





Approximate Multiobjective Multiclass Support Vector Machine Restricting Classifier Candidates Based on k-Means Clustering

Keiji Tatsumi^(✉) , Takahumi Sugimoto, and Yoshifumi Kusunoki 

Graduate School of Engineering, Osaka University,
2-1 Yamada-Oka, Suita, Osaka, Japan
{tatsumi,kusunoki}@eei.eng.osaka-u.ac.jp

Abstract. In this paper, we propose a reduction method for the multiobjective multiclass support vector machine (MMSVM), one of all-together method of the SVM. The method can maintain the discrimination ability, and reduce the computational complexity of the original MMSVM. First, we derive an approximate convex multiobjective optimization problem for the MMSVM by linearizing some constraints, and we secondly restrict the normal vectors of classifier candidates by using centroids obtained from the k-means clustering for each class dataset. The derived problem can be solved by the reference point method based on the centers of gravity of class datasets, in which the geometric margins between all pairs are exactly maximized. Some numerical experiments for benchmark problems show that the proposed method can reduce the computational complexity without decreasing its generalization ability widely.

Keywords: Multiclass classification · Support vector machine · Multiobjective optimization · Reference point method · k-means clustering

1 Introduction

The binary support vector machine (SVM) [12] is one of popular machine learning methods, which finds a classifier with a high classification ability by maximizing the geometric margin between data and a separating hyperplane. In addition, various extended methods of the binary SVM have been investigated for multi-class classification. In this paper, we focus on all-together (AT) method among the extended ones, especially, the multiobjective multiclass SVM (MMSVM) [7]. It was reported that comparing with the simplest AT method maximizing functional margins, the MMSVM can obtain a classifier with a higher classification rate by maximizing exactly the geometric margin between each class pair. However, it requires a larger amount of computational resources than the simplest AT and other methods [7–9].

Therefore, in this paper, we propose a method of reducing the computational complexity without decreasing its generalization ability widely. First, we derive an approximate convex multi-objective optimization problem (MOP) by linearizing some constraints of the original non-convex MOP which is used to find a classifier in MMSVM. Secondly, we restrict the normal vectors of classifier candidates to a space spanned by centroids obtained from the preliminary k-means clustering. Thirdly, we solve the derived MOP by applying the reference point method. Through numerical experiments for benchmark problems, we evaluate performance of classifiers obtained by the proposed method and its computational complexity.

2 Multiclass Classification

The multiclass classification means discriminating data into more than two classes. We assume that a dataset (x^i, y_i) , $i = 1, \dots, l$ are generated by the same distribution $P(x, y)$, where $x^i \in \mathcal{R}^n$ denotes an n -dimensional input, and $y_i \in M := \{1, \dots, m\}$ denotes a label which the corresponding x^i should be classified into. The aim is finding a classifier $f(x)$ which satisfies $y_i = f(x^i)$, $i = 1, \dots, l$ and which can correctly classify a new unknown input x from the same distribution. In this paper, we assume that there exists an appropriate feature space F and a corresponding function $\phi : \mathcal{R}^n \rightarrow F$. Thus, we mainly discuss a linear classification on F which uses the kernel method.

In the representative SVMs for multiclass classification such as one-against-all (OAA) [2] and all-together (AT) methods [11, 12], the following discriminant function is often used:

$$f(x) = \operatorname{argmax}_{p \in M} w^p \top \phi(x) + b^p$$

where w^p , b^p , $p \in M$ denote a weight vector and a bias value, respectively. Thus, the aim is finding appropriate (w^p, b^p) , $p \in M$.

2.1 SVM Maximizing Functional Margins

As the simplest AT method, the SVM maximizing the sum of functional margins was proposed in [11, 12], which can be straightforwardly derived from the binary SVM.

$$\begin{aligned} \text{(AT)} \quad & \min \sum_{p \in M} \sum_{q \in M} \|w^p - w^q\|^2 \\ & \text{s.t. } (w^p - w^q) \top \phi(x^i) + (b^p - b^q) \geq 1, \quad i \in I_p, q > p, \quad p, q \in M, \\ & (w^q - w^p) \top \phi(x^i) + (b^q - b^p) \geq 1, \quad i \in I_q, q > p, \quad p, q \in M, \end{aligned}$$

where $I_p := \{i \in \{1, \dots, l\} \mid y_i = p\}$, $p \in M$. Note that maximizing the functional margin in binary SVM can guarantee exact maximization of the distance between data and a separating hyperplane, called *geometric margin*, which can contribute its high generalization ability. On the other hand, in the problem (AT) for the

multiclass classification, maximizing functional margins $1/\|w^p - w^q\|$ does not necessarily guarantees the maximization of the geometric margins. Namely, the functional margin for a class pair pq does not necessarily represent the distance between the corresponding separating hyperplane:

$$(w^p - w^q)^\top \phi(x) + (b^p - b^q) = 0$$

and the closest data in classes $\{p, q\}$, as pointed out in [7], which is represented by

$$d_{pq}(w, b) = \min_{i \in I_p \cup I_q} \frac{|(w^p - w^q)^\top \phi(x)^i + (b^p - b^q)|}{\|w^p - w^q\|}, \quad q > p, \quad p, q \in M.$$

Thus, it might be difficult to expect the generalization ability similar to the binary SVM. The method of maximizing exactly geometric margins was already proposed in [7]. We introduce it in the next section.

