OPTIMIZATION ALGORITHMS FOR NON-PARALLEL SUPPORT VECTOR MACHINES AND ITS APPLICATIONS

M. Sc. Thesis

By

ANSHUL SHARMA

DISCIPLINE OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY INDORE

MAY 2018

OPTIMIZATION ALGORITHMS FOR NON-PARALLEL SUPPORT VECTOR MACHINES AND ITS APPLICATIONS

A THESIS

Submitted in partial fulfillment of the requirements

for the award of the degree

of
Master of Science

by

ANSHUL SHARMA

DISCIPLINE OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY INDORE

MAY 2018

Acknowledgements

I owe my deep gratitude to my thesis supervisor Dr. M. Tanveer, for providing me an opportunity to do the project and giving me all support, guidance and also for sharing his pearls of wisdom with me during the entire course of my work without which it would be impossible to complete this thesis.

I am thankful and fortunate enough to get constant encouragement, support and guidance from the Research Scholar, Bharat Richhariya, Mathematics Department, IIT Indore, which helped me in successfully completing my work. Inspite of his tight schedule, he was always available for consulting.

Sincere gratitude is also extended to the PSPC members, Dr. Niraj Kumar Shukla and Prof. Ram Bilas Pachori, for their valuable remarks, suggestion and questionnaires.

I would also like to thank Dr. Sk. Safique Ahmad, Head-Discipline of Mathematics, for ensuring that the research lab is facilitated with the state-of-the-art computers to carry out various researches with ease.

I would also like to thank the Mathematics Department of IIT Indore, my lab-mates and friends for their encouragement and support and those who have directly or indirectly helped me in completing my work.

Anshul Sharma
1603141001
Discipline of Mathematics
Indian Institute of Technology Indore

Abstract

Least squares twin multi-class classification support vector machine (LST-KSVC) [18] and K-nearest neighbor-based weighted multi-class twin support vector machine (KWMTSVM) [23] are novel multi-class classifiers based on the least squares twin support vector machine (LSTSVM) [17] and twin support vector machine (TSVM) [14] respectively. LST-KSVC and KWMTSVM obtain two non-parallel hyperplanes for focused classes by solving a system of linear equations and two small size quadratic programming problems (QPPs) respectively. Local information of data points is neglected in LST-KSVC, and optimal hyperplanes are constructed by assuming that each data point contribute the same weight but generally each data point have different predominance on the optimal hyperplanes. KWMTSVM solves the QPPs which consume more time to compute the optimal hyperplanes. To reduce the drawbacks of the above algorithms we proposed a novel algorithm based on least squares version of KWMTSVM in chapter 4 of this thesis termed as **Least squares K-nearest neighbor-based weighted multi-class twin support vector machine (LS-KWMTSVM)**. In our algorithm to enterprise the local information we introduce the weight matrix $D_i (i = 1, 2)$ in the objective function of QPPs and to enterprise the inter class information we use F_{v1}, F_{v2}, H_v weight vectors in constraints. If any component of vectors is zero then corresponding constraint is redundant so we can escape it. Evacuation of redundant constraints and solving a system of linear equation instead of QPPs makes our algorithm faster than KWMTSVM. LS-KWMTSVM evaluates all the training data points into a "1-versus-1-versus-rest" structure, so it generates ternary output which helps to deal with imbalance datasets.

Classical TSVM uses hinge loss function [4] which is sensitive to noise and unstable for re-sampling. To elevate the performance of TSVM, we introduce a novel algorithm in chapter 5 of this thesis termed as **General twin support vector machine with pinball loss**. In our proposed algorithm we use quantile distance [15, 20] and pinball loss function [15,20] instead of shortest distance and hinge loss respectively. We justify theoretically and experimentally that our proposed algorithm is noise insensitive i.e., it give better classification results for noise corrupted data.

