A twin hyper-ellipsoidal model with a single quadratic constraint for multiclass classification

Ryusei Nagai*, Yuya Yamakawa†and Nobuo Yamashita‡

June 14, 2024

Abstract. Various types of support vector machines (SVMs) have been proposed for multiclass classification tasks. Recently, hypersphere and hyperellipsoid classifiers have attracted much attention since they outperform the existing hyperplane classifiers. The twin hypersphere multi-classification support vector machine (THKSVM) is an effective SVM-based method that exploits multiple hyperspheres to classify data effectively. The THKSVM outperforms the existing methods based on the SVM in terms of computational time, even though its prediction accuracy is competitive. In this paper, we modify the THKSVM to achieve better classifiers within a reasonable timeframe. Firstly, we adopt hyperellipsoids instead of hyperspheres for classification purposes. Secondly, we replace the multiple constraints in the THKSVM with a single, specific constraint. Consequently, our proposed optimization model comprises a single nonconvex quadratic constraint and a convex quadratic objective function. Although the problem remains nonconvex optimization, the uniqueness of the constraint allows us to analytically obtain its global optimum. Therefore, we anticipate that our proposed method will surpass the THKSVM in terms of computational efficiency and prediction accuracy. Finally, we conducted several numerical experiments to show the effectiveness of our proposed method.

Keywords. Multiclass classification, Support vector machine, Twin Hypersphere SVM, Twin Hyper-ellosoidal SVM

1 Introduction

The fundamental techniques in machine learning, including data classification and regression, find applications in various fields such as text categorization, face recognition, and bioinformatics. In particular, the support

*Graduate School of Informatics, Kyoto University, Yoshidahommachi, Sakyo-ku, Kyoto-shi, Kyoto 606-8501, Japan, E-mail: ryusei.nagai@amp.i.kyoto-u.ac.jp

†Graduate School of Informatics, Kyoto University, Yoshidahommachi, Sakyo-ku, Kyoto-shi, Kyoto 606-8501, Japan, E-mail: yuya@i.kyoto-u.ac.jp

‡Graduate School of Informatics, Kyoto University, Yoshidahommachi, Sakyo-ku, Kyoto-shi, Kyoto 606-8501, Japan, Email: nobuo@kyoto-u.ac.jp

vector machine (SVM) [1] is a core method in machine learning that classifies datasets using a hyperplane. Although various methods based on the SVM, including the twin SVM (TSVM), twin hypersphere SVM (THSVM), and twin hyperellipsoidal SVM (TESVM), have primarily focused on binary classification tasks, recent advancements have extended some of these approaches to accommodate multiclass classification.

Concerning multiclass classification, K-SVCR [8] and Twin-KSVC [10] are the basic methods, which are based on the SVM. The K-SVCR is a classification algorithm with ternary output based on Vapnik's support vector theory [1] and can be regarded as an extension of the SVM for multiclass classification, employing a 1-versus-1 structure. The learning phase of the K-SVCR requires solving QPPs K(K-1)/2 times, making it unsuitable for large-scale datasets. The Twin-KSVC is an improved classification algorithm that combines the structural advantage of the K-SVCR, which employs a "1-versus-1-versus-rest" structure, with the high computational speed of the TWSVM. Like the K-SVCR, it also utilizes a "1-versus-1-versus-rest" structure and classifies a dataset using two nonparallel hyperplanes. Since these hyperplanes are constructed by solving smaller QPPs compared to the K-SVCR, the Twin-KSVC generally outperforms the K-SVCR in terms of computational speed.

The THKSVM is an extension of the THSVM for multiclass classification and separates a dataset by exploiting K hyper-spheres, where K represents the number of classes in the dataset. Similar to the THSVM, the THKSVM maintains the same level of prediction accuracy. However, constructing these hyperspheres involves solving nonconvex optimization problems, with no guarantee of their global optimal solutions, leading to a higher computational cost than K-SVCR and Twin-KSVC, as K hyperspheres need to be constructed. Moreover, since the THKSVM utilizes hyperspheres for prediction, there is a possibility that its prediction accuracy could be improved by replacing hyperspheres with hyperellipsoids.

In this paper, we propose a Twin Hyper-Ellipsoidal Multiclass Classification Support Vector Machine (TEKSVM). Although the proposed TEKSVM is derived from the THKSVM for multiclass classification, it differs in two key aspects. The first difference lies in the fact that the TEKSVM utilizes hyperellipsoids for dataset classification, instead of hyperspheres. Therefore, we anticipate that the proposed method will achieve higher prediction accuracy compared to the THKSVM, like the improvement seen with the TESVM for binary classification due to the utilization of hyperellipsoids. Regarding the second difference, we consider replacing the nonconvex optimization problems used in the THKSVM with quadratic programming problems over one inequality quadratic constraint (QP1QC). Thanks to this change, the global optima of the QP1QC can be analytically computed, whereas obtaining global optima for the nonconvex optimization problem in the THKSVM, as stated above is difficult. Moreover, the proposed method

can be regarded as new rather than a simple extension of the THKSVM. We summarize the contribution of this paper as follows.

(a) To propose a new method for multiclass classification.

(b) To derive an effective computational method for solving the QP1QC.

(c) To confirm the validity of the proposed method.

This paper is organized as follows. In Section 2, we introduce some notation and basic results regarding a quadratic programming problem with one inequality quadratic constraint. Furthermore, we review existing SVM based multiclass classification methods. Section 3 presents the TEKSVM and discusses its properties. In Section 4, we conduct numerical experiments to show the effectiveness of the proposed TEKSVM. Finally, Section 6 provides conclusions and remarks.

2 Preliminaries

In this section, we introduce some notation and existing results for subsequent discussions. Then we review three existing SVM-based models. We denote the Euclidean norm of $x \in \mathbb{R}^n$ by $\|x\| := \sqrt{x^\top x}$. If a symmetric matrix $A \in \mathbb{R}^{n\times n}$ is positive semidefinite, we denote $A \succeq O$. For a matrix $A \in \mathbb{R}^{m\times n}$, we denote the range of A as $\mathcal{R}(A)$.

2.1 Quadratic programming over one inequality quadratic constraint

Now we introduce quadratic programming over one inequality quadratic constraint (QP1QC):

$$\begin{array}{ll} \underset{x}{\text{minimize}} & x^\top P x - 2f^\top x \\ \text{subject to} & x^\top Q x - 2g^\top x \leq \mu, \end{array} \tag{QP1QC}$$

where P, Q are $n\times n$ real symmetric matrices, f, g are n-dimensional vectors, and μ is a real number.

In general, finding the global optimum of quadratic programming problems with quadratic constraints is challenging due to their non-convexity. However, in the case of (QP1QC), it is known that its global optimum can be obtained under certain special assumptions. To explain this fact, we first define the primal Slater condition for (QP1QC).