2.2 SVM Maximizing Geometric Margins

In order to maximize exactly the geometric margins, an AT method called MMSVM was already proposed, which was formulated as the following multiobjective optimization problem (MOP) [7]:

$$\begin{aligned} \text{(M)} \quad & \max_{w, b, \sigma} \theta_{12}(w, \sigma), \dots, \theta_{m-1, m}(w, \sigma), \\ & \text{s.t. } (w^p - w^q)^\top \phi(x^i) + (b^p - b^q) \geq \sigma_{pq}, \quad i \in I_p, q > p, \quad p, q \in M, \\ & (w^q - w^p)^\top \phi(x^i) + (b^q - b^p) \geq \sigma_{pq}, \quad i \in I_q, q > p, \quad p, q \in M, \\ & \sigma_{pq} \geq 1, \quad q > p, \quad p, q \in M, \end{aligned}$$

where we define $\theta_{pq}(w, \sigma) = \sigma_{pq}/\|w^p - w^q\|$, $q > p$, $p, q \in M$. Note that (M) has more than two objective functions, and the number of them is that of all combinations of class pairs. In [7], it was shown that at any Pareto optimal solution (w^*, b^*, σ^*) of (M), each of objective functional values $\theta_{pq}(w^*, \sigma^*)$ is equal to the geometric margin $d_{pq}(w^*, b^*)$ of the corresponding class pair [7].

Since in general the optimal solutions of the MOP are often given as a set called *Pareto optimal* solutions, and, in addition, (M) is not convex. The problem is more difficult to solve than the single-objective optimization problem (SOP). However, a method of finding a Pareto optimal solution by solving a convex SOP was introduced, and the kernel method can be easily applied to (M).

Now, let's consider the kernel method for (M). The weight vector w^p of the separating hyperplane is represented as a weighted sum of $\phi(x^i)$ by introducing new decision variables $\alpha_i^p \in R$, $i = 1, \dots, l$, $p \in M$:

$$w^p = \sum_{i=1}^l \alpha_i^p \phi(x^i), \quad p \in M. \tag{1}$$

Then, by defining $K := (\phi(x^1), \dots, \phi(x^l))^\top (\phi(x^1), \dots, \phi(x^l))$, $\alpha^p := (\alpha_1^p, \dots, \alpha_l^p)^\top$, $p \in M$, $\bar{\theta}_{pq}(\alpha, \sigma) := \sigma_{pq} / \sqrt{(\alpha^p - \alpha^q)^\top K (\alpha^p - \alpha^q)}$, $q > p$, $p, q \in M$, $\kappa(x^i) := (k(x^1, x^i), \dots, k(x^l, x^i))^\top$, $i = 1, \dots, l$, (M) can be rewritten as

$$(M2) \quad \max_{\alpha, b, \sigma} \bar{\theta}_{12}(\alpha, \sigma), \dots, \bar{\theta}_{(m-1)m}(\alpha, \sigma)$$

$$\text{s.t. } (\alpha^p - \alpha^q)^\top \kappa(x^i) + (b^p - b^q) \geq \sigma_{pq}, \quad i \in I_p, \quad q > p, \quad p, q \in M,$$

$$(\alpha^q - \alpha^p)^\top \kappa(x^i) + (b^q - b^p) \geq \sigma_{pq}, \quad i \in I_q, \quad q > p, \quad p, q \in M,$$

$$\sigma_{pq} \geq 1, \quad q > p, \quad p, q \in M,$$

In addition, the discriminant function can be represented by $f(x) = \operatorname{argmax}_{p \in M} \{ \alpha^{p* \top} \kappa(x) + b^{p*} \}$, where $(\alpha^*, b^*, \sigma^*)$ is the Pareto optimal solution of (M2).

As a method of solving (M2), the ε -constraint method is used, which is one of popular scalarization methods for the MOP [4]. In the method, the SOP is derived instead of (M2), in which one of objective functions of (M2) is used as the objective function of the new SOP, and other objective functions are changed into its constraints by using an appropriate constant vector ε . In addition, the following transformation was used to solve the SOP [7, 8].

Now, we focus on all positive eigenvalues values $\lambda_1, \dots, \lambda_\tau$ of K , and the corresponding eigenvectors t^1, \dots, t^τ , where $\tau > 0$ denotes the number of the positive eigenvalues. Then, we have that

$$K = [t^1, \dots, t^\tau] \operatorname{diag}\{\lambda_1, \dots, \lambda_\tau\} [t^1, \dots, t^\tau]^\top =: TAT^\top \quad (2)$$

