\boldsymbol{e}^K)^T, \cdots, (\boldsymbol{e}^K)^T}_{K-1}]^T \in \mathcal{R}^{L(K-1)}$. The objective function of problem (7) can then be expressed compactly as

$$\frac{\nu}{2}\boldsymbol{w}^T Q \boldsymbol{w} + (1-\nu)\boldsymbol{e}^T \boldsymbol{y}. \tag{8}$$

Additionally, Q is a symmetric positive definite matrix, which can be inferred from the following proposition. The proof of the proposition is omitted since it is similar to that in [22].

Proposition 2.1 Denote $C = \sqrt{K+1} I_{KN} - \frac{\sqrt{K+1}-1}{K}\Phi$. Then

(1). $Q = C^2$.

(2). C is nonsingular, and $C^{-1} = \frac{1}{\sqrt{K+1}} I_{KN} + \frac{\sqrt{K+1}-1}{K\sqrt{K+1}}\Phi$.

Let H^{ij} be the $KN \times N$ matrix with all blocks being $N \times N$ zero matrices except the ith block being I_N and the jth block being $-I_N$:

$$H^{ij} = [\mathrm{O}, \cdots, \mathrm{O}, I_N, \mathrm{O}, \cdots, \mathrm{O}, -I_N, \mathrm{O}, \cdots, \mathrm{O}]^T. \tag{9}$$

Then by (2) we get

$$\boldsymbol{w}^i - \boldsymbol{w}^j = (H^{ij})^T \boldsymbol{w}. \tag{10}$$

Let $\boldsymbol{r}^{ij}$ be the K-dimensional vector with all components being zero except the ith component being 1 and the jth component being -1:

$$\boldsymbol{r}^{ij} = [0, \cdots, 0, 1, 0, \cdots, 0, -1, 0, \cdots, 0]^T. \tag{11}$$

Then by (3) we get

$$b^i - b^j = (\boldsymbol{r}^{ij})^T \boldsymbol{b}. \tag{12}$$

Let $\boldsymbol{h}_p^{ij}$ be the $L(K-1)$-dimensional vector with all components being zero except the $\Big((K-1)\sum_{k=1}^{i-1} l_k + (j-1)l_i + p\Big)$th component being 1:

$$\boldsymbol{h}_p^{ij} = [0, \cdots, 0, \cdots, 0, 1, 0, \cdots, 0, \cdots, 0]^T. \tag{13}$$

Then by (4) we get

$$y_p^{ij} = (\boldsymbol{h}_p^{ij})^T \boldsymbol{y}. \tag{14}$$

By (10), (12) and (14), the first constraint in problem (7) can be rewritten as follows:

$$\rho_p^i \|(H^{ij})^T \boldsymbol{w}\| \le A_p^i (H^{ij})^T \boldsymbol{w} - (\boldsymbol{r}^{ij})^T \boldsymbol{b} + (\boldsymbol{h}_p^{ij})^T \boldsymbol{y} - 1. \tag{15}$$

Therefore, by (8), (15) and Proposition 2.1, formulation (7) can be written as follows:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{y},t} \quad & \nu t + (1-\nu)\boldsymbol{e}^T\boldsymbol{y} \\ \text{s.t.} \quad & \frac{1}{2}\|C\boldsymbol{w}\|^2 \le t, \\ & \rho_p^i\|(H^{ij})^T\boldsymbol{w}\| \le A_p^i(H^{ij})^T\boldsymbol{w} - (\boldsymbol{r}^{ij})^T\boldsymbol{b} + (\boldsymbol{h}_p^{ij})^T\boldsymbol{y} - 1, \\ & \qquad\qquad p = 1,\cdots,l_i,\ i,j = 1,\cdots,K,\ i \neq j, \\ & \boldsymbol{y} \ge \boldsymbol{0}. \end{aligned} \tag{16}$$

Furthermore, formulation (16) can be cast as the following SOCP:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{y},t} \quad & \nu t + (1-\nu)\boldsymbol{e}^T\boldsymbol{y} \\ \text{s.t.} \quad & \left\| \begin{bmatrix} \sqrt{2}C\boldsymbol{w} \\ 1-t \end{bmatrix} \right\| \le 1+t, \\ & \rho_p^i\|(H^{ij})^T\boldsymbol{w}\| \le A_p^i(H^{ij})^T\boldsymbol{w} - (\boldsymbol{r}^{ij})^T\boldsymbol{b} + (\boldsymbol{h}_p^{ij})^T\boldsymbol{y} - 1, \\ & \qquad\qquad p = 1,\cdots,l_i,\ i,j = 1,\cdots,K,\ i \neq j, \\ & \boldsymbol{y} \ge \boldsymbol{0}. \end{aligned} \tag{17}$$

3 Robust Piecewise-Linear M-SVM Classifier

In this section, we construct a robust piecewise-linear M-SVM classifier based on the dual formulation of (17).

3.1 Dual of the Robust Piecewise-Linear M-SVM Formulation

Denote

$$\bar{A} = [B_1^T, B_2^T, \cdots, B_K^T]^T \in \mathcal{R}^{L(K-1)\times KN} \tag{18}$$

with

$$B_i = \begin{bmatrix} -A^i & \cdots & \mathrm{O} & A^i & \mathrm{O} & \cdots & \mathrm{O} \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ \mathrm{O} & \cdots & -A^i & A^i & \mathrm{O} & \cdots & \mathrm{O} \\ \mathrm{O} & \cdots & \mathrm{O} & A^i & -A^i & \cdots & \mathrm{O} \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{O} & \cdots & \mathrm{O} & A^i & \mathrm{O} & \cdots & -A^i \end{bmatrix} \in \mathcal{R}^{l_i(K-1)\times KN}.$$