Definition 1. *We say the primal Slater condition is satisfied if there exists* x_0 *such that*

$$x_0^\top Q x_0 - 2g^\top x_0 < \mu.$$

Using the aforementioned primal Slater condition, Hsia et al. [7] showed the following two theorems.

Theorem 1. [7, Theorem 4] *Under the primal Slater condition,* (QP1QC) *is unbounded below if and only if the system of* σ*:*

$$\begin{cases} P + \sigma Q \succeq O, \\ \sigma \geq 0, \\ f + \sigma g \in \mathcal{R}(P + \sigma Q), \end{cases} \tag{1}$$

has no solution.

Theorem 2. [7, Theorem 6] *Under the primal Slater condition, if the value of* (QP1QC) *is finite and* $I_{\succeq}(P, Q)$ *is an interval, the infimum of* (QP1QC) *is always attainable. Here,* $I_{\succeq}(P, Q)$ *is defined as* $I_{\succeq}(P, Q) \coloneqq \{\sigma \in \mathbb{R} \mid P + \sigma Q \succeq O\}$*, where* P *and* Q *are symmetric matrices.*

We review some existing multi-classification methods based on the SVM. Let T be a given dataset defined as

$$T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$$

where $x_i \in \mathbb{R}^n$ and $y_i \in \{1, 2, \ldots, K\}$ for $i = 1, 2, \ldots, l$. From now on, φ represents an arbitrary mapping to a feature space, and we will consider classification in this feature space.

2.2 K-SVCR

The K-SVCR (Support Vector Classification Regression for K-class classification) [8] selects two specified classes and trains a classifier using these two classes and all others. For convenience, let us denote the two focused classes as s and t represented them as

$$\begin{aligned} I^+ &\coloneqq \{i \mid y_i = s\}, \\ I^- &\coloneqq \{i \mid y_i = t\}, \\ I^r &\coloneqq \{i \mid y_i \neq s, t\}. \end{aligned}$$

Moreover, we define $\tilde{y}_i$ as follows:

$$\tilde{y}_i = \begin{cases} 1 & \text{if } i \in I^+, \\ -1 & \text{if } i \in I^-, \\ 0 & \text{if } i \in I^r. \end{cases}$$

As illustrated in Figure 1, for classes s and t, the K-SVCR seeks hyperplanes that separate classes I^+ and I^- such that the points in I^r lie within the ϵ-band of the hyperplanes. In Figure 1, I^+ and I^- respectively denote the set of indices i such that $y_i = 1$ and $y_i = 2$, while I^r denotes the set of

indices i such that $y_i = 3, 4$. To find such hyperplanes, the K-SVCR solves the following optimization problem.

$$\begin{aligned} \underset{w,b,\xi,\eta,\zeta}{\text{minimize}} \quad & \frac{1}{2}\|w\|^2 + c_1 \sum_{i \in I^+ \cup I^-} \xi_i + c_2 \sum_{i \in I^r} (\eta_i + \eta_i^*) \\ \text{subject to} \quad & \tilde{y}_i (w^\top \varphi(x_i) + b) \geq 1 - \xi_i, \ i \in I^+ \cup I^-, \\ & -\epsilon - \zeta_i \leq w^\top \varphi(x_i) + b \leq \epsilon + \eta_i, \ i \in I^r, \\ & \xi_i \leq 0, \ i \in I^+ \cup I^-, \\ & \eta_i, \zeta_i \leq 0, \ i \in I^r, \end{aligned} \tag{2}$$

where ξ_i, η_i, ζ are slack variables, c_1 and c_2 are parameters, and ϵ is restricted to be smaller than 1 to avoid overlapping. Moreover, the corresponding decision function is defined as

$$f(x) = \begin{cases} 1 & \text{if } (w^*)^\top \varphi(x) + b^* \geq \epsilon, \\ -1 & \text{if } (w^*)^\top \varphi(x) + b^* \leq -\epsilon, \\ 0 & \text{otherwise,} \end{cases}$$

where w^* and b^* are the optimal solution of problem (2). For each decision function, if $w^\top \varphi(x) + b \geq \epsilon$ is satisfied, the first class receives one vote, and the others receive none. However, if $w^\top \varphi(x) + b \leq -\epsilon$ holds, one vote is assigned to the second class and none to the others. If neither of these inequalities are satisfied, the votes for the first and second classes are reduced by one, while the votes for the others remain the same. After tallying all the votes, the testing point x is assigned to the class with the highest number of votes.

Note that the K-SVCR solves problem (2) once for any two classes s and t, i.e., it deals with the optimization problem $K(K-1)/2$ times. Moreover, since optimization problem (2) is solved for the two focused classes and all others, the K-SVCR is said to have a "1-versus1-versus-rest" structure.

2.3 Twin-KSVC

The Twin-KSVC [10] also features a "1-versus-1-versus-rest" structure but employs two nonparallel hyperplanes for the two focused classes, unlike the K-SVCR. To construct the two hyperplanes $x^\top w_1 + b_1 = 0$ and $x^\top w_2 + b_2 = 0$, the Twin-KSVC uses (w_1, b_1) and (w_2, b_2) that respectively solve problems

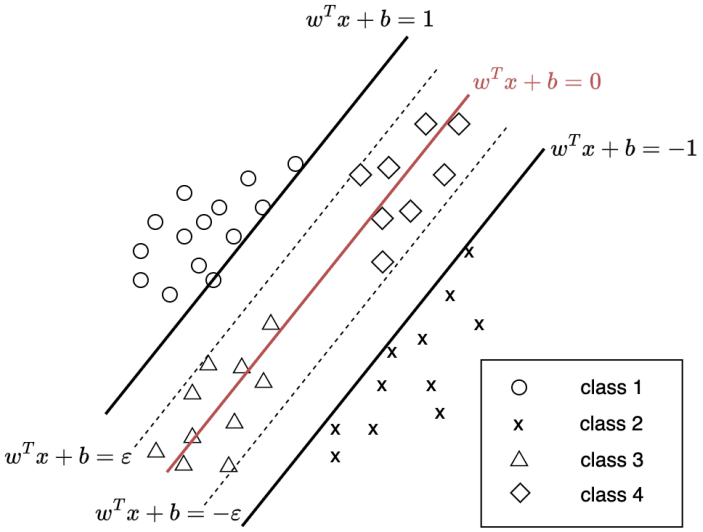

Figure 1: The hyperplanes of the K-SVCR for the two focused classes 1 and 2.