Then, new decision variables z^p are defined as $z^p := A^{\frac{1}{2}} T^\top \alpha^p$, $p \in M$, and the following convex SOP is obtained:

$$(\varepsilon M2) \quad \max_{z, b, \sigma} \frac{c_{rs}}{\|z^r - z^s\|}$$

$$\text{s.t. } \frac{\sigma_{pq}}{\|z^p - z^q\|} \geq \varepsilon_{pq}, \quad q > p, \quad (p, q) \neq (r, s), \quad p, q \in M,$$

$$(z^p - z^q)^\top A^{\frac{1}{2}} \bar{t}^i + (b^p - b^q) \geq \sigma_{pq}, \quad i \in I_p, \quad q > p, \quad p, q \in M,$$

$$(z^q - z^p)^\top A^{\frac{1}{2}} \bar{t}^i + (b^q - b^p) \geq \sigma_{pq}, \quad i \in I_q, \quad q > p, \quad p, q \in M,$$

$$\sigma_{pq} \geq 1, \quad q > p, \quad (p, q) \neq (r, s), \quad p, q \in M,$$

$$\sigma_{rs} = c_{rs},$$

where \bar{t}^i denotes the i -th row vector of T , the constant ε_{pq} , $p, q \in M$ is appropriately selected for the feasibility of $(\varepsilon M2)$, and class pair rs is appropriately selected. Note that the constraint $\sigma_{rs} = c_{rs}$ with a sufficiently large constant c_{rs} is added so that $(\varepsilon M2)$ is convex, and a large c_{rs} guarantees that the optimal solution of $(\varepsilon M2)$ is Pareto optimal [7]. Moreover, $(\varepsilon M2)$ is a second-order cone programming problem (SOCP), which is a convex problem having the second-order and linear constraints, and which can be effectively solved by using some primal-dual interior method [1]. In addition, numerical experiments showed that the geometric margins of separating hyperplanes constructed by the optimal solution of $(\varepsilon M2)$ are larger than those obtained by the functional margin method

(AT), and that classifiers of (M2) obtained by $(\varepsilon M2)$ have better classification ability than (AT) [7–9].

Next, let us evaluate the computational resources required to solve $(\varepsilon M2)$ and (AT). Since in $(\varepsilon M2)$, a constant vector ε is determined by the optimal solution of (AT), $(\varepsilon M2)$ requires solving (AT). In addition, CPU time of solving an SOCP $(\varepsilon M2)$ is considerably larger than that of (AT) because of many decision variables. Moreover, if the element number l of all datasets is large, the diagonalization of $l \times l$ matrix K also requires a large amount of computational resources. Therefore, in [5], an approximation method for (M) was proposed, in which a single-objective SOCP is derived by introducing the sum of objective functions of (M) and linearizing the right-hand side of the first and second constraints of (M). The SOCP is easily solved due to its convexity, and an feasible solution of (M) can be easily obtained from the optimal solution of the SOCP. The numerical experiments showed that the generalization ability of classifiers obtained by approximation method is better than that of (AT).

In this paper, we derive an approximate MOP by using the same approximation technique, and, furthermore, we restrict the normal vectors of classifier candidates to a space spanned by centroids obtained from a preliminary clustering in order to reduce its computational complexity.

3 Approximate MMSVM

In this section, we introduce the following problem by defining $\delta_{pq} := \sigma_{pq}^2$ and putting a constant upper limit $\rho \geq 1$ on δ_{pq} , $q > p$, $p, q \in M$.

$$\begin{aligned}
 \min_{w, b, \delta} & \eta_{12}(w, \delta), \dots, \eta_{(m-1)m}(w, \delta) \\
 \text{(S1)} \quad \text{s.t.} & (w^p - w^q)^\top \phi(x^i) + (b^p - b^q) \geq \sqrt{\delta_{pq}}, \quad i \in I_p, \quad q > p, \quad p, q \in M, \\
 & (w^q - w^p)^\top \phi(x^i) + (b^q - b^p) \geq \sqrt{\delta_{pq}}, \quad i \in I_q, \quad q > p, \quad p, q \in M, \\
 & 1 \leq \delta_{pq} \leq \rho, \quad q > p, \quad p, q \in M.
 \end{aligned}$$

Here, η_{pq} is defined by $\eta_{pq}(w, \delta) = \|w^p - w^q\|^2 / 2\delta_{pq}$, $q > p$, $p, q \in M$. If ρ is sufficiently large, (S1) can be considered to be equivalent to (M). Then, in order to approximate (S1), we replace the right-hand sides of the first and second constraint inequalities with $(\delta_{pq} + \sqrt{\rho}) / (1 + \sqrt{\rho})$ by using a constant ρ in the same way to [5]. Then, we obtain

$$\begin{aligned}
 \min_{w, b, \delta} & \eta_{12}(w, \delta), \dots, \eta_{(m-1)m}(w, \delta) \\
 \text{(S2)} \quad \text{s.t.} & (w^p - w^q)^\top \phi(x^i) + (b^p - b^q) \geq \frac{\delta_{pq} + \sqrt{\rho}}{1 + \sqrt{\rho}}, \quad i \in I_p, \quad q > p, \quad p, q \in M, \\
 & (w^q - w^p)^\top \phi(x^i) + (b^q - b^p) \geq \frac{\delta_{pq} + \sqrt{\rho}}{1 + \sqrt{\rho}}, \quad i \in I_q, \quad q > p, \quad p, q \in M, \\
 & 1 \leq \delta_{pq} \leq \rho, \quad q > p, \quad p, q \in M.
 \end{aligned}$$