Denote

$$\bar{H} = [\bar{M}_1^T, \bar{M}_2^T \cdots, \bar{M}_K^T]^T \in \mathcal{R}^{LN(K-1)\times KN}, \tag{19}$$

where

$$\bar{M}_i = \begin{bmatrix} -M_i & \cdots & \mathrm{O} & M_i & \mathrm{O} & \cdots & \mathrm{O} \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ \mathrm{O} & \cdots & -M_i & M_i & \mathrm{O} & \cdots & \mathrm{O} \\ \mathrm{O} & \cdots & \mathrm{O} & M_i & -M_i & \cdots & \mathrm{O} \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ \mathrm{O} & \cdots & \mathrm{O} & M_i & \mathrm{O} & \cdots & -M_i \end{bmatrix} \in \mathcal{R}^{l_i N(K-1)\times KN},$$

with

$$M_i := M_i(\boldsymbol{\rho}) = [\rho_1^i I_N, \cdots, \rho_{l_i}^i I_N]^T \in \mathcal{R}^{l_i N \times N}, \quad i = 1, \cdots, K.$$

Denote

$$\bar{E} = [E_1^T, E_2^T, \cdots, E_K^T]^T \in \mathcal{R}^{L(K-1)\times K} \tag{20}$$

with

$$E_i = \begin{bmatrix} -\boldsymbol{e}^i & \cdots & \boldsymbol{0} & \boldsymbol{e}^i & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ \boldsymbol{0} & \cdots & -\boldsymbol{e}^i & \boldsymbol{e}^i & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{e}^i & -\boldsymbol{e}^i & \cdots & \boldsymbol{0} \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{e}^i & \boldsymbol{0} & \cdots & -\boldsymbol{e}^i \end{bmatrix} \in \mathcal{R}^{l_i(K-1)\times K}.$$

We can derive the following dual of problem (17) (see Appendix A):

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{s},\sigma,\tau} \quad & \boldsymbol{e}^T\boldsymbol{\alpha} - (\sigma + \tau) \\ \text{s.t.} \quad & \bar{E}^T\boldsymbol{\alpha} = \boldsymbol{0}, \\ & \boldsymbol{\alpha} \le (1-\nu)\boldsymbol{e}, \\ & \sigma - \tau = \nu, \\ & \left\| \begin{bmatrix} -\frac{1}{\sqrt{2(K+1)}}(\bar{A}^T\boldsymbol{\alpha} + \bar{H}^T\boldsymbol{s}) \\ \tau \end{bmatrix} \right\| \le \sigma, \\ & \|\boldsymbol{s}_p^{ij}\| \le \alpha_p^{ij}, \quad p = 1, \cdots, l_i, \; i, j = 1, \cdots, K, \; j \ne i, \end{aligned} \tag{21}$$

where

$$\boldsymbol{\alpha} = [(\boldsymbol{\alpha}^{12})^T, \cdots, (\boldsymbol{\alpha}^{1K})^T, \cdots, (\boldsymbol{\alpha}^{K1})^T, \cdots, (\boldsymbol{\alpha}^{K(K-1)})^T]^T \in \mathcal{R}^{L(K-1)}, \tag{22}$$

$$\begin{aligned} \boldsymbol{s} = \Big[&(\boldsymbol{s}_1^{12})^T, \cdots, (\boldsymbol{s}_{l_1}^{12})^T, \cdots, (\boldsymbol{s}_1^{1K})^T, \cdots, (\boldsymbol{s}_{l_1}^{1K})^T, \cdots, \\ &\cdots, (\boldsymbol{s}_1^{K(K-1)})^T, \cdots, (\boldsymbol{s}_{l_K}^{K(K-1)})^T\Big]^T \in \mathcal{R}^{LN(K-1)}. \end{aligned} \tag{23}$$

In addition, we get the following complementary equations at optimality:

$$\begin{bmatrix} \alpha_p^{ij} \\ \boldsymbol{s}_p^{ij} \end{bmatrix}^T \begin{bmatrix} A_p^i (H^{ij})^T \boldsymbol{w} - (\boldsymbol{r}^{ij})^T \boldsymbol{b} + (\boldsymbol{h}_p^{ij})^T \boldsymbol{y} - 1 \\ \rho_p^i (H^{ij})^T \boldsymbol{w} \end{bmatrix} = 0, \tag{24}$$

$$p = 1, \cdots, l_i, \ i, j = 1, \cdots, K, \ j \neq i,$$

$$\begin{bmatrix} \sigma \\ -\frac{1}{\sqrt{2(K+1)}} (\bar{A}^T \boldsymbol{\alpha} + \bar{H}^T \boldsymbol{s}) \\ \tau \end{bmatrix}^T \begin{bmatrix} 1 + t \\ \sqrt{2} C \boldsymbol{w} \\ 1 - t \end{bmatrix} = 0, \tag{25}$$

$$((1 - \nu)\boldsymbol{e} - \boldsymbol{\alpha})^T \boldsymbol{y} = 0. \tag{26}$$

3.2 Robust Classifier

From formulation (21) we get $\sigma > 0$. In fact, if $\sigma = 0$, then $\tau = 0$. The third constraint of formulation (21) becomes $\nu = 0$, which contradicts $\nu > 0$. By the complementary equation (25), we have the following implications (see Appendix B for the complementary conditions in SOCP, i.e., (57)–(59)):

If $\left\| \begin{bmatrix} -\frac{1}{\sqrt{2(K+1)}} (\bar{A}^T \boldsymbol{\alpha} + \bar{H}^T \boldsymbol{s}) \\ \tau \end{bmatrix} \right\| < \sigma$, then $\left\| \begin{bmatrix} \sqrt{2} C \boldsymbol{w} \\ 1 - t \end{bmatrix} \right\| = 1 + t = 0$. But this contradicts $t \geq 0$. So we must have $\left\| \begin{bmatrix} -\frac{1}{\sqrt{2(K+1)}} (\bar{A}^T \boldsymbol{\alpha} + \bar{H}^T \boldsymbol{s}) \\ \tau \end{bmatrix} \right\| = \sigma$. Since $\sigma > 0$, we have $\left\| \begin{bmatrix} \sqrt{2} C \boldsymbol{w} \\ 1 - t \end{bmatrix} \right\| = 1 + t$. Hence there exists $\mu > 0$ such that

$$\sqrt{2} C \boldsymbol{w} = \frac{\mu}{\sqrt{2(K+1)}} (\bar{A}^T \boldsymbol{\alpha} + \bar{H}^T \boldsymbol{s}) \quad \text{and} \quad 1 - t = -\mu\tau. \tag{27}$$