(3) and (4) defined as follows:

$$\begin{aligned} \underset{w,b,\xi,\eta}{\text{minimize}} \quad & \frac{1}{2}\sum_{i\in I^+}(\varphi(x_i)^\top w + b)^2 + c_1\sum_{i\in I^-}\xi_i + c_2\sum_{i\in I^r}\eta_i \\ \text{subject to} \quad & -(\varphi(x_i)^\top w + b) + \xi_i \geq 1,\ i \in I^-, \\ & -(\varphi(x_i)^\top w + b) + \eta_i \geq 1-\epsilon,\ i \in I^r, \\ & \xi_i \geq 0,\ i \in I^-, \\ & \eta_i \geq 0,\ i \in I^r, \end{aligned} \tag{3}$$

and

$$\begin{aligned} \underset{w,b,\xi,\eta}{\text{minimize}} \quad & \frac{1}{2}\sum_{i\in I^-}(\varphi(x_i)^\top w + b)^2 + c_1\sum_{i\in I^+}\xi_i + c_2\sum_{i\in I^r}\eta_i \\ \text{subject to} \quad & (\varphi(x_i)^\top w + b) + \xi_i \geq 1,\ i \in I^+, \\ & (\varphi(x_i)^\top w + b) + \eta_i \geq 1-\epsilon,\ i \in I^r, \\ & \xi_i \geq 0,\ i \in I^+, \\ & \eta_i \geq 0,\ i \in I^r, \end{aligned} \tag{4}$$

respectively, where ξ and η are slack variables. Figure 2 shows an example of the two nonparallel hyperplanes of the Twin-KSVC, where the two focused classes are 1 and 2.

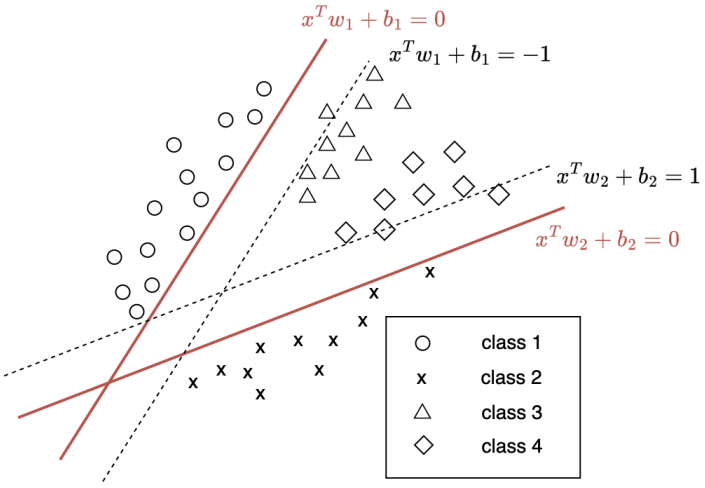

Figure 2: Two nonparallel hyperplanes of the Twin-KSCV for two classes 1 and 2.

The corresponding decision function is defined as

$$f(x) = \begin{cases} 1 & \text{if } \varphi(x)^\top w_1 + b_1 \geq -1 + \epsilon, \\ -1 & \text{if } \varphi(x)^\top w_2 + b_2 \leq 1 - \epsilon, \\ 0 & \text{otherwise.} \end{cases}$$

Similar to the K-SVCR, the final class label for a testing point x is decided by a voting rule. In total, the Twin-KSVC solves problems (3) and (4) $K(K-1)/2$ times and generally operates faster than the K-SVCR. However, its computational speed is not fast enough for large-scale datasets. One reason for this is that the Twin-KSVC involves the inverse of matrices when solving dual problems.

2.4 Twin hypersphere multiclass classification support vector machine

The THKSVM approach utilizes K hyperspheres to classify a dataset. For each $k \in \{1, 2, \ldots, K\}$, the center c_k and squared radius R_k of the k-th hypersphere are obtained by solving the following problem:

$$\begin{aligned} \underset{c,R,\xi}{\text{minimize}} \quad & \frac{1}{2} \sum_{i \in B_k} \|\varphi(x_i) - c\|^2 - v_k R + d_k \sum_{j \in A_k} \xi_j \\ \text{subject to} \quad & \|\varphi(x_j) - c\|^2 \geq R - \xi_j, \ j \in A_k, \\ & R \geq 0, \ \xi_j \geq 0, \ j \in A_k, \end{aligned} \tag{5}$$

where $A_k = \{i \mid y_i = k\}$ and $B_k = \{i \mid y_i \neq k\}$, and v_k and d_k are parameters.

Problem (5) generates a hypersphere for the k-th class that includes as many training points as possible except for the k-th class. Thus, we can say that the THKSVM employs a "rest-versus-1" structure. The first term of the objective function in (5) aims to place the k-th hypersphere as close as possible to the training points of the other $K-1$ class. The second term of the objective function in (5) is designed to maximize the radius of the k-th hypersphere. The first constraint ensures that the k-th hypersphere does not encompass the training points of the k-th class. Moreover, the variable ξ quantifies an error wherein the k-th hypersphere includes the training points of the k-th class. The third term of the objective function minimizes the sum of such errors. Figure 3 indicates an example hypersphere of the THKSVM.

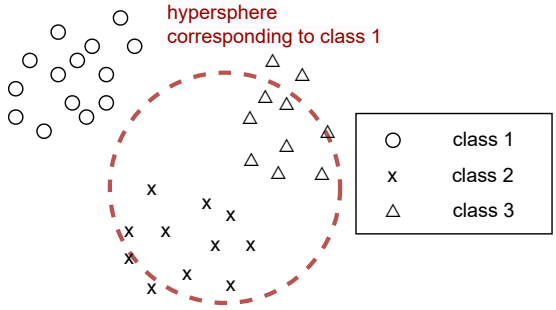

Figure 3: The hypersphere of the THKSVM corresponding to the class 1.

Xu and Guo [3] consider solving Lagrangian dual problem of (5) to calculate the optimal solutions of (5). The dual can be written as

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & t_k \sum_{j_1 \in A_k} \sum_{j_2 \in A_k} \alpha_{j_1} \alpha_{j_2} K(x_{j_1}, x_{j_2}) \\ & - \sum_{j \in A_k} \alpha_j \left(t_k \sum_{i \in B_k} K(x_j, x_i) - K(x_j, x_j) \right) \qquad (6) \\ \text{subject to} \quad & \sum_{j \in A_k} \alpha_j \geq v_k, \quad 0 \leq \alpha_j \leq d_k \ \forall j \in A_k, \end{aligned}$$

where $t_k = 2/(l_B - 2v_k)$ and l_B is the number of training points that do not belong to the k-th class. By considering the KKT conditions of problem (5),

its stationary point (c_k, R_k, ξ_k) can be calculated by

$$\begin{aligned} c_k &= \frac{t_k}{2}\left(\sum_{i\in B_k}\varphi(x_i) - 2\sum_{j\in A_k}\alpha_j\varphi(x_j)\right), \\ R_k &= \frac{1}{|I_k|}\sum_{j\in I_k}\|\varphi(x_j) - c_k\|^2, \\ [\xi_k]_j &= \begin{cases} 0 & \text{if } j \in I_k \cup \{i \mid \alpha_i = 0\}, \\ \frac{1}{|I_k|}\sum_{i\in I_k}\|\varphi(x_i) - c_k\|^2 & \\ \qquad - \|\varphi(x_j) - c_k\|^2 & \text{otherwise,} \end{cases} \end{aligned} \tag{7}$$

where $I_k = \{j \mid 0 < \alpha_j < d_k, j \in A_k\}$.