Contents

List of Figures

List of Tables

Abbreviations

K.K.T.	**K**arush-**K**uhn **T**ucker
KWMTSVM	**K**-nearest neighbor-based **W**eighted **M**ulti-class **T**win **S**upport **V**ector **M**achine
LSTSVM	**L**east **S**quares **Twin S**upport **V**ector **M**achine
SVM	**S**upport **V**ector **M**achine
QPP	**Q**uadratic **P**rogramming **P**roblem
TPMSVM	**T**win **P**arametric **M**argin **S**upport **V**ector **M**achine
TSVM	**T**win **S**upport **V**ector **M**achine

Chapter 1

Introduction

Support vector machine (SVM) and its variants [4, 14, 17] are classification tools for supervised learning and its applications. The central idea of SVM is to construct the optimal hyperplane between two different given classes. The optimal hyperplane is defined as the one giving maximum margin between the training data points that are close to the supporting hyperplanes. In contrast with other machine learning techniques like artificial neural network (ANN) which implements empirical risk minimization principle, SVM implements the structural risk minimization principle. SVM solves convex programming problem, reassure that once a optimal solution exists, it is unique and global.

In general there are two methods to develop classifiers:
1. Parametric approach [8], in which antecedent knowledge about data distribution is given.
2. A non-parametric method [8], where no antecedent knowledge is given. Support vector machines are typical non-parametric classifiers.

1.1 Linearly Separable Problem

Generally, a classification problem is considered as follows:
For a given training set $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell)\}$, where $x_i \in \mathbb{R}^n$ are training

points (inputs), its components are called features, $y_i \in Y = \{-1, 1\}$ are corresponding class labels (outputs) for $i = 1, 2, \ldots, \ell$. Objective of classification problem is to find a real valued function $f(x)$ in $\mathbb{R}$ which can determine the value of y (class label of x) for any given x by the decision function $h(x) = sgn(f(x))$.

Definition 1 [7]: Consider the training set $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell)\}$, where $x_i \in \mathbb{R}^n$, $y_i \in Y = \{-1, 1\}$, $i = 1, 2, \ldots, \ell$. If there exist $w \in \mathbb{R}^n$, $b \in \mathbb{R}$ and a positive real number ϵ such that for any subscript i with $y_i = 1$, we have $w^T x_i + b \geq \epsilon$ and for any subscript i with $y_i = -1$, we have $w^T x_i + b \leq \epsilon$, then we say the training set and its corresponding classification problem are linearly separable. If a classification problem is not linearly separable then we call it nonlinearly separable problem. Example of linearly and nonlinearly separable problems shown in Figure 1.1 and 1.2 respectively.

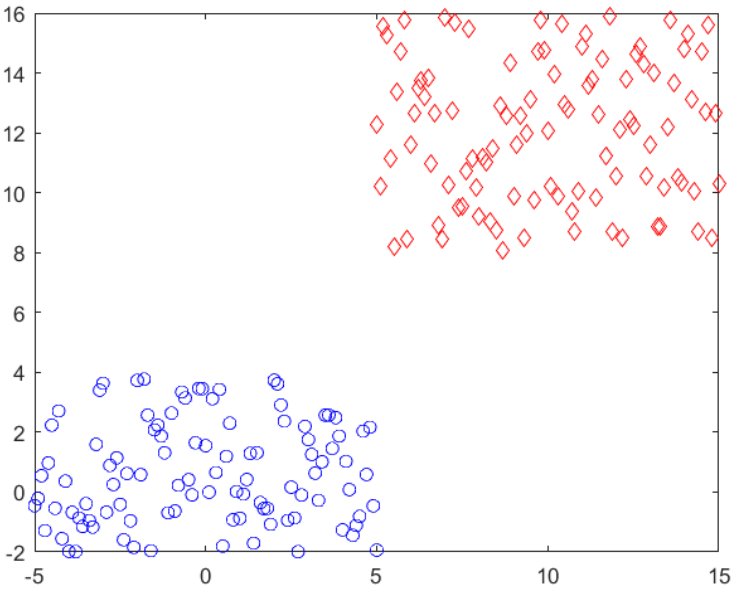

FIGURE 1.1: Example of linearly separable data.