In addition, it is easy to get the following equalities by Proposition 2.1:

$$C^{-1} \bar{A}^T = \frac{1}{\sqrt{K+1}} \bar{A}^T \quad \text{and} \quad C^{-1} \bar{H}^T = \frac{1}{\sqrt{K+1}} \bar{H}^T. \tag{28}$$

Hence by (27) and (28), we get

$$\boldsymbol{w} = \frac{t-1}{2\tau(K+1)}(\bar{A}^T\boldsymbol{\alpha} + \bar{H}^T\boldsymbol{s}).$$

Furthermore, by (2), (18) and (19), we get

$$\boldsymbol{w}^i = \frac{t-1}{2\tau(K+1)} \sum_{j=1,j\neq i}^{K} \left\{ \left[\sum_{p=1}^{l_i} \alpha_p^{ij}(A_p^i)^T - \sum_{p=1}^{l_j} \alpha_p^{ji}(A_p^j)^T \right] + \left[\sum_{p=1}^{l_i} \rho_p^i \boldsymbol{s}_p^{ij} - \sum_{p=1}^{l_j} \rho_p^j \boldsymbol{s}_p^{ji} \right] \right\}.$$

Therefore, the decision functions are given by

$$\begin{aligned} f_i(\boldsymbol{x}) &= \boldsymbol{x}^T\boldsymbol{w}^i - b^i \\ &= \frac{t-1}{2\tau(K+1)} \sum_{j=1,j\neq i}^{K} \left\{ \left[\sum_{p=1}^{l_i} \alpha_p^{ij}\boldsymbol{x}^T(A_p^i)^T - \sum_{p=1}^{l_j} \alpha_p^{ji}\boldsymbol{x}^T(A_p^j)^T \right] + \right. \\ &\qquad \left. \left[\sum_{p=1}^{l_i} \rho_p^i \boldsymbol{x}^T\boldsymbol{s}_p^{ij} - \sum_{p=1}^{l_j} \rho_p^j \boldsymbol{x}^T\boldsymbol{s}_p^{ji} \right] \right\} - b^i,\ i = 1, \cdots, K. \quad (29) \end{aligned}$$

In particular, if we set $\rho_p^i = 0,\ i = 1, \cdots, K,\ p = 1, \cdots, l_i$, then (29) becomes

$$f_i(\boldsymbol{x}) = \frac{t-1}{2\tau(K+1)} \sum_{j=1,j\neq i}^{K} \left[\sum_{p=1}^{l_i} \alpha_p^{ij}\boldsymbol{x}^T(A_p^i)^T - \sum_{p=1}^{l_j} \alpha_p^{ji}\boldsymbol{x}^T(A_p^j)^T \right] - b^i,\ i = 1, \cdots, K. \quad (30)$$

Since $\rho_p^i = 0,\ p = 1, \cdots, l_i,\ i = 1, \cdots, K$, imply that the parameter perturbations are not considered (cf. (5)), (30) corresponds to the discriminants for the case of no measurement noise.

With these decision functions, the classification of an example $\boldsymbol{x}$ is to find a class i such that $f_i(\boldsymbol{x}) = \max\{f_1(\boldsymbol{x}), \cdots, f_K(\boldsymbol{x})\}$.

4 Robust Piecewise-Nonlinear M-SVM Classifier

The above discussion is concerned with the piecewise-linear case. In this section, the analysis will be extended to the nonlinear case.

To construct separating functions in a higher dimensional feature space, a nonlinear mapping $\psi : \mathcal{X} \to \mathcal{F}$ is used to transform the original examples into the feature space which is equipped with the inner product defined by

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \psi(\boldsymbol{x}), \psi(\boldsymbol{x}') \rangle,$$

where $k(\cdot,\cdot) : \mathcal{R}^N \times \mathcal{R}^N \rightarrow \mathcal{R}$ is a function called a kernel. Typical choices of kernels include polynomial kernels $k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T\boldsymbol{x}' + 1)^d$ with an integer parameter d and RBF kernels $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2/\kappa)$ with a real parameter κ.

4.1 Robust Piecewise-Nonlinear M-SVM Formulation

We assume

$$\psi((\hat{A}^i_p)^T) = \psi((A^i_p)^T) + \tilde{\rho}^i_p \tilde{\boldsymbol{a}}^i_p, \quad \tilde{\boldsymbol{a}}^i_p \in \tilde{\mathcal{U}}, \tag{31}$$

where $\tilde{\mathcal{U}}$ is a unit sphere in the feature space. For the nonlinear case, $\tilde{\rho}^i_p$ in the feature space associated with a kernel $k(\cdot,\cdot)$ can be computed as

$$\begin{aligned} \tilde{\rho}^i_p &= \|\psi((\hat{A}^i_p)^T) - \psi((A^i_p)^T)\| \\ &= \Big(\langle\psi((\hat{A}^i_p)^T), \psi((\hat{A}^i_p)^T)\rangle - 2\langle\psi((\hat{A}^i_p)^T), \psi((A^i_p)^T)\rangle + \langle\psi((A^i_p)^T), \psi((A^i_p)^T)\rangle\Big)^{1/2} \\ &= \Big(k((\hat{A}^i_p)^T, (\hat{A}^i_p)^T) - 2k((\hat{A}^i_p)^T, (A^i_p)^T) + k((A^i_p)^T, (A^i_p)^T)\Big)^{1/2}. \end{aligned}$$

For example, for RBF kernels, since

$$k((\hat{A}^i_p)^T, (\hat{A}^i_p)^T) = 1, \quad k((\hat{A}^i_p)^T, (A^i_p)^T) = \exp(-(\rho^i_p)^2/\kappa), \quad \text{and} \quad k((A^i_p)^T, (A^i_p)^T) = 1,$$

we have

$$\tilde{\rho}^i_p = \left(2 - 2\exp(-(\rho^i_p)^2/\kappa)\right)^{1/2}. \tag{32}$$