When predicting a new test point, the point is assigned to a class by

$$\underset{j=1,2,\ldots,K}{\operatorname{argmax}} \frac{\|\varphi(x) - c_j\|^2}{R_j}. \tag{8}$$

In terms of prediction accuracy, the THKSVM competes with other existing methods, such as K-SVCR, Twin-KSVC, and 1-versus-rest TSVM, as shown in [3]. Furthermore, the THKSVM outperforms other methods in terms of computational speed. There are three reasons for this. First, the classifiers required for the THKSVM are minimal, that is, the THKSVM solves problem (5) only K times. In contrast, existing methods such as the K-SVCR and Twin KSVC require $K(K-1)/2$ classifiers because they adopt the "1-versus-1-versus-rest" structure. Second, the THKSVM does not require high-cost computations. Third, optimization problem (5) has fewer constraints than other optimization problems utilized in existing methods

However, the THKSVM has to solve the nonconvex optimization problem (5). In general, (c_k, R_k, ξ_k) calculated by (7), which represents the KKT conditions of (5), is not guaranteed to be a global optimum, although it is a stationary point.

3 Twin hyper-ellipsoidal multiclass classification support vector machine using (QP1QC)

We propose a new method that generates hyperellipsoids to classify multiclass data. Based on the concept of the twin hyper-ellipsoidal support vector machine (TESVM) proposed by Ebrahimpour et al. [2], we extend the THKSVM to classify multiclass data using hyperellipsoids instead of hyperspheres. To this end, we first consider the following problem, which replaces

the Euclidean distance with the Mahalanobis distance in problem (5).

$$\begin{aligned} \underset{c,R,\xi}{\text{minimize}} \quad & \frac{v_k}{l_{B_k}} \sum_{i \in B_k} (\varphi(x_i) - c)^\top \Sigma_{B_k}^{-1} (\varphi(x_i) - c) - R + \frac{d_k}{l_{A_k}} \sum_{j \in A_k} \xi_j \\ \text{subject to} \quad & (\varphi(x_j) - c)^\top \Sigma_{B_k}^{-1} (\varphi(x_j) - c) \geq R - \xi_j, \ j \in A_k, \\ & R \geq 0, \ \xi_j \geq 0, \ j \in A_k, \end{aligned} \tag{9}$$

where Σ_{B_k} is the variance-covariance matrix of all the training points after removing k-th training points, namely, it represents the variance-covariance matrix of training points belonging to B_k, l_{A_k} and l_{B_k} are respectively defined by $l_{A_k} = |A_k|$ and $l_{B_k} = |B_k|$, and v_k and d_k are parameters. Figure 4 shows a hyperellipsoid corresponding to class 1.

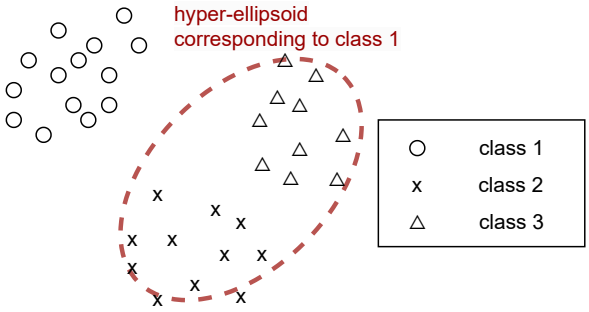

Figure 4: The hyperellipsoid corresponding to the class 1.

Notice that finding a global optimum of problem (9) is challenging because it is nonconvex. Meanwhile, as described below, the following problem obtained by slightly modifying problem (9) has a global optimum if the parameters are appropriately set.

$$\begin{aligned} \underset{c,R,\xi}{\text{minimize}} \quad & \frac{v_k}{l_{B_k}} \sum_{i \in B_k} (\varphi(x_i) - c)^\top \Sigma_{B_k}^{-1} (\varphi(x_i) - c) - R + \frac{d_k}{l_{A_k}} \sum_{j \in A_k} \xi_j^2 \\ \text{subject to} \quad & \sum_{j \in A_k} (\varphi(x_j) - c)^\top \Sigma_{B_k}^{-1} (\varphi(x_j) - c) \geq \sum_{j \in A_k} (R - \xi_j), \\ & R \geq 0, \ \xi_j \geq 0, \ j \in A_k, \end{aligned} \tag{10}$$

Although problem (10) is derived from problem (9), there are several differences as follows:

- The last term of the objective function is the square sum of each element of ξ.

- Instead of imposing inequality constraints separately for each $j \in A_k$, we consider the sum of each inequality as a constraint.

In the proposed method, we construct hyperellipsoids by solving problem (10). This change brings us several benefits described below.

(i) Problem (10) has a global optimum;

(ii) the global optimum can be calculated explicitly.

From now on, we discuss how to solve problem (10). First, we show that problem (10) is equivalent to the following problem in the sense that their optimal solutions coincide.

$$\begin{aligned} \underset{c,R,\xi}{\text{minimize}} \quad & \frac{v_k}{l_{B_k}} \sum_{i \in B_k} (\varphi(x_i) - c)^\top \Sigma_{B_k}^{-1} (\varphi(x_i) - c) - R + \frac{d_k}{l_{A_k}} \sum_{j \in A_k} \xi_j^2 \\ \text{subject to} \quad & \sum_{j \in A_k} (\varphi(x_j) - c)^\top \Sigma_{B_k}^{-1} (\varphi(x_j) - c) \geq \sum_{j \in A_k} (R - \xi_j). \end{aligned} \tag{11}$$

As shown in Theorem 3.2, solving problems (10) and (11) are equivalent. We then observe that problem (11) satisfies the primal Slater condition because there exists a strictly feasible solution $(c, R, \xi) = (0, 0, e)$, and it can be represented as the following quadratic programming over one inequality quadratic constraint (QP1QC):

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & u^\top P_k u - 2 f_k^\top u \\ \text{subject to} \quad & u^\top Q_k u - 2 g_k^\top u \leq \mu_k, \end{aligned}$$

where u, P_k, Q_k, f_k, g_k, and μ are defined as follows:

$$u = \begin{pmatrix} c \\ R \\ \xi \end{pmatrix}, \quad P_k = \begin{pmatrix} v_k \Sigma_{B_k}^{-1} & 0 & O \\ 0^\top & 0 & 0^\top \\ O & 0 & (d_k / l_{A_k}) I \end{pmatrix},$$

$$Q_k = \begin{pmatrix} -l_{A_k} \Sigma_{B_k}^{-1} & 0 & O \\ 0^\top & 0 & 0^\top \\ O & 0 & O \end{pmatrix}, \quad f_k = \begin{pmatrix} \frac{v_k}{l_{B_k}} \sum_{i \in B_k} (\Sigma_{B_k}^{-1} \varphi(x_i)) \\ 1/2 \\ 0 \end{pmatrix},$$

$$g_k = \begin{pmatrix} -\sum_{j \in A_k} (\Sigma_{B_k}^{-1} \varphi(x_j)) \\ -l_{A_k}/2 \\ e/2 \end{pmatrix}, \quad \mu_k = \sum_{j \in A_k} \varphi(x_j)^\top \Sigma_{B_k}^{-1} \varphi(x_j).$$

Fortunately, Hsia et al. [7] has researched this types of problems and provided a sufficient conditions under which its optimal solution exists. By using Theorem 1 and 2, we can prove that problem (11) has an optimal solution.