Figure 1 shows the relation of $\sqrt{\delta}$ and $(\delta + \sqrt{\rho}) / (1 + \sqrt{\rho})$, which shows that $\sqrt{\delta} \geq \frac{\delta + \sqrt{\rho}}{1 + \sqrt{\rho}}$ for any $\delta \in [1, \rho]$. By making use of the property, for any Pareto

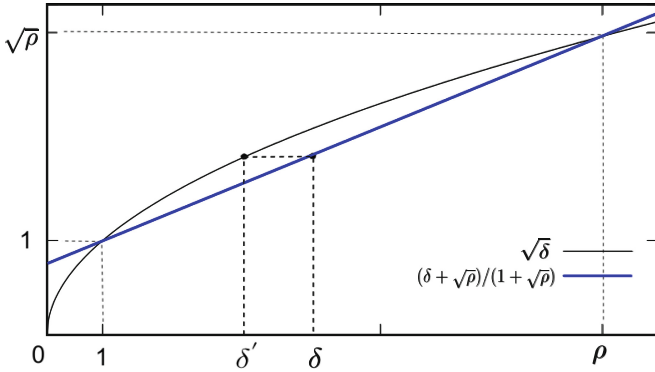


Fig. 1. Approximate affine function

optimal solution (w, b, δ) of (S2), we can obtain a feasible solution (w, b, δ') of (S1) as follows:

$$\delta' = \left(\left(\frac{\delta_{12} + \sqrt{\rho}}{1 + \sqrt{\rho}} \right)^2, \dots, \left(\frac{\delta_{(m-1)m} + \sqrt{\rho}}{1 + \sqrt{\rho}} \right)^2 \right)^\top.$$

Since (S2) can be regarded as convex, it is easier to solve the problem than (M2). In addition, we can show the following properties between solutions of (S1) and (S2): The relation of objective function values at (w, b, δ) and (w, b, δ') is given by

$$\frac{\eta_{pq}(w, b, \delta')}{\eta_{pq}(w, b, \delta)} = \frac{\delta_{pq}}{\delta'_{pq}} \leq \frac{(1 + \sqrt{\rho})^2}{4\sqrt{\rho}}, \quad q > p, \quad p, q \in M.$$

Thus, for the Pareto solution or the feasible solution $(\bar{w}, \bar{b}, \bar{\delta})$ of (S1) which dominate (w, b, δ') such that $\eta_{pq}(\bar{w}, \bar{b}, \bar{\delta}) \leq \eta_{pq}(w, b, \delta')$, $q > p$, $p, q \in M$, we have that

$$\eta_{pq}(w, b, \delta') \leq \frac{(1 + \sqrt{\rho})^2}{4\sqrt{\rho}} \eta_{pq}(\bar{w}, \bar{b}, \bar{\delta}), \quad q > p, \quad p, q \in M.$$

The approximate method for MMSVM proposed in this section is called AMMSVM.

4 AMMSVM Based on K-Means Clustering

Next, we introduce a dimension reduction which restricts the weights of (1) of separating hyperplanes to AMMSVM. This method is based on the assumption that the representation of appropriate weights of the separating hyperplanes

does not need all datasets such as (1), and thus, the weights can be represented by the weighted sum of a smaller number of data. Namely, instead of using all $\phi(x^i), i = 1, \dots, l$, the method selects representative points in F for weights w^p of each $p \in M$. Since by using this restriction, the feasible region of the proposed problem is smaller than the original (M2), solving time and used memories can be expected to be widely reduced. At the same time, the proposed method keeps all the constraints of (M2) which guarantee that all training data is correctly classified. Thus, the method is quite different from a reduction method of deleting training data by some preliminary technique [3]. The proposed method uses the k-means clustering [6] with an appropriate number of clusters to each class data $\phi(x^i), i \in I_p$, and centroids of obtained clusters are used as representative points for the class.

4.1 k-Means Clustering

In the proposed method, the k-means clustering is applied to each dataset $\phi(x^i), i \in I_p$ for class p in order to obtain clusters $\{\phi(x^l)\}_{l \in I_p^k}, k = 1, \dots, c_p$, such that $I_p = \cup_{k=1}^{c_p} I_p^k$, individually, which means minimizing the following function:

$$E^p = \sum_{k=1}^{c_p} \sum_{i \in I_p^k} \|\phi(x^i) - \psi^{p,k}\|^2,$$

where c_p denotes the number of clusters which is appropriately selected for class p . In the numerical experiments at Sect. 5, we set $c_p = \lfloor r|I_p| \rfloor$, and r is a small constant. Centroids of each cluster k are given by $\psi^{p,k} = \sum_{i \in I_p^k} \phi(x^i) / |I_p^k|$. Here, note that the kernel method can be easily applied to the clustering method.

It is well-known that the k-means does not necessarily find the global minimum of E^p . Thus, we executed 20 times k-means clustering and select the centroids of the clustering in which the least E^p was obtained in the numerical experiments.