The robust version of the piecewise-nonlinear M-SVM can be expressed as follows:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{y}} \quad & \nu\left(\frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{i-1}\|\boldsymbol{w}^i - \boldsymbol{w}^j\|^2 + \frac{1}{2}\sum_{i=1}^{K}\|\boldsymbol{w}^i\|^2\right) + (1-\nu)\sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}(\boldsymbol{e}^i)^T\boldsymbol{y}^{ij} \\ \text{s.t.} \quad & (\psi((A^i_p)^T))^T(\boldsymbol{w}^i - \boldsymbol{w}^j) + \tilde{\rho}^i_p(\tilde{\boldsymbol{a}}^i_p)^T(\boldsymbol{w}^i - \boldsymbol{w}^j) - (b^i - b^j) + y^{ij}_p \geq 1, \ \forall \tilde{\boldsymbol{a}}^i_p \in \tilde{\mathcal{U}}, \\ & y^{ij}_p \geq 0, \quad p = 1, \cdots, l_i, \ i, j = 1, \cdots, K, \ i \neq j, \end{aligned}$$

which can be rewritten as the following SOCP:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{y}} \quad & \nu\left(\frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{i-1}\|\boldsymbol{w}^i - \boldsymbol{w}^j\|^2 + \frac{1}{2}\sum_{i=1}^{K}\|\boldsymbol{w}^i\|^2\right) + (1-\nu)\sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}(\boldsymbol{e}^i)^T\boldsymbol{y}^{ij} \\ \text{s.t.} \quad & (\psi((A^i_p)^T))^T(\boldsymbol{w}^i - \boldsymbol{w}^j) - \tilde{\rho}^i_p\|\boldsymbol{w}^i - \boldsymbol{w}^j\| - (b^i - b^j) + y^{ij}_p \geq 1, \\ & y^{ij}_p \geq 0, \quad p = 1, \cdots, l_i, \ i, j = 1, \cdots, K, \ i \neq j. \end{aligned} \tag{33}$$

Denote

$$\tilde{A} = \left[\tilde{B}_1^T, \tilde{B}_2^T, \cdots, \tilde{B}_K^T\right]^T,$$

where

$$\tilde{B}_i = \begin{bmatrix} -\Psi(A^i) & \cdots & \text{O} & \Psi(A^i) & \text{O} & \cdots & \text{O} \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ \text{O} & \cdots & -\Psi(A^i) & \Psi(A^i) & \text{O} & \cdots & \text{O} \\ \text{O} & \cdots & \text{O} & \Psi(A^i) & -\Psi(A^i) & \cdots & \text{O} \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ \text{O} & \cdots & \text{O} & \Psi(A^i) & \text{O} & \cdots & -\Psi(A^i) \end{bmatrix}$$

with

$$\Psi(A^i) = \left[\psi((A_1^i)^T), \cdots, \psi((A_{l_i}^i)^T)\right]^T.$$

Denote

$$\tilde{H} = [\tilde{M}_1^T, \tilde{M}_2^T \cdots, \tilde{M}_K^T]^T,$$

where

$$\tilde{M}_i = \begin{bmatrix} -M_i' & \cdots & \text{O} & M_i' & \text{O} & \cdots & \text{O} \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ \text{O} & \cdots & -M_i' & M_i' & \text{O} & \cdots & \text{O} \\ \text{O} & \cdots & \text{O} & M_i' & -M_i' & \cdots & \text{O} \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ \text{O} & \cdots & \text{O} & M_i' & \text{O} & \cdots & -M_i' \end{bmatrix}$$

with

$$M_i' := M_i'(\tilde{\boldsymbol{\rho}}) = [\tilde{\rho}_1^i I_N, \cdots, \tilde{\rho}_{l_i}^i I_N]^T. \tag{34}$$

In a similar manner to that of getting formulation (21), we get the dual of problem (33) as follows:

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{s},\sigma,\tau} \quad & \boldsymbol{e}^T\boldsymbol{\alpha} - (\sigma+\tau) \\ \text{s.t.} \quad & \bar{E}^T\boldsymbol{\alpha} = \mathbf{0}, \\ & \boldsymbol{\alpha} \le (1-\nu)\boldsymbol{e}, \\ & \sigma - \tau = \nu, \\ & \left\| \begin{bmatrix} -\frac{1}{\sqrt{2(K+1)}}(\tilde{A}^T\boldsymbol{\alpha} + \tilde{H}^T\boldsymbol{s}) \\ \tau \end{bmatrix} \right\| \le \sigma, \\ & \|\boldsymbol{s}_p^{ij}\| \le \alpha_p^{ij}, \quad p = 1,\cdots,l_i,\ \ i,j = 1,\cdots,K,\ \ j \ne i. \end{aligned} \tag{35}$$

4.2 Robust Classifier in a Feature Subspace

In the previous subsection, we have gotten the robust formulation (35) in the feature space. However, the feature space $\mathcal{F}$ may have an arbitrarily large dimension, possibly infinite. Usually the kernel principal component analysis (KPCA) [18, 22] is used for feature extraction. In this subsection, we first reduce the feature space $\mathcal{F}$ to an S-dimensional subspace with $S < L$ by KPCA, and then construct the corresponding robust classifier of piecewise-nonlinear M-SVM in the subspace.

Consider the kernel matrix $G = (k((A_p^i)^T, (A_p^j)^T)) \in \mathcal{R}^{L\times L}$ associated with a kernel $k(\cdot,\cdot)$. Since G is a symmetric positive semi-definite matrix, there is an orthogonal matrix V such that $G = V\Lambda V^T$, where Λ is a diagonal matrix whose diagonal elements are the eigenvalues $\lambda_i \geq 0$, $i = 1, \cdots, L$, of G, and $\boldsymbol{v}_i$, $i = 1, \cdots, L$, the columns of V, are the corresponding eigenvectors. Suppose $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L$. Select the $S(< L)$ largest positive eigenvalues and the corresponding eigenvectors. Denote $D_S = \left[\sqrt{\lambda_1}\boldsymbol{v}_1, \sqrt{\lambda_2}\boldsymbol{v}_2, \cdots, \sqrt{\lambda_S}\boldsymbol{v}_S\right]$, where the components of $\boldsymbol{v}_i$ are written as follows:

$$\boldsymbol{v}_i = [v_{i,1}^1, \cdots, v_{i,l_1}^1, v_{i,1}^2, \cdots, v_{i,l_2}^2, \cdots, v_{i,1}^K, \cdots, v_{i,l_K}^K]^T.$$