Theorem 3. *If parameters v_k and d_k are chosen such that $v_k > 1$ and $d_k > 0$, the optimal solution of problem* (11) *is always attainable.*

Proof. We begin by showing that problem (11) is bounded below if v_k and d_k are chosen such that $v_k > 1$ and $d_k > 0$. According to Theorem 1, it is sufficient to prove that there exists σ such that

$$P_k + \sigma Q_k \succeq O, \quad \sigma \geq 0, \quad f_k + \sigma g_k \in \mathcal{R}(P_k + \sigma Q_k). \tag{12}$$

By using the definition of P_k and Q_k, we get

$$P_k + \sigma Q_k = \begin{pmatrix} (v_k - \sigma l_{A_k})\Sigma_{B_k}^{-1} & 0 & O \\ 0^\top & 0 & 0^\top \\ O & 0 & (d_k/l_{A_k})I \end{pmatrix}.$$

Thus, if we set $\sigma = 1/l_{A_k}$, then $P_k + \sigma Q_k \succeq O$ holds. If we define $\bar{u}$ as

$$\bar{u} = \begin{pmatrix} \frac{1}{v_k - 1}\left(\frac{v_k}{l_{B_k}}\sum_{i \in B_k}\varphi(x_i) - \frac{1}{l_{A_k}}\sum_{j \in A_k}\varphi(x_j)\right) \\ 0 \\ \frac{1}{2d_k}e \end{pmatrix},$$

we get

$$\begin{aligned}(P_k + \sigma Q_k)\bar{u} &= \begin{pmatrix} \Sigma_{B_k}^{-1}\left(\frac{v_k}{l_{B_k}}\sum_{i \in B_k}\varphi(x_i) - \frac{1}{l_{A_k}}\sum_{j \in A_k}\varphi(x_j)\right) \\ 0 \\ \frac{1}{2l_{A_k}}e \end{pmatrix} \\ &= f_k + \sigma g_k.\end{aligned}$$

Therefore, $f_k + \sigma g_k \in \mathcal{R}(P_k + \sigma Q_k)$ is satisfied. Since (12) is verified, Theorem 1 implies that problem (11) is bounded below. Meanwhile, we can easily confirm that $I_{\succeq}(P_k, Q_k)$ is represented as follows:

$$\begin{aligned}I_{\succeq}(P_k, Q_k) &= \{\sigma \in \mathbb{R} \mid P_k + \sigma Q_k \succeq O\} \\ &= \{\sigma \mid v_k - \sigma l_{A_k} \geq 0\} \\ &= \{\sigma \mid \sigma \leq v_k / l_{A_k}\}.\end{aligned}$$

It then follows from Theorem 2.2 that problem (11) has a global optimum. □

From Theorem 3, if the parameters v_k and d_k are chosen such that $v_k > 1$ and $d_k > 0$, problem (11) is bounded below and its optimal solution always exists. From now on, we assume that the parameters v_k and d_k satisfy $v_k > 1$ and $d_k > 0$.

Theorem 4. *Suppose that problem* (10) *has a global optimum (c_k, R_k, ξ_k). Then, the following statements hold:*

(a) (c_k, R_k, ξ_k) *is also a global optimum of problem* (11),

(b) (c_k, R_k, ξ_k) *is represented by*

$$
\begin{aligned}
c_k &= \frac{1}{1-v_k}\left(\frac{1}{l_{A_k}}\sum_{j\in A_k}\varphi(x_j) - \frac{v_k}{l_{B_k}}\sum_{i\in B_k}\varphi(x_i)\right),\\
R_k &= \frac{1}{l_{A_k}}\sum_{j\in A_k}\Big((\varphi(x_j)-c_k)^\top\Sigma_{B_k}^{-1}(\varphi(x_j)-c_k)\Big) + \frac{1}{2d_k},\\
[\xi_k]_j &= \frac{1}{2d_k}, \quad j\in A_k.
\end{aligned}
$$

Proof. We first confirm that Cottle's constraint qualification is satisfied in problem (11). Let us denote the objective and constraint functions in problem (11) as

$$
\begin{aligned}
f_k(c,R,\xi) &\coloneqq F_k(c) - R + \frac{d_k}{l_{A_k}}\sum_{j\in A_k}\xi_j^2\\
g_k(c,R,\xi) &\coloneqq \sum_{j\in A_k}(R-\xi_j) - G_k(c),
\end{aligned}
$$

where $F_k(c)$ and $G_k(c)$ are defined as

$$
\begin{aligned}
F_k(c) &\coloneqq \frac{v_k}{l_{B_k}}\sum_{i\in B_k}(\varphi(x_i)-c)^\top\Sigma_{B_k}^{-1}(\varphi(x_i)-c)\\
G_k(c) &\coloneqq \sum_{j\in A_k}(\varphi(x_j)-c)^\top\Sigma_{B_k}^{-1}(\varphi(x_j)-c).
\end{aligned}
$$

Since $\nabla g_k(c,R,\xi) = (-\nabla G_k(c)^\top, l_{A_k}, -e^\top)^\top$, by taking $w = (0^\top, 1/2, e^\top)^\top$, we get

$$
\langle \nabla g_k(c,R,\xi), w\rangle = \frac{l_{A_k}}{2} - l_{A_k} = -\frac{l_{A_k}}{2} < 0.
$$

Thus, Cottle's constraint qualification holds. Since Theorem 3 ensures that problem (11) has a global optimum (c_k, R_k, ξ_k), there exists a Lagrange multiplier λ_k such that $(c_k, R_k, \xi_k, \lambda_k)$ satisfies the KKT conditions, that is,

$$
\begin{aligned}
\nabla F_k(c_k) - \lambda_k \nabla G_k(c_k) &= 0, && (13)\\
-1 + l_{A_k}\lambda_k &= 0, && (14)\\
(2d_k/l_{A_k})[\xi_k]_j - \lambda_k &= 0, \quad j\in A_k, && (15)\\
\lambda_k g_k(c_k,R_k,\xi_k) = 0, \quad g_k(c_k,R_k,\xi_k) &\le 0, \quad \lambda_k \ge 0 && (16)
\end{aligned}
$$