4.2 Dimension Reduction Based on k-Means Clustering

The centroids obtained by the k-means clustering for the dataset in class p are represented by $\psi^{p,k}, k = 1, \dots, c_p$. By introducing new decision variables $\beta_{q,k}^p, k = 1, \dots, n_p, p, q \in M$, the weight w_p of AMMSVM for class $p \in M$ are given by

$$w^p = \sum_{q \in M} \sum_{h=1}^{c_p} \beta_{q,h}^p \psi^{q,h} = \Psi \beta^p \tag{3}$$

where matrix Ψ and a decision vector β^p is defined as

$$\Psi := (\psi^{1,1}, \psi^{1,2}, \dots, \psi^{1,c_1}, \psi^{2,1}, \dots, \psi^{m,c_m}) \in \mathcal{R}^{c_{all} \times c_{all}},$$

$$\beta^p := (\beta_{1,1}^p, \beta_{1,2}^p, \dots, \beta_{1,c_1}^p, \beta_{2,1}^p, \dots, \beta_{m,c_m}^p)^\top \in \mathcal{R}^{c_{all}},$$

and c_{all} is defined as $\sum_{p \in M} c_p$. Then, the discriminant function is represented by

$$f(x) = \operatorname{argmax}_p \{(\Psi\beta^p)^\top \phi(x) + b^p\},$$

and the decision variables (β^p, b^p) , $q > p$, $p, q \in M$ are determined by solving the following MOP:

(KMS)

$$\begin{aligned} \min_{\beta, b, \delta} & \frac{1}{2} \frac{\|\Psi\beta^1 - \Psi\beta^2\|^2}{\delta_{12}}, \dots, \frac{1}{2} \frac{\|\Psi\beta^{m-1} - \Psi\beta^m\|^2}{\delta_{(m-1)m}} \\ \text{s.t.} & (\Psi\beta^p - \Psi\beta^q)^\top \phi(x^i) + (b^p - b^q) \geq \frac{\delta_{pq} + \sqrt{\rho}}{1 + \sqrt{\rho}}, \quad i \in I_p, \quad q > p, \quad p, q \in M, \\ & (\Psi\beta^q - \Psi\beta^p)^\top \phi(x^i) + (b^q - b^p) \geq \frac{\delta_{pq} + \sqrt{\rho}}{1 + \sqrt{\rho}}, \quad i \in I_q, \quad q > p, \quad p, q \in M, \\ & 1 \leq \delta_{pq} \leq \rho, \quad q > p, \quad p, q \in M. \end{aligned}$$

The geometric margins between all class pairs are maximized by solving (KMS) under the restriction (3). Moreover, since a centroid of each cluster is represented by the weighted sum of $\phi(x^i)$, the kernel method can be applied to (KMS).

Now, similarly to (2), we have that $\Psi^\top \Psi = \hat{T} \hat{\Lambda} \hat{T}^\top$, where $\hat{\Lambda} \in \mathcal{R}^{\tau_c \times \tau_c}$ is a diagonal matrix whose diagonal components are all positive eigenvalues of $\Psi^\top \Psi$, $T \in \mathcal{R}^{c_{all} \times \tau_c}$ consists of the corresponding eigenvectors, and τ_c denotes the number of the positive eigenvalues. Then, by introducing decision variables:

$$z^p = \hat{\Lambda}^{\frac{1}{2}} \hat{T}^\top \beta^p, \tag{4}$$

and defining as $\bar{k}^p(x) := \left(\sum_{j \in I_p^1} k(x^j, x) / |I_p^1|, \dots, \sum_{j \in I_p^{c_p}} k(x^j, x) / |I_p^{c_p}| \right)^\top$ and $\bar{\kappa}(x) := (\bar{k}^1(x), \dots, \bar{k}^m(x))^\top$, we can transform (KMS) to the following MOP:

(KMS2)

$$\begin{aligned} \min_{z, b, \delta} & \frac{1}{2} \frac{\|z^1 - z^2\|^2}{\delta_{12}}, \dots, \frac{1}{2} \frac{\|z^{m-1} - z^m\|^2}{\delta_{(m-1)m}} \\ \text{s.t.} & (z^p - z^q)^\top \hat{\Lambda}^{-\frac{1}{2}} \hat{T}^\top \bar{\kappa}(x^i) + (b^p - b^q) \geq \frac{\delta_{pq} + \sqrt{\rho}}{1 + \sqrt{\rho}}, \quad i \in I_p, \quad q > p, \quad p, q \in M, \\ & (z^q - z^p)^\top \hat{\Lambda}^{-\frac{1}{2}} \hat{T}^\top \bar{\kappa}(x^i) + (b^q - b^p) \geq \frac{\delta_{pq} + \sqrt{\rho}}{1 + \sqrt{\rho}}, \quad i \in I_q, \quad q > p, \quad p, q \in M, \\ & 1 \leq \delta_{pq} \leq \rho, \quad q > p, \quad p, q \in M. \end{aligned}$$

We can easily show that any Pareto optimal solution of (KMS) is obtained by solving (KMS2) as follows:

Theorem 1. *For a Pareto optimal solution (z^*, b^*, δ^*) of (KMS2), a solution (β^*, b^*, δ^*) of which β^* is defined as $\beta^{p*} = \hat{T} \hat{\Lambda}^{-\frac{1}{2}} z^{p*}$, $p \in M$ is Pareto optimal for (KMS). Conversely, for a Pareto optimal solution (β^*, b^*, δ^*) of (KMS), (z^*, b^*, δ^*) in which z^* is defined by (4) is Pareto optimal for (KMS2).*