Define the vectors

$$\boldsymbol{u}_i := \frac{\sum_{j=1}^K \sum_{p=1}^{l_j} v_{i,p}^j\, \psi((A_p^j)^T)}{\sqrt{\lambda_i}},\ i = 1, \cdots, S.$$

Then we have

$$\boldsymbol{u}_i{}^T \boldsymbol{u}_i = \frac{1}{\lambda_i} \boldsymbol{v}_i^T G \boldsymbol{v}_i = 1,$$

and

$$\boldsymbol{u}_i{}^T \boldsymbol{u}_j = \frac{1}{\sqrt{\lambda_i \lambda_j}} \boldsymbol{v}_i^T G \boldsymbol{v}_j = 0, \quad i \neq j.$$

Therefore, $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_S\}$ forms an orthogonal basis of an S-dimensional subspace of $\mathcal{F}$. Let $\psi_S(\boldsymbol{x})$ be the S-dimensional sub-coordinate of $\psi(\boldsymbol{x})$, which is given by

$$\psi_S(\boldsymbol{x}) = \left(\frac{1}{\sqrt{\lambda_1}} \sum_{j=1}^K \sum_{p=1}^{l_j} v_{1,p}^j\, k(\boldsymbol{x}, (A_p^j)^T), \cdots, \frac{1}{\sqrt{\lambda_S}} \sum_{j=1}^K \sum_{p=1}^{l_j} v_{S,p}^j\, k(\boldsymbol{x}, (A_p^j)^T) \right)^T. \tag{36}$$

Then, similarly to (29), we can get the decision functions associated with the robust formulation of piecewise-nonlinear M-SVM in the feature subspace as follows:

$$
\begin{aligned}
f_i(\boldsymbol{x}) \;&=\; \frac{t-1}{2\tau(K+1)} \sum_{j=1,j\neq i}^{K} \Bigg\{ \Bigg[\sum_{p=1}^{l_i} \alpha_p^{ij} \psi_S(\boldsymbol{x})^T \psi_S((A_p^i)^T) - \sum_{p=1}^{l_j} \alpha_p^{ji} \psi_S(\boldsymbol{x})^T \psi_S((A_p^j)^T) \Bigg] \\
&\qquad + \Bigg[\sum_{p=1}^{l_i} \tilde{\rho}_p^i \psi_S(\boldsymbol{x})^T \boldsymbol{s}_p^{ij} - \sum_{p=1}^{l_j} \tilde{\rho}_p^j \psi_S(\boldsymbol{x})^T \boldsymbol{s}_p^{ji} \Bigg] \Bigg\} - b^i, \; i = 1, \cdots, K. \qquad (37)
\end{aligned}
$$

5 Preliminary Numerical Results

In this section, through numerical experiments, we examine the performance of the robust piecewise-nonlinear M-SVM formulation and the original model for multi-class classification problems. We use RBF kernel in the experiments. As we have described in Subsection 4.2, we first construct an $L \times L$ kernel matrix G associated with the RBF kernel for the training dataset. Then we decompose G and select an appropriate number S. Using (36), we obtain the S-dimensional sub-coordinate of each point. The problems used in the experiments are the robust model (35) and the original model obtained by setting $\tilde{\boldsymbol{\rho}} = \mathbf{0}$ in (34). In the latter model, we have $\tilde{H} = \mathrm{O}$. Thus we may write the problem as follows:

$$
\begin{aligned}
\max_{\boldsymbol{\alpha},\sigma,\tau} \quad & \boldsymbol{e}^T\boldsymbol{\alpha} - (\sigma + \tau) \\
\text{s.t.} \quad & \bar{E}^T\boldsymbol{\alpha} = \mathbf{0}, \\
& \boldsymbol{\alpha} \leq (1-\nu)\boldsymbol{e}, \qquad\qquad (38) \\
& \sigma - \tau = \nu, \\
& \left\| \begin{bmatrix} -\frac{1}{\sqrt{2(K+1)}}\tilde{A}^T\boldsymbol{\alpha} \\ \tau \end{bmatrix} \right\| \leq \sigma.
\end{aligned}
$$

Table 1: Description of Iris, Wine and Glass datasets.

name	dimension (N)	#classes (K)	#examples (L)
Iris	4	3	150
Wine	13	3	178
Glass	9	6	214

Table 2: Results for Iris, Wine and Glass datasets with noise ($\rho = 0.3$, $\kappa = 2$, $\nu = 0.05$).

R_a	Robust (I)	Iris			Wine			Glass		
	Original (II)	S	R_t	PT*	S	R_t	PT	S	R_t	PT
0.5	I	1	0.5364	62.67	5	0.5179	90.0	1	0.5737	35.24
	II	1	0.5364	60.67	5	0.5179	88.89	1	0.5737	31.43
0.6	I	2	0.7950	89.33	9	0.6102	88.89	2	0.6826	66.67
	II	2	0.7950	87.33	9	0.6102	80.0	2	0.6826	32.86
0.7	I	2	0.7950	89.33	16	0.7103	87.78	3	0.7523	66.67
	II	2	0.7950	87.33	16	0.7103	82.22	3	0.7523	38.57
0.8	I	3	0.8836	85.33	—	—	—	4	0.8002	66.67
	II	3	0.8836	84.0	—	—	—	4	0.8002	45.24
0.99	I	12	0.9911	88.0	—	—	—	—	—	—
	II	12	0.9911	86.67	—	—	—	—	—	—

*PT: Percentage of tenfold testing correctness on validation set.

The experiments were implemented on a PC (1GB RAM, CPU 3.00GHz) using SeDuMi1.05 [19] as a solver. This solver is developed by J. Sturm for optimization problems over symmetric cones including SOCP. Some experimental results on real-world datasets taken from the UCI machine learning repository [6] are reported below. Table 1 gives a description of the datasets. In the experiments, the datasets were normalized to lie in between -1 and 1. For simplicity, we set all ρ_p^i in (5) to be a constant ρ. The measurement noise $\boldsymbol{a}_p^i$ was generated randomly from the normal distribution and scaled on the unit sphere. Two experiments were performed. In the first, an appropriate value of S for getting reasonable discriminants was sought. The second experiment was conducted on the three datasets with the measurement noise. Ten-fold cross validation was used in the experiments.