Using (14) and (15) yields $\lambda_k = 1/l_{A_k} > 0$ and $[\xi_k]_j = 1/(2d_k) > 0,\ j \in A_k$. From (13), we obtain

$$\begin{aligned}
&\nabla F_k(c_k) - \lambda_k \nabla G_k(c_k) \\
&= 2(v_k - \lambda_k l_{A_k})\Sigma_{B_k}^{-1} c_k + 2\Sigma_{B_k}^{-1}\left(\lambda_k \sum_{j\in A_k} \varphi(x_j) - \frac{v_k}{l_{B_k}} \sum_{i\in B_k} \varphi(x_i)\right) = 0,
\end{aligned}$$

which implies

$$c_k = \frac{1}{1 - v_k}\left(\frac{1}{l_{A_k}} \sum_{j\in A_k} \varphi(x_j) - \frac{v_k}{l_{B_k}} \sum_{i\in B_k} \varphi(x_i)\right),$$

Moreover, (16) and $\lambda_k > 0$ derive

$$\sum_{j\in A_k} (R_k - [\xi_k]_j) - G_k(c_k) = g_k(c_k, R_k, \xi_k) = 0,$$

namely,

$$\begin{aligned}
R_k &= \frac{1}{l_{A_k}} G_k(c_k) + \frac{1}{2d_k} \\
&= \frac{1}{l_{A_k}} \sum_{j\in A_k} \Big((\varphi(x_j) - c_k)^\top \Sigma_{B_k}^{-1} (\varphi(x_j) - c_k)\Big) + \frac{1}{2d_k} > 0.
\end{aligned}$$

These results mean that (c_k, R_k, ξ_k) is feasible for problem (10). From now on, we show that (c_k, R_k, ξ_k) is an optimal solution of problem (10). Assume, to the contrary, i.e., there exists a feasible solution $(\hat{c}_k, \hat{R}_k, \hat{\xi_k})$ of problem (10) such that

$$f_k(\hat{c}_k, \hat{R}_k, \hat{\xi_k}) < f_k(c_k, R_k, \xi_k). \tag{17}$$

Recall that (c_k, R_k, ξ_k) is an optimal solution of problem (11). Since $(\hat{c}_k, \hat{R}_k, \hat{\xi_k})$ is also feasible to problem (11), we have

$$f_k(\hat{c}_k, \hat{R}_k, \hat{\xi_k}) \geq f_k(c_k, R_k, \xi_k).$$

This contradicts (17). Therefore (c_k, R_k, ξ_k) is an optimal solution of problem (10). □

Now, we notice that the following two models can be constructed. The first one is a model using problem (9), which is a simple extension of THKSVM by using the Mahalanobis distance. We call this model TEKSVM. Meanwhile, the second one is a model utilizing problem (10). We call this model a relaxed THKSVM because it is based on (QP1QC), which relaxes some constraints in problem (9).

In the remainder of the section, we discuss the following:

(i) The interpretation of the relaxed TEKSVM,

(ii) classification for given test data,

(iii) efficient computations for a Mahalanobis distance kernel.

Regarding item (i), we begin by interpreting the objective and constraint functions in problems (9) and (10). As shown in Figure 4, the original problem (9) for the k-th class generates a hyperellipsoid that contains as many training points as possible belonging to classes other than k-th, and is located as far as possible from the points belonging to the k-th class. Moreover, the same interpretation can not be applied to the relaxed problem, i.e., (10) or (11), because the constraints differ from those in (9). However, the following equivalent reformulation of the constraint provides a different interpretation:

$$\frac{1}{l_{A_k}} \sum_{j \in A_k} (\varphi(x_j) - c)^\top \Sigma_{B_k}^{-1} (\varphi(x_j) - c) \geq R - \frac{1}{l_{A_k}} \sum_{j \in A_k} \xi_j.$$

This implies that the average Mahalanobis distance between c and training points belonging to A_k is required to be greater than or equal to $R - (1/l_{A_k}) \sum_{j \in A_k} \xi_j$. In other words, the relaxed TEKSVM considers the average Mahalanobis distance between each training point, rather than the individual Mahalanobis distance.

Next, we consider item (ii). It may be reasonable to adopt a method based on the existing one used in the THKSVM. That is, a new test point x is assigned to the class $j \in \{1, 2, \ldots, K\}$ defined below depending on which of the K hyperellipsoids it lies farthest from.

$$j = \underset{k=1,2,\ldots,K}{\arg\max} \frac{(\varphi(x) - c_k)^\top \Sigma_{B_k}^{-1} (\varphi(x) - c_k)}{R_k}.$$

Finally, we discuss efficient computations related to a Mahalanobis distance kernel calculated in the relaxed TEKSVM. Theorem 4 implies that the relaxed TEKSVM needs to calculate $\varphi(x_i)^\top \Sigma_{B_k}^{-1} \varphi(x_j)$ included in R_k although it does not require any iterative methods to solve problem (10), where x_i and x_j are arbitrary data points. Since computing quadratic forms involves significant computational costs, it is advisable to avoid calculating them directly. To this end, we utilize a Mahalanobis distance kernel defined by $K_{B_k}(x_i, x_j) \coloneqq \varphi(x_i) \Sigma_{B_k}^{-1} \varphi(x_j)$. As is the existing binary case described in [2], we will show that the Mahalanobis distance kernel can be calculated by exploiting a positive definite kernel. Notice that this method requires computing the inverse of the variance-covariance matrix Σ_{B_k} or an approximation thereof. Let training points belonging to B_k denote $x_{B_k}^{(1)}, x_{B_k}^{(2)}, \ldots, x_{B_k}^{(l_{B_k})}$, and let $\varphi(X_{B_k}) = \varphi(x_{B_k}^{(1)}), \varphi(x_{B_k}^{(2)}), \ldots, \varphi(x_{B_k}^{(l_{B_k})})$.

Then, the variance-covariance matrix Σ_{B_k} can be expressed as

$$\Sigma_{B_k} = \varphi(X_{B_k}) J_{B_k} J_{B_k}^\top \varphi(X_{B_k})^\top,$$

where J_{B_k} is a matrix satisfying

$$J_{B_k} J_{B_k}^\top = \frac{1}{l_{B_k}} \left(\mathrm{I} - \frac{1}{l_{B_k}} \boldsymbol{e}\boldsymbol{e}^\top \right).$$