4.3 Solving Based on Reference Point Method

In this subsection, we apply the reference point method to solve (KMS2), which finds a Pareto optimal solution by minimizing the distance between a given reference point and Pareto optimal solutions in the objective space. The following SOP can be derived:

(KMS3)

$$\begin{aligned} & \min_{z, b, r, \delta} \max_{p, q \in M} \{ \omega_{pq} (r_{pq} - r_{pq}^*) \} + \mu \sum_{p, q \in M} \omega_{pq} r_{pq} \\ & \text{s.t. } 2r_{pq} \delta_{pq} \geq \|z^p - z^q\|^2, \quad r_{pq} \geq 0, \quad q > p, \quad p, q \in M, \\ & (z^p - z^q)^\top \hat{\Lambda}^{-\frac{1}{2}} \hat{T}^\top \bar{\kappa}(x^i) + (b^p - b^q) \geq \frac{\delta_{pq} + \sqrt{\rho}}{1 + \sqrt{\rho}}, \quad i \in I_p, \quad q > p, \quad p, q \in M, \\ & (z^q - z^p)^\top \hat{\Lambda}^{-\frac{1}{2}} \hat{T}^\top \bar{\kappa}(x^i) + (b^q - b^p) \geq \frac{\delta_{pq} + \sqrt{\rho}}{1 + \sqrt{\rho}}, \quad i \in I_q, \quad q > p, \quad p, q \in M, \\ & 1 \leq \delta_{pq} \leq \rho, \quad q > p, \quad p, q \in M. \end{aligned}$$

Here, r_{pq}^* is a given reference point which is used as the criterion of minimizing. In (KMS3), the distance between the reference point and Pareto optimal solutions is measured by the augmented Tchebyshev function, in which ω is a weight vector which determines a balance between objective functions, and μ shows the rate of the second term to the first one. The weight vector ω and reference point r^* are selected as the following three kinds of pairs:

$$\begin{aligned} \text{(R0)} \quad & \omega_{pq} = 1, \quad r_{pq}^* = 0, \\ \text{(R1)} \quad & \omega_{pq} = \|g^p - g^q\|^2, \quad r_{pq}^* = 0, \\ \text{(R2)} \quad & \omega_{pq} = \|g^p - g^q\|^2, \quad r_{pq}^* = 1/\|g^p - g^q\|^2, \end{aligned}$$

where g^p denotes the center of gravity of the data set of class p , namely, $g^p = (1/|I_p|) \sum_{i \in I_p} \phi(x^i)$. The selection is based on the idea that the appropriate balance of margins is roughly estimated at the balance of the distances between the centers of gravity. Then, the discriminant function is represented by

$$f(x) = \operatorname{argmax}_p \left\{ z^{*p\top} \hat{\Lambda}^{-\frac{1}{2}} \hat{T}^\top (k(x^1, x^i), \dots, k(x^l, x^i))^\top + b^{*p} \right\}. \quad (5)$$

where (z^*, b^*, δ^*) denotes the optimal solution of (KMS3). Although (KMS3) is not smooth, an equivalent smooth SOCP can be easily derived, which means

that a Pareto optimal solution of (KMS3) can be easily obtained by solving the SOCP. Moreover, in the numerical experiments, we solve the dual problem of the smooth SOCP problem because its computational time can be expected to be less than that of the original problem.

4.4 Comparison of Computational Complexities

Now, let us compare computational complexities of (ε M2) and (KMS3). The constant vector ε used in (ε M2), are determined by solutions of (AT), which is a large-scale quadratic optimization problem using all class datasets at once. On the other hand, the calculation of centroids for (KMS3) requires a considerably small amount of computational resources even if it is executed 20 times because the k-means clustering is individually applied to each class dataset.

Next, let us compare the diagonalization of kernel matrices used in (ε M2) and the dual problem of (KMS3), and the numbers of the decision variables and constraints of them. The size of matrices diagonalized for (ε M2) and (KMS3) are $l \times l$ and $c_{all} \times c_{all}$, respectively. The numbers of decision variables of (ε M2) and the dual problem of (KMS3) are $m(m+2\tau+1)/2-1$ and $m(m+\tau_c)$, respectively, and the sizes of constraints of them are $(m-1)(l+m/2)-1$ and $(m-1)(l+3m/2)$, respectively. Since in general, we have that $m \leq \tau_c \leq c_{all} \ll \tau \leq l$, more reduction can be expected if c_{all} is small.

5 Numerical Experiments

We applied the existing methods, AT and MMSVM and the variations of the proposed methods to seven benchmark problems [10], and compared the mean correct classification rate and mean CPU time by using the 10-fold cross-validation, in which hyperparameters were appropriately selected. To solve optimization problems, we used software package MOSEK. As variations of the proposed methods, we used the AMMSVM which does not use the dimension reduction based on the k-means clustering and which is solved by the reference point method, and KMSs in which r was varied in $\{0.05, 0.1, 0.15, 0.2\}$, which are represented by AMM, KMS5, KMS10, KMS15 and KMS20, respectively, and three kinds of reference point and weights, R0, R1 and R2, were used for AMMSVM and each KMS. We used the RBF kernel, namely, $k(x, y) = \exp(-\gamma\|x - y\|^2)$.