In order to seek an appropriate value of S, a ratio R_a is set. It is chosen from the set $\{0.5, 0.6, 0.7, 0.8, 0.99\}$. For each value of R_a, we find the smallest integer S such that $\sum_{i=1}^{S} \lambda_i / \sum_{i=1}^{L} \lambda_i \geq R_a$ and let $R_t := \sum_{i=1}^{S} \lambda_i / \sum_{i=1}^{L} \lambda_i$. At the same time, we test the accuracy on the validation set by computing the percentage of tenfold testing correctness. Table 2 contains these three kinds of results for the robust model and the original model

Table 3: Percentage of tenfold test correctness for the datasets with noise ($\kappa = 2$, $\nu = 0.05$).

Dataset	Robust (I)	ρ				
(S)	Original (II)	0.1	0.2	0.3	0.4	0.5
Iris	I	88.67	88.0	89.33	91.33	90.0
(2)	II	87.33	87.33	87.33	86.67	85.33
Wine	I	91.11	90.0	90.0	87.78	84.44
(5)	II	88.89	88.83	88.89	85.56	82.22
Glass	I	66.19	65.71	66.67	66.67	66.67
(4)	II	46.19	45.71	45.24	49.05	49.52

on Iris, Wine and Glass datasets with the measurement noise scaled by $\rho = 0.3$. When R_a is large, we were unable to solve the problems for Wine and Glass datasets because of memory limitations. Nevertheless, it can be seen from Table 2 that the values of R_t around 50% up to 70% yield reasonable discriminants. Moreover, in all cases, S is much smaller than the data size L.

Table 3 shows the percentage of tenfold testing correctness for the robust model and the original model on the three datasets with various noise levels ρ. It can be observed that the performance of the robust model is consistently better than that of the original model, especially for Glass dataset.

6 Conclusion

In this paper, we have established the robust linear and nonlinear formulations for multi-class classification based on M-SVM method. KPCA has been used to reduce the feature space to an S-dimensional subspace. The preliminary numerical experiments show that the performance of the robust model is better than that of the original model.

Unfortunately, the conic convex optimization solver SeDuMi1.05 [19] used in our numerical experiments could only solve problems for small datasets. The sequential minimal optimization (SMO) techniques [16] are essential in large-scale implementation of SVM. The future subjects include developing SMO-based robust algorithms for multi-class classification.

Appendix A. Dual of Formulation (17)

In order to get the dual of problem (17), we first state a more general primal and dual form of the SOCP. The notations used in *A.1* are independent of those in the other part of the paper.

A.1. A General Primal and Dual Pair

For the SOCP

$$\begin{aligned} \min_{\boldsymbol{x},\,\boldsymbol{y},\,\boldsymbol{z}} \quad & \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{y} + \boldsymbol{e}^T\boldsymbol{z} \\ \text{s.t.} \quad & A^T\boldsymbol{x} + B^T\boldsymbol{y} + C^T\boldsymbol{z} = \boldsymbol{f}, \\ & \boldsymbol{y} \geq \boldsymbol{0}, \\ & \boldsymbol{z} \in \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_l}, \end{aligned} \tag{39}$$

its dual is written as follows:

$$\begin{aligned} \max_{\boldsymbol{w},\boldsymbol{u},\boldsymbol{v}} \quad & \boldsymbol{f}^T\boldsymbol{w} \\ \text{s.t.} \quad & A\boldsymbol{w} = \boldsymbol{c}, \\ & B\boldsymbol{w} + \boldsymbol{u} = \boldsymbol{d}, \\ & C\boldsymbol{w} + \boldsymbol{v} = \boldsymbol{e}, \\ & \boldsymbol{u} \geq \boldsymbol{0}, \\ & \boldsymbol{v} \in \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_l}. \end{aligned} \tag{40}$$

Now consider the problem

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \quad & \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{y} \\ \text{s.t.} \quad & \|\bar{G}_i^T\boldsymbol{x} + \boldsymbol{q}_i\| \leq \boldsymbol{g}_i^T\boldsymbol{x} + \boldsymbol{h}_i^T\boldsymbol{y} + \boldsymbol{r}_i^T\boldsymbol{z} + a_i, \quad i = 1, \cdots, m, \\ & \boldsymbol{y} \geq \boldsymbol{0}. \end{aligned} \tag{41}$$

This problem can be formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{\zeta}} \quad & \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{y} \\ \text{s.t.} \quad & \boldsymbol{\zeta}_i - \begin{bmatrix} \boldsymbol{g}_i^T\boldsymbol{x} + \boldsymbol{h}_i^T\boldsymbol{y} + \boldsymbol{r}_i^T\boldsymbol{z} + a_i \\ \bar{G}_i^T\boldsymbol{x} + \boldsymbol{q}_i \end{bmatrix} = \boldsymbol{0}, \qquad i = 1, \cdots, m, \\ & \boldsymbol{\zeta}_i \in \mathcal{K}^{n_i}, \qquad i = 1, \cdots, m, \\ & \boldsymbol{y} \geq \boldsymbol{0}, \end{aligned}$$

which can be further rewritten as

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{\zeta}} \quad & \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{y} \\
\text{s.t.} \quad & \begin{bmatrix} -G_1 & -G_2 & \cdots & -G_m \\ -H_1 & -H_2 & \cdots & -H_m \\ -R_1 & -R_2 & \cdots & -R_m \\ I & \mathrm{O} & \cdots & \mathrm{O} \\ \mathrm{O} & I & \cdots & \mathrm{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{O} & \mathrm{O} & \cdots & I \end{bmatrix}^T \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \\ \boldsymbol{\zeta}_1 \\ \boldsymbol{\zeta}_2 \\ \vdots \\ \boldsymbol{\zeta}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_m \end{bmatrix} \\
& \boldsymbol{\zeta}_i \in \mathcal{K}^{n_i}, \quad i = 1, \cdots, m, \\
& \boldsymbol{y} \geq \boldsymbol{0},
\end{aligned}$$

where $G_i = [\boldsymbol{g}_i, \bar{G}_i],\ H_i = [\boldsymbol{h}_i, \mathrm{O}],\ R_i = [\boldsymbol{r}_i, \mathrm{O}],\ \boldsymbol{\beta}_i = [a_i, \boldsymbol{q}_i^T]^T,\ i = 1, \cdots, m$. In view of the primal-dual pair (40) and (41), we obtain the dual of problem (41) as follows:

$$\begin{aligned}
\max_{\boldsymbol{\eta},\boldsymbol{\lambda}} \quad & -\sum_{i=1}^{m} \boldsymbol{\beta}_i^T \boldsymbol{\eta}_i \\
\text{s.t.} \quad & \sum_{i=1}^{m} G_i \boldsymbol{\eta}_i = \boldsymbol{c}, \\
& \sum_{i=1}^{m} H_i \boldsymbol{\eta}_i + \boldsymbol{\lambda} = \boldsymbol{d}, \\
& \sum_{i=1}^{m} R_i \boldsymbol{\eta}_i = \boldsymbol{0}, \\
& \boldsymbol{\lambda} \geq \boldsymbol{0}, \\
& \boldsymbol{\eta}_i \in \mathcal{K}^{n_i}, \quad i = 1, \cdots, m.
\end{aligned} \tag{42}$$

A. 2. Dual of Problem (17)

In the following we derive the dual of formulation (17). The primal problem (17) can be put in the following equivalent form:

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{b},\boldsymbol{y}} \quad & \left[\mathbf{0}^T,\nu\right]\boldsymbol{x}+(1-\nu)\boldsymbol{e}^T\boldsymbol{y} \\
\text{s.t.} \quad & \left\|\begin{bmatrix}\sqrt{2}C & 0\\ \mathbf{0}^T & -1\end{bmatrix}\boldsymbol{x}+\begin{bmatrix}\mathbf{0}\\ 1\end{bmatrix}\right\| \le \left[\mathbf{0}^T,1\right]\boldsymbol{x}+1, \\
& \left\|\left[\rho_p^i(H^{ij})^T,\mathbf{0}\right]\boldsymbol{x}\right\| \le \left[A_p^i(H^{ij})^T,\mathbf{0}\right]\boldsymbol{x}+(\boldsymbol{h}_p^{ij})^T\boldsymbol{y}-(\mathbf{r}^{ij})^T\boldsymbol{b}-1, \\
& \qquad\qquad p=1,\cdots,l_i,\ i,j=1,\cdots,K,\ i\neq j, \\
& \boldsymbol{y}\ge\mathbf{0},
\end{aligned}$$

where $\boldsymbol{x}=\left[\boldsymbol{w}^T,t\right]^T$. Then, by (42), we get the dual of problem (17) as follows:

$$\max_{\boldsymbol{\alpha},\boldsymbol{s},\sigma,\tau} \quad \sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}\sum_{p=1}^{l_i}\alpha_p^{ij}-(\sigma+\tau) \tag{43}$$

$$\text{s.t.} \quad \sqrt{2}C^T\boldsymbol{\xi}+\sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}\sum_{p=1}^{l_i}\left(\alpha_p^{ij}H^{ij}(A_p^i)^T+\rho_p^iH^{ij}\boldsymbol{s}_p^{ij}\right)=\mathbf{0}, \tag{44}$$

$$\sigma-\tau=\nu, \tag{45}$$

$$\sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}\sum_{p=1}^{l_i}\alpha_p^{ij}\boldsymbol{h}_p^{ij}+\boldsymbol{\lambda}=(1-\nu)\boldsymbol{e}, \tag{46}$$

$$-\sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}\sum_{p=1}^{l_i}\alpha_p^{ij}\boldsymbol{r}^{ij}=\mathbf{0}, \tag{47}$$

$$\left\|\begin{bmatrix}\boldsymbol{\xi}\\ \tau\end{bmatrix}\right\|\le\sigma, \tag{48}$$

$$\|\boldsymbol{s}_p^{ij}\|\le\alpha_p^{ij}, \qquad p=1,\cdots,l_i,\ i,j=1,\cdots,K,\ i\neq j, \tag{49}$$

$$\boldsymbol{\lambda}\ge\mathbf{0}. \tag{50}$$

By (9), (18) and (22) we get

$$\sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}\sum_{p=1}^{l_i}\alpha_p^{ij}H^{ij}(A_p^i)^T=\bar{A}^T\boldsymbol{\alpha}. \tag{51}$$

By (19) and (23) we get

$$\sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}\sum_{p=1}^{l_i}\rho_p^iH^{ij}\boldsymbol{s}_p^{ij}=\bar{H}^T\boldsymbol{s}. \tag{52}$$

Hence by (51) and (52), we can express (44) compactly as follows:

$$\sqrt{2}C^T\boldsymbol{\xi}+\bar{A}^T\boldsymbol{\alpha}+\bar{H}^T\boldsymbol{s}=\mathbf{0}. \tag{53}$$

By (53) and (28), we get the following equation:

$$\boldsymbol{\xi} = -\frac{1}{\sqrt{2(K+1)}}(\bar{A}^T\boldsymbol{\alpha} + \bar{H}^T\boldsymbol{s}). \tag{54}$$

By (13) and (22), we have

$$\sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}\sum_{p=1}^{l_i}\alpha_p^{ij}\boldsymbol{h}_p^{ij} = \boldsymbol{\alpha}.$$

Hence (46) can be expressed as follows:

$$(1-\nu)\boldsymbol{e} - \boldsymbol{\lambda} - \boldsymbol{\alpha} = \boldsymbol{0}. \tag{55}$$

By (11), (20) and (22), we can rewrite (47) as follows:

$$-\bar{E}^T\boldsymbol{\alpha} = \boldsymbol{0}. \tag{56}$$

Combining (54)–(56), problem (43)–(50) can be written as (21).