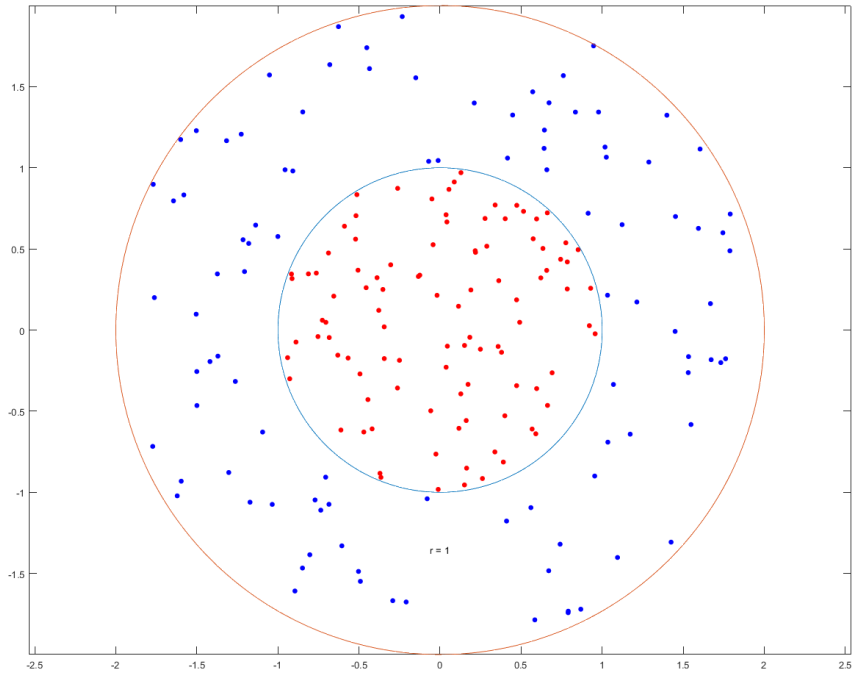

FIGURE 1.2: Example of nonlinearly separable data.

1.2 Karush-Kuhn Tucker (K.K.T.) Conditions

Consider the following constrained optimization problem with inequality constraints:

$$\begin{aligned} &\min \ f(x) \\ &s.t. \ g_i(x) \leq 0, \ i = 1, 2, \ldots, m, \end{aligned} \tag{1.2.1}$$

where f and $g_i : \mathbb{R}^n \to \mathbb{R}$ are continuously differentiable functions.

Necessary Part of K.K.T. Conditions [16]: Let $\tilde{x}$ be local minimization point of the problem (1.2.1). Then there exists multipliers (called K.K.T. multipliers) $\tilde{\lambda}_i$, $i = 1, 2, \ldots, m$ such that the following conditions are holds:

1. $\nabla f(\tilde{x}) + \sum_{i=1}^{m} \tilde{\lambda}_i \nabla g_i(\tilde{x}) = 0.$
2. $\nabla g_i(\tilde{x}) = 0$, where $i = 1, 2, \ldots, m.$
3. $\tilde{\lambda}_i \nabla g_i(\tilde{x}) = 0$, where $i = 1, 2, \ldots, m.$

4. $\tilde{\lambda_i} \geq 0 \; \forall \, i.$

these conditions are called K.K.T. conditions.

Sufficient Part of K.K.T. Conditions [16]: Let $(\tilde{x}, \tilde{\lambda_1}, \tilde{\lambda_2}, \ldots, \tilde{\lambda_m})$ satisfies the K.K.T. conditions (1)-(4). Let f, $g_i \, (\forall \, i)$ be differential convex function, then $\tilde{x}$ is a global solution of the optimization problem (1.2.1).