The results are shown in Tables 1 and 2. In Table 1, the numbers in parentheses denote the best hyperparameters of each method: (γ) in AT and MMSVM, and (γ, ρ, μ) in AMMSVM and KMSs. The italic and bold number denote the first and second best classification rate for each problem.

Table 1 shows that MMSVM obtained a high classification rate for many problems, while the classification rates of AMMSVM and KMSs with a large r are equal or slightly smaller than those of MMSVM. The classification rates of KMSs mostly increase as r increases. In particular, although the rates are considerably low if r is small for Ecoli, the highest rate was obtained by the KMS with $r = 0.15$. In addition, AMMSVM and KMSs achieved a higher rate than MMSVM

Table 1. Mean correct classification rate of four methods for benchmark problems

	Wine	Balance	DNA	Car	Dermatology	Zoo	Ecoli
AT	98.89 (1)	90.72 (5)	96.00 (0.05)	99.31 (5)	97.26 (0.5)	97.09 (0.5)	80.95 (50)
MMSVM	98.89 (1)	97.92 (1)	95.95 (0.05)	99.48 (5)	98.09 (5.00e-04)	97.09 (1.00e-04)	80.96 (50)
AMM R0	98.89 (1,10,10)	100.00 (0.05,10,1e-5)	95.80 (0.05,1,10)	99.42 (5,10,1e-5)	97.81 (1e-4,100,1e-5)	97.09 (0.5,1,1e-5)	81.26 (50,1,1e-5)
R1	98.89 (5,1,1e-5)	100.00 (0.05,100,1e-5)	95.80 (0.05,1,100)	99.36 (1,100,1e-5)	97.81 (1e-4,10,100)	97.09 (1e-4,100,1e-5)	81.26 (50,100,1e-5)
R2	98.89 (5,1,1e-5)	100.00 (0.05,100,1e-5)	95.80 (0.05,1,100)	99.36 (1,100,1e-5)	97.81 (1e-4,10,100)	97.09 (1e-4,100,1e-5)	81.26 (50,100,1e-5)
KMS5 R0	98.33 (1,100,1e-5)	100.00 (1e-4,1e+3,1e-5)	92.90 (0.05,10,1e-5)	98.15 (0.5,10,100)	98.08 (0.01,100,1e-5)	98.09 (0.05,100,1e-5)	43.8 (500,1,1e-5)
R1	98.33 (10,10,10)	100.00 (1e-4,10,1e-5)	93.40 (0.05,1e+6,1e-5)	98.32 (1,1,100)	98.36 (0.01,1e+3,10)	98.09 (1e-4,100,10)	43.80 (500,1,1e-5)
R2	98.89 (10,10,1)	100.00 (1e-4,10,1e-5)	92.95 (0.05,1,10)	98.21 (1,1,10)	98.36 (0.005,100,0.1)	98.09 (1e-4,100,10)	43.80 (500,1,1e-5)
KMS10 R0	98.33 (5,10,1e-5)	100.00 (0.05,10,1e-5)	94.35 (0.05,1,10)	99.31 (1,1,1e-5)	98.09 sparabreak (0.005,10,10)	98.09 (0.05,100,1e-5)	80.66 (1,1e+6,1e-4)
R1	98.33 (10,1,1e-5)	100.00 (0.05,10,100)	94.30 (0.05,10,1e-5)	99.02 (1,10,1)	98.09 (0.1,1e+5,1e-5)	98.09 (1e-3,1e+6,1e-5)	80.66 (0.1,1e+6,1e-5)
R2	98.89 (10,1,1)	100.00 (0.05,100,1e-5)	93.75 (0.05,1,1)	99.19 (1,10,100)	98.36 (0.1,1e+4,1e-5)	98.09 (1e-4,1e+5,1e-4)	78.58 (5,1e+5,1e-4)
KMS15 R0	98.33 (0.5,1e+6,1)	100.00 (0.05,10,1e-5)	94.70 (0.05,1,10)	99.31 (1,100,1)	97.81 (1e-4,100,1e-5)	98.09 (0.1,100,1e-5)	82.75 (10,100,1e-5)
R1	99.44 (5,1,1e-5)	100.00 (0.05,100,1e-5)	94.55 (0.05,1,100)	99.25 (1,100,1e-5)	97.82 (0.1,100,10)	98.09 (0.05,100,1e-5)	81.60 (5,1e+3,1e-5)
R2	98.33 (5,1,1e-5)	100.00 (0.05,100,1e-5)	94.85 (0.05,1,1)	99.25 (1,100,1e-5)	98.09 (1,100,1e-5)	98.09 (0.1,100,1e-5)	79.50 (1,100,1e-5)
KMS20 R0	98.33 (1,1,10)	100.00 (0.05,10,1e-5)	95.00 (0.05,10,1)	99.25 (1,100,100)	97.82 (0.1,100,1e-5)	98.09 (0.1,100,1e-5)	81.86 (5,1e+4,1e-5)
R1	98.33 (1,10,10)	100.00 (0.05,100,1e-5)	94.75 (0.05,10,1e-5)	99.31 (1,10,10)	98.63 (0.1,10,100)	98.09 (0.1,100,1e-5)	81.56 (10,10,1e-5)
R2	98.89 (1,10,0.1)	100.00 (0.05,100,1e-5)	94.75 (0.05,1e+5,1e-5)	99.25 (1,10,1)	97.82 (0.1,10,1)	98.09 (0.1,100,1e-5)	80.42 (5,100,1e-5)