Appendix B. Complementarity Conditions of SOCP

Let $\operatorname{bd}\mathcal{K}^n$ denote the boundary of $\mathcal{K}^n$:

$$\operatorname{bd}\mathcal{K}^n = \left\{\begin{bmatrix} z_0 \\ \bar{\boldsymbol{z}} \end{bmatrix} \in \mathcal{K}^n : \ \|\bar{\boldsymbol{z}}\| = z_0\right\}.$$

Let $\operatorname{int}\mathcal{K}^n$ denote the interior of $\mathcal{K}^n$:

$$\operatorname{int}\mathcal{K}^n = \left\{\begin{bmatrix} z_0 \\ \bar{\boldsymbol{z}} \end{bmatrix} \in \mathcal{K}^n : \ \|\bar{\boldsymbol{z}}\| < z_0\right\}.$$

For two elements $\begin{bmatrix} z_0 \\ \bar{\boldsymbol{z}} \end{bmatrix} \in \mathcal{K}^n$ and $\begin{bmatrix} z_0' \\ \bar{\boldsymbol{z}}' \end{bmatrix} \in \mathcal{K}^n$, $\begin{bmatrix} z_0 \\ \bar{\boldsymbol{z}} \end{bmatrix}^T \begin{bmatrix} z_0' \\ \bar{\boldsymbol{z}}' \end{bmatrix} = 0$ if and only if the following conditions are satisfied [15]:

$$\begin{bmatrix} z_0 \\ \bar{\boldsymbol{z}} \end{bmatrix} \in \operatorname{int}\mathcal{K}^n \quad\Rightarrow\quad \|\bar{\boldsymbol{z}}'\| = z_0' = 0, \tag{57}$$

$$\begin{bmatrix} z_0' \\ \bar{\boldsymbol{z}}' \end{bmatrix} \in \operatorname{int}\mathcal{K}^n \quad\Rightarrow\quad \|\bar{\boldsymbol{z}}\| = z_0 = 0, \tag{58}$$

$$\begin{bmatrix} z_0 \\ \bar{\boldsymbol{z}} \end{bmatrix} \in \operatorname{bd}\mathcal{K}^n \setminus \{\boldsymbol{0}\},\ \begin{bmatrix} z_0' \\ \bar{\boldsymbol{z}}' \end{bmatrix} \in \operatorname{bd}\mathcal{K}^n \setminus \{\boldsymbol{0}\} \quad\Rightarrow\quad \begin{bmatrix} z_0 \\ \bar{\boldsymbol{z}} \end{bmatrix} = \mu\begin{bmatrix} z_0' \\ -\bar{\boldsymbol{z}}' \end{bmatrix}, \tag{59}$$

where $\mu > 0$ is a constant. These three conditions are regarded as a generalization of the complementary slackness conditions in linear programming.

References

[1] F. Alizadeh and D. Goldfarb (2003). Second-order cone programming. *Math. Program.*, Ser. B **95**, 3–51.

[2] E. L. Allwein, R. E. Schapire and Y. Singer (2001). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, **1**, 113–141.

[3] C. Angulo, X. Parra and A. Català (2003). K-SVCR. A support vector machine for multi-class classification. *Neurocomputing*, **55**, 57–77.

[4] K. P. Bennett and O.L. Mangasarian (1994). Multicategory discrimination via linear programming. *Optimization Methods and Software.* **3**, 27–39.

[5] E. J. Bredensteiner and K. P. Bennett (1999). Multicategory Classification by support vector machines. *Computational Optimization and Applications*, **12**, 53–79.

[6] C. L. Blake and C. J. Merz (1998). UCI repository of machine learning databases. University of California. [www http://www.ics.uci.edu/∼mlearn/MLRepository.html]

[7] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. Müller, E. Sackinger, P. Simard and V. Vapnik (1994). Comparison of classifier methods: a case study in handwriting digit recognition. in: IAPR (Ed.), *Proceedings of the International Conference on Pattern Recognition*, pp. 77–82. IEEE Computer Society Press.

[8] M. Fukushima, Z.Q. Luo and P. Tseng (2002). Smoothing functions for second-order-cone complementarity problems. *SIAM Journal on Optimization*, **12**, 436–460.

[9] T.G. Dietterich and G. Bakiri (1995). Solving multi-class learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, **2**, 263–286.

[10] D. Goldfarb and G. Iyengar (2003). Robust convex quadratically constrained programs. *Mathematical Programming*, **97**, 495–515.

[11] Y. Guermeur (2002). Combining discriminant models with new multi-class SVMs. *Pattern Analysis and Applications*, **5**, 168–179.

[12] T. J. Hastie and R. J. Tibshirani (1998). Classification by pairwise coupling. in: M. I. Jordan, M. J. Kearns and S. A. Solla (Eds.), *Advances in Neural Information Processing Systems*, **10**, 507–513. MIT Press, Cambridge, MA.

[13] S. Hayashi, N. Yamashita and M. Fukushima (2005). A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM Journal on Optimization*, **15**, 593–615.

[14] U. Kreßel (1999). Pairwise classification and support vector machines. in: B. Schölkopf, C. J. C. Burges and A. J. Smola (Eds.), *Advances in Kernel Methods: Support Vector Learning*, pp. 255–268. MIT Press, Cambridge, MA.

[15] M. S. Lobo, L. Vandenberghe, S. Boyd and H. Lébret (1998). Applications of second-order cone programming. *Linear Algebra and Applications*, **284**, 193–228.

[16] J. Platt (1999). Sequential minimal optimization: A fast algorithm for training support vector machines. in: B. Schölkopf, C. J. C. Burges and A. J. Smola (Eds.), *Advances in Kernel Methods: Support Vector Learning*, pp. 185–208. MIT Press, Cambridge, MA.

[17] J. Platt, N. Cristianini and J. Shawe-Taylor (2000). Large margin DAGs for multiclass classification. in: S. A. Solla, T. K. Leen and K. -R. Müller (Eds.), *Advances in Neural Information Processing Systems*, **12**, 547–553. MIT Press, Cambridge, MA.

[18] B. Schölkopf, A. Smola and K. Müller (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, **10**, 1299–1319.

[19] J. Sturm (2001). Using SeDuMi, a matlab toolbox for optimization over symmetric cones. Department of Ecnometrics, Tilburg University, The Netherlands.

[20] V. Vapnik (1998). *Statistical Learning Theory.* Wiley, New York.

[21] J. Weston and C. Watkins (1998). Multi-class support vector machines. CSD-TR-98-04 Royal Holloway, University of London, Egham, UK.

[22] Y. Yajima (2005). Linear programming approaches for multicategory support vector machines. *European Journal of Operational Research*, **162**, 514–531.

[23] P. Zhong and M. Fukushima (2005). A New multi-class support vector algorithm. *Optimization Methods and Software.* To appear.