1.3 Linear Support Vector Machine

Let the data points to be classified denoted by the set T. Then we need to obtain $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that

$$w^T x_i + b \geq 1 \text{ for } y_i = 1 \text{ and } w^T x_i + b \leq -1 \text{ for } y_i = -1. \tag{1.3.1}$$

The optimal hyperplane $w^T x + b = 0$, lies midway between the supporting hyperplanes given by:

$$w^T x + b = 1 \text{ and } w^T x + b = -1 \tag{1.3.2}$$

and classify the two classes from each other with margin of $\frac{1}{\|w\|}$ on each side. Data points lies on the supporting hyperplanes given by (1.3.2) are termed as support vectors. The classifier is obtained by maximizing the margin, it is equivalent to the following optimization problem:

$$\begin{aligned} &\min_{w,\, b} \; \frac{1}{2}\|w\|^2 \\ &s.t. \; y_i(w^T x_i + b) \geq 1, \; i = 1, 2, \ldots, \ell. \end{aligned} \tag{1.3.3}$$

When the two classes are not strictly linearly separable, in order to relax the condition to separate all data point correctly, allow the existence of data points that violate the constraints $y_i(w^T x_i + b) \geq 1$ by introducing the slack variables $\xi_i \geq 0, \;\; i = 1, \, 2, \, \ldots, \ell$. In order to reduce this violation, avoid making ξ_i too large, superimpose a penalty

upon them in objective function. Then formulation of the linear SVM is given by [4]:

$$\begin{aligned} \min_{w,\,b,\,\xi} \ & \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{\ell}\xi_i \\ s.t. \ & y_i(w^T x_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \ \ i = 1, 2, \ldots, \ell, \end{aligned} \tag{1.3.4}$$

where $\xi = (\xi_1,\, \xi_2, \ldots,\, \xi_\ell)^T$ is slack variable and $c > 0$ is a penalty parameter. The parameter c determines the weight between the two terms $\|w\|^2$ and $\sum_{i=1}^{\ell}\xi_i$.

The Lagrange function corresponding to the problem (1.3.4) is given by:

$$L(w,\, b,\, \xi,\, \alpha,\, \beta) = \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{\ell}\xi_i - \sum_{i=1}^{\ell}\beta_i\xi_i - \sum_{i=1}^{\ell}\alpha_i(y_i(w^T x_i + b) + \xi_i - 1), \tag{1.3.5}$$

where $\alpha = (\alpha_1,\, \alpha_2, \ldots,\, \alpha_\ell)^T \geq 0$ and $\beta = (\beta_1,\, \beta_2, \ldots,\, \beta_\ell)^T \geq 0$ are Lagrange multipliers. Using the K.K.T. optimality conditions [16] we obtain:

$$w - \sum_{i=1}^{\ell}\alpha_i y_i x_i = 0, \ \sum_{i=1}^{\ell}\alpha_i y_i = 0 \text{ and } c - \alpha_i - \beta_i = 0 \ \ \forall\, i = 1, 2, \ldots, \ell. \tag{1.3.6}$$

By using (1.3.5) and (1.3.6), the dual formulation of (1.3.4) is given by:

$$\begin{aligned} \max_{\alpha} \ & \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j y_i y_j x_i^T x_j \\ s.t. \ & \sum_{i=1}^{\ell}\alpha_i y_i = 0, \ 0 \leq \alpha_i \leq c, \ \forall\, i = 1, 2, \ldots, \ell. \end{aligned} \tag{1.3.7}$$

Remark: Assume that α^* is a solution of dual problem. The input x_i, associated to the training point $(x_i,\, y_i)$, is called **support vector** if the corresponding component α_i^* of α^*, is strictly positive.

Algorithm 1: *Linear Support Vector Machine* [7]

- Input the training set $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell)\}$, where $x_i \in \mathbb{R}^n$ and $y_i \in Y = \{-1, 1\}$, $i = 1, 2, \ldots, \ell$.
- Choose an appropriate penalty parameter $c > 0$.
- Formulate the convex quadratic programming problem:

$$\max_{\alpha} \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j x_i^T x_j$$
$$s.t. \sum_{i=1}^{\ell} \alpha_i y_i = 0,\ 0 \leq \alpha \leq c,\ i = 1, 2, \ldots, \ell$$