for some problems. The superiority of AMMSVM and KMSs is considered to be caused by the diversity of obtained solutions: The reference point method used in AMMSVM and KMSs finds a solution under various kinds of balances of objective functions, while in MMSVM, a single margin is maximized by the ε -constraint method using the optimal solution of (AT). From Table 2, we can see that AMMSVM reduced CPU time than AT or MMSVM without a dimension reduction, and KMSs did more greatly for large-scale problems even if $r = 0.15$ or 0.20 . Comparing performance of KMSs using three kinds of reference point and weights, namely, KMSs with R0, R1 and R2, each method obtained a high rate for different problems, though no significant difference was observed.

Table 2. Mean CPU time (sec) of four methods for benchmark problems

	Wine	Balance	DNA	Car	Dermatology	Zoo	Ecoli
AT	0.313	4.567	104.677	258.805	12.566	1.103	21.233
MMSVM	0.802	15.727	854.189	1019.577	5.414	3.206	34.363
AMM R0	0.253	1.484	103.038	184.486	1.934	0.248	11.956
R1	0.441	1.478	103.688	162.613	1.930	0.295	11.625
R2	0.247	1.491	79.488	120.569	1.552	0.317	11.261
KMS5 R0	0.116	0.559	6.392	4.009	0.292	0.356	0.748
R1	0.169	0.614	6.144	3.942	0.645	0.370	0.409
R2	0.105	0.581	4.145	6.344	0.645	0.141	0.458
KMS10 R0	0.291	0.858	8.363	8.644	0.542	0.445	1.055
R1	0.294	0.878	12.692	8.431	0.950	0.123	0.830
R1	0.109	0.853	12.656	12.136	0.759	0.156	0.513
KMS15 R0	0.375	1.361	19.758	25.919	0.550	0.408	0.961
R1	0.373	1.242	15.536	26.998	1.280	0.130	0.923
R2	0.131	1.308	21.891	28.375	1.300	0.494	1.044
KMS20 R0	0.423	1.361	38.100	46.297	0.917	0.481	1.331
R1	0.455	1.828	38.447	48.706	1.538	0.195	1.314
R2	0.120	3.120	37.603	50.661	1.591	0.503	1.889

6 Conclusion

In this paper, we have proposed an approximate method of MMSVM which approximates its non-convex MOP by linearizing the constraints, and a reduction method which restricts the normal vectors of separating hyperplanes by using centroids from the k-means clustering. Through numerical experiments, we have observed that the proposed methods are effective to reduce the computational resources without decreasing the classification ability widely.

References

1. Alizadeh, F., Goldfarb, D.: Second-order cone programming. *Math. Program. Ser. B* **95**, 3–51 (2003)
2. Bottou, L., et al.: Comparison of classifier methods: a case study in handwriting digit recognition. In: *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, pp. 77–87 (1994)
3. Boyang, L., Qiangwei, W., Jinglu, H.: A fast SVM training method for very large datasets. In: *Proceedings of the 2009 International Joint Conference on Neural Networks*, pp. 14–19 (2009)
4. Ehrgott, M.: *Multicriteria Optimization*. Springer, Heidelberg (2005). <https://doi.org/10.1007/3-540-27659-9>
5. Kusunoki, Y., Tatsumi, K.: A multi-class support vector machine based on geometric margin maximization. In: Huynh, V.-N., Inuiguchi, M., Tran, D.H., Denoeux, T. (eds.) *IUKM 2018. LNCS (LNAI)*, vol. 10758, pp. 101–113. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75429-1_9
6. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: *Fifth Berkeley Symposium on Mathematics, Statistics and Probability*, pp. 281–297 (1967)
7. Tatsumi, K., Tanino, T., Hayashida, K.: Multiobjective multiclass support vector machines maximizing geometric margins. *Pac. J. Optim.* **6**(1), 115–140 (2010)
8. Tatsumi, K., Kawachi, R., Tanino, T.: Nonlinear extension of multiobjective multiclass support vector machine. In: *Proceedings of the IEEE SMC*, pp. 1338–1343 (2010)
9. Tatsumi, K., Tanino, T.: Support vector machines maximizing geometric margins for multi-class classification. *Off. J. Span. Soc. Stat. Oper. Res.* **22**(3), 815–840 (2014)
10. Lichman, M.: *UCI machine learning repository* (2013). <http://archive.ics.uci.edu/ml>
11. Weston, J., Watkins, C.: Multi-class Support Vector Machines. In: Verleysen, M. (ed.) *ESANN99*, Belgium, Brussels (1999)
12. Vapnik, V.N.: *Statistical Learning Theory*. Wiley, New York (1998)