Since Σ_{B_k} can be close to a singular matrix, we use a matrix $\Sigma_{B_k} + \delta I$ instead of Σ_{B_k}, where $\delta > 0$ is a sufficiently small constant. Using Sherman-Morrison-Woodbury formula $(A+BC)^{-1} = A^{-1} - A^{-1}B(\mathrm{I}+CA^{-1}B)^{-1}CA^{-1}$ derives

$$\begin{aligned}
&(\delta \mathrm{I} + \Sigma_{B_k})^{-1} \\
&= \left(\delta \mathrm{I} + (\varphi(X_{B_k}) J_{B_k})(J_{B_k}^\top \varphi(X_{B_k})^\top) \right)^{-1} \\
&= \frac{1}{\delta} \mathrm{I} - \varphi(X_{B_k}) J_{B_k} \left(\mathrm{I} + \frac{1}{\delta} J_{B_k}^\top \varphi(X_{B_k})^\top \varphi(X_{B_k}) J_{B_k} \right)^{-1} J_{B_k}^\top \varphi(X_{B_k})^\top \\
&= \frac{1}{\delta} \mathrm{I} - \frac{1}{\delta} \varphi(X_{B_k}) J_{B_k} \left(\delta \mathrm{I} + J_{B_k}^\top \bar{K}_{B_k} J_{B_k} \right)^{-1} J_{B_k}^\top \varphi(X_{B_k})^\top,
\end{aligned}$$

where $\bar{K}_{B_k} = \varphi(X_{B_k})^\top \varphi(X_{B_k})$. Thus, we can calculate $K_{B_k}(x_i, x_j)$ as

$$\begin{aligned}
&K_{B_k}(x_i, x_j) \\
&= \frac{1}{\delta} K(x_i, x_j) \\
&\quad - \frac{1}{\delta} \varphi(x_i)^\top \varphi(X_{B_k}) J_{B_k} \left(\delta \mathrm{I} + J_{B_k}^\top \bar{K}_{B_k} J_{B_k} \right)^{-1} J_{B_k}^\top \varphi(X_{B_k})^\top \varphi(x_j).
\end{aligned} \tag{18}$$

4 Numerical Experiments

This section discusses numerical results to confirm the effectiveness of the proposed relaxed TEKSVM. We conduct experiments to compare the proposed method with the K-SVCR, Twin-KSVC, THKSVM, and relaxed THKSVM, where the relaxed THKSVM is a THKSVM-based method that classifies data using hyperspheres generated by solving the following problem:

$$\begin{aligned}
\underset{c,R,\xi}{\text{minimize}} \quad & \frac{v_k}{l_{B_k}} \sum_{i \in B_k} \|\varphi(x_i) - c\|^2 - R + \frac{d_k}{l_{A_k}} \sum_{j \in A_k} \xi_j^2 \\
\text{subject to} \quad & \sum_{j \in A_k} \|\varphi(x_j) - c\|^2 \geq \sum_{j \in A_k} (R - \xi_j).
\end{aligned}$$

This problem is similar to (11), and utilizes the Euclidean distance instead of the Mahalanobis distance. When calculating the Mahalanobis distance kernel given by (18), the Gaussian kernel below is employed:

$$K(x_i, x_j) = \exp\left\{ -\frac{\|x_i - x_j\|^2}{2p} \right\}. \tag{19}$$

The Gaussian kernel is also employed for the other existing methods. We conducted experiments using datasets described in in Table 1 and 2. All the methods were implemented with Python 3.8.7 and ran on a machine with 2.8GHz quad-core Intel Core i7 CPU and 16 GB RAM.

Table 1: Details of benchmark datasets used in experiments

Dataset name	♯ of classes	♯ of data points	♯of features
Iris	3	150	4
Wine	3	178	13
Soybean	4	47	35
Teaching evaluation	3	150	5
Ecoli	5	327	7
Hayes roth	3	132	4
Balance	3	625	4

Table 2: Details of large datasets used in experiments

Dataset name	♯ of classes	♯ of data points	♯ of features
yeast	14	2417	103
USPS	10	9298	256

4.1 Experiment procedure

The procedure for the experiments is shown below. The process was iterated five times. As described in Step 3, parameter selection was conducted through the grid search and 5-fold cross-validation. This involved dividing the training data into five subsets, with four used for training the model and one for validation. The process was repeated five times, and the average prediction accuracy on the validation data was used for parameter selection. In the parameter selection process, each parameter was set as shown in Table 3, and the parameter p in the Gaussian kernel, given by (19), was chosen from $\{2^i \mid i = -2, -1, 0, 1, 2\}$. Moreover, the parameter δ in the Mahalanobis distance kernel defined by (18) is selected from $\{10^i \mid i = -1, -2, -4, -6, -8\}$. Note that the parameters v and d in the relaxed THKSVM and relaxed TEKSVM respectively satisfy $v > 1$ and $d > 0$ as mentioned in Theorem 3. Comparing the average accuracy of each parameter combination, the best combination is chosen. The large datasets were used only to emphasize improved computational speed. Thus, for the large datasets, we skipped Step 3 by using prefixed parameters, and compared the running time among all the methods.

Experiment procedure

Step 1. Standardize the data.

Step 2. Split randomly the data into two sets in a ratio of 8 : 2. The first set is used for training, and the second set is for testing.

Step 3. Tune parameters by a grid search and 5-fold cross-validation.

Step 4. Construct the models using the parameters.

Step 5. Classify the test data using the constructed models.

Table 3: Parameter space of four models

Model name	Parameter	Parameter space
K-SVCR	ϵ	$\{0, 0.333, 0.666, 0.999\}$
	$c1$	$\{2^i \mid i = -4, -2, 0, 1, 2, 4, 6\}$
	$c2$	$\{2^i \mid i = -4, -2, 0, 1, 2, 4, 6\}$
Twin-KSVC	ϵ	$\{0, 0.1, 0.2\}$
	$c1$	$\{2^i \mid i = -4, -2, 0, 1, 2, 4, 6, 8\}$
	$c2$	$\{2^i \mid i = -4, -2, 0, 1, 2, 4, 6, 8\}$
THKSVM	v	$\{2^i \mid i = -4, -3, \ldots, 8\}$
	d	$\{2^i \mid i = -8, -7, \ldots, 8\}$
relaxed THKSVM	v	$\{2^i \mid i = 1, 2, \ldots, 8\}$
	d	$\{2^i \mid i = -8, -7, \ldots, 8\}$
relaxed TEKSVM	v	$\{2^i \mid i = 1, 2, \ldots, 8\}$
	d	$\{2^i \mid i = -8, -7, \ldots, 8\}$

4.2 Result comparisons

The results of the numerical experiments are shown in Table 4 and 5. Table 4 reports the average prediction accuracy, along with the standard deviation, and the running time of Steps 4 and 5 over five iterations on the benchmark datasets. Table 5 indicates the average running time of Steps 4 and 5 over five iterations on the large datasets. From Table 4, we obtain the following conclusions:

- In terms of prediction accuracy, the proposed relaxed TEKSVM performs at least as well as the other models on more than half of the datasets. In particular, the relaxed TEKSVM is superior to the THKSVM for all datasets. These results imply that it is effective to utilize hyper-ellipsoids for data classification.

- Comparing the THKSVM with the relaxed THKSVM, we can see that the relaxed version performs slightly better than the ordinary one in

terms of prediction accuracy. This suggests that by relaxing the optimization problems, the optimization process does not treat each training point individually and may reduce the impact of outliers.

- From the perspective of running time, the relaxed TEKSVM and the relaxed THKSVM significantly outperform the existing models. As detailed in Section 3.2, the solutions to problem (11) can be computed analytically, and hence they do not need to use any iterative methods to perform optimization. It is noteworthy that the relaxed TEKSVM is slightly slower than the relaxed THKSVM. This can be attributed to the calculations of the Mahalanobis distance kernel in the relaxed TEKSVM. Nevertheless, the relaxed TEKSVM demonstrates high-speed performance compared to the existing methods.

On the other hand, we summarize the results of Table 5 in the following.

- The relaxed TEKSVM runs significantly faster than the K-SVCR and Twin-KSVC. However, comparing the relaxed TEKSVM with the relaxed THKSVM, the TEKSVM is slower than the THKSVM although this difference cannot be observed on the benchmark datasets. This is because calculations of the Mahalanobis distance kernel in the relaxed TEKSVM take more time than solving the optimization problems in the THKSVM.

- From a computational time perspective, regarding the THKSVM and relaxed THKSVM, the relaxation technique is effective in speeding up calculations.

Table 4: Experiment result on benchmark datasets

Dataset	THKSVM Accuracy(%) Time(s)	K-SVCR Accuracy(%) Time(s)	Twin-KSVC Accuracy(%) Time(s)	relaxed THKSVM Accuracy(%) Time(s)	relaxed TEKSVM Accuracy(%) Time(s) Parameter δ
iris	$91.58 \pm 6.09 \times 10^{-2}$ 0.151	$96.32 \pm 2.68 \times 10^{-2}$ 0.621	$\mathbf{97.37} \pm 3.32 \times 10^{-2}$ 0.542	$92.11 \pm 4.99 \times 10^{-2}$ 0.00379	$95.79 \pm 3.57 \times 10^{-2}$ 0.00911 $\delta = 10^{-1}$
wine	$99.56 \pm 8.89 \times 10^{-3}$ 0.245	$99.11 \pm 1.09 \times 10^{-3}$ 0.634	$99.56 \pm 8.89 \times 10^{-3}$ 0.575	$\mathbf{100} \pm 0.0$ 0.00314	$\mathbf{100.0} \pm 0.0$ 0.0135 $\delta = 10^{-6}$
soybean	$\mathbf{100} \pm 0.0$ 0.070	$\mathbf{100} \pm 0.0$ 0.484	$\mathbf{100.0} \pm 0.0$ 0.384	$\mathbf{100}.00 \pm 0.0$ 0.00283	$\mathbf{100.0} \pm 0.0$ 0.0119 $\delta = 10^{-6}$
teaching evaluation	$53.68 \pm 9.50 \times 10^{-2}$ 0.158	$55.26 \pm 6.44 \times 10^{-2}$ 0.565	$52.63 \pm 1.05 \times 10^{-1}$ 0.564	$53.16 \pm 1.00 \times 10^{-1}$ 0.0044	$\mathbf{57.89} \pm 7.44 \times 10^{-2}$ 0.0154 $\delta = 10^{-4}$
ecoli	$84.88 \pm 2.51 \times 10^{-2}$ 0.284	$86.59 \pm 1.09 \times 10^{-2}$ 5.340	$80.73 \pm 9.10 \times 10^{-3}$ 4.943	$86.34 \pm 2.10 \times 10^{-2}$ 0.0330	$\mathbf{86.61} \pm 1.951 \times 10^{-2}$ 0.0613 $\delta = 10^{-4}$
balance scale	$90.57 \pm 9.36 \times 10^{-3}$ 0.508	$92.36 \pm 4.43 \times 10^{-2}$ 3.838	$\mathbf{96.18} \pm 1.14 \times 10^{-2}$ 3.383	$90.57 \pm 1.58 \times 10^{-2}$ 0.0144	$91.72 \pm 1.34 \times 10^{-2}$ 0.131 $\delta = 10^{-4}$
hayes roth	$63.03 \pm 1.17 \times 10^{-1}$ 0.137	$\mathbf{80.00} \pm 3.09 \times 10^{-2}$ 0.485	$74.54 \pm 6.23 \times 10^{-2}$ 0.452	$73.33 \pm 1.21 \times 10^{-2}$ 0.00346	$76.37 \pm 2.97 \times 10^{-2}$ 0.0102 $\delta = 10^{-1}$

Table 5: Experiment result on large datasets

Dataset	THKSVM	K-SVCR	Twin-KSVC	relaxed THKSVM	relaxed TEKSVM
	Time(s)	Time(s)	Time(s)	Time(s)	Time(s)
yeast	3.022	2.281×10^3	7.461×10^2	0.646	3.835×10^1
USPS	2.302×10^1	1.006×10^5	1.711×10^4	1.087×10^1	1.828×10^3

5 Conclusion

In this paper, we proposed a novel relaxed twin hyper-ellipsoidal multiclass classification support vector machine (relaxed TEKSVM). The proposed method generates hyperellipsoids by solving optimization problems called the (QP1QC), unlike other existing methods. Although (QP1QC) is non-convex, it possesses at least one global optimum. Moreover, the optimum can be represented explicitly, enabling us to solve the (QP1QC) quickly. In the numerical experiments on benchmark datasets, the proposed method was competitive with some existing methods in terms of accuracy while significantly improving computational speed.

References

[1] Cortes, C., Vapnik, V.: Support-Vector Networks. Mach. Learn. **20**, 273–297 (1995)

[2] Ebrahimpour, Z., Wan, W., Khoojine, A.S., Hou, L.: Twin Hyper-Ellipsoidal Support Vector Machine for Binary Classification. IEEE Access **8**, 87341–87353 (2020)

[3] Xu, Y., Guo, R.: A Twin Hyper-Sphere Multi-Class Classification Support Vector Machine. J. Intell. Fuzzy Syst. **27**, 1783–1790 (2014)

[4] Tanveer, M., Rajani, T., Rastogi, R., Shao, Y.H., Ganaie, M.A.: Comprehensive Review on Twin Support Vector Machines. Ann. Oper. Res. **7**, 1–46 (2022)

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

[6] Xinjun P., Dong X.: A twin-hypersphere support vector machine classifier and the fast learning algorithm. Inf. Sci. **221**, 12–27 (2013)

[7] Hsia, Y., Lin, G.-X., Sheu, R.-L.: A Revisit to Quadratic Programming with One Inequality Quadratic Constraint via Matrix Pencil. Pac. J. Optim **10**, 461–481 (2014)

[8] Angulo C, Parra X, Catal A: K-SVCR: A Support Vector Machine for Multi-Class Classification. Neurocomputing **55**, 57-–77 (2003)

[9] Jayadeva, Khemchandani, R., Chandra, S.: Twin Support Vector Machines for Pattern Classification. IEEE T. Pattern. Anal. **29**, 905–910 (2007)

[10] Xu, Y., Guo, R., Wang, L.: A Twin Multi-Class Classification Support Vector Machine. Cogn. Comput. **5**, 580—588 (2013)