  and obtain the optimal solution $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)^T$.
- Compute (w, b) by $w = \sum_{i=1}^{\ell} \alpha_i y_i x_i$ and $b = y_j - \sum_{i=1}^{\ell} \alpha_i y_i x_i^T x_j$, j is the j^{th} component of α such that $\alpha_j > 0$.
- Construct the separating hyperplane $w^T x + b = 0$ and its associated decision function $h(x) = sgn(f(x))$ where

$$f(x) = w^T x + b = \sum_{i=1}^{\ell} \alpha_i y_i x_i^T x + b. \tag{1.3.8}$$

1.4 Nonlinear Support Vector Machine

There are some classification problem which cannot be classified by linear hyperplane in the input space. For example, consider 20 training points in $\mathbb{R}^2$ as shown in Figure 1.3 [7]. In this figure, " + " and " ∘ " represent inputs corresponding to label $y_i = +1$ and $y_i = -1$ respectively. The relevant separating curve for this problem looks like an ellipse centered at the origin in the $([x]_1, [x]_2)$ plane, that is a nonlinear classifier. In fact, this ellipse can be converted into a linear hyperplane by using a nonlinear map $\phi(x) = X$ from $[x]_1 O [x]_2$ plane to the $[X]_1 O [X]_2$ plane as shown in Figure 1.3.

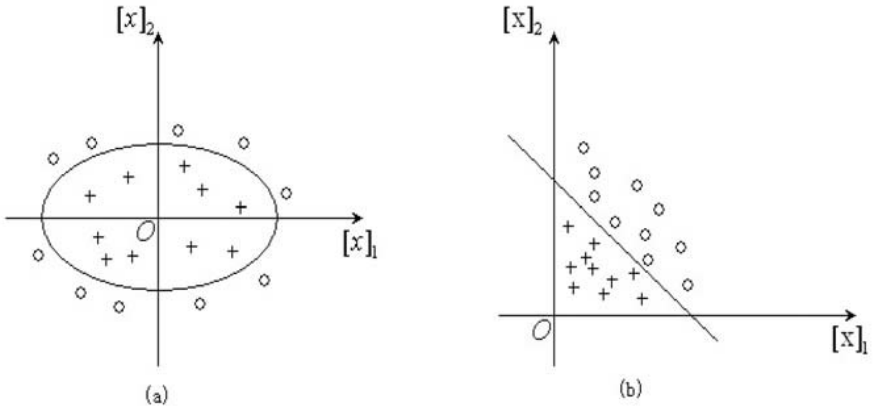

FIGURE 1.3: Input space (a) and features space (b) [7].

Definition 2 [1]: Assume x and $x' \in \mathbb{R}^\ell$ in input space then $K(x,\, x') = \phi(x)^T\phi(x')$ is called *Kernel function* where $\phi(x)$ is a nonlinear map from input space to higher dimensional space where data points are linearly separable. The advantage of using kernels is that we do not need to treat the higher dimensional feature space explicitly.

The optimization problem for the nonlinear SVM is as follows [4]:

$$\begin{aligned} &\min_{w,\,b,\,\xi} \ \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{\ell}\xi_i \\ &s.t. \ \ y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \ \ \xi_i \geq 0, \ \ i = 1, 2, \ldots, \ell, \end{aligned} \tag{1.4.1}$$

where $\xi = (\xi_1,\, \xi_2, \ldots,\, \xi_\ell)^T$ is slack variable and $c > 0$ is a penalty parameter.

The Lagrange function corresponding to the problem (1.4.1) is given by:

$$\begin{aligned} L(w,\, b,\, \xi,\, \alpha,\, \beta) = \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{\ell}\xi_i - \sum_{i=1}^{\ell}\alpha_i(y_i(w^T\phi(x_i) + b) + \xi_i - 1) \\ - \sum_{i=1}^{\ell}\beta_i\xi_i. \end{aligned} \tag{1.4.2}$$

Using the K.K.T. optimality conditions, the dual formulation of (1.4.1) is given by:

$$\begin{aligned} \max_{\alpha} \ & \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \\ s.t. \ & \sum_{i=1}^{\ell} \alpha_i y_i = 0, \ 0 \leq \alpha_i \leq c, \ \forall \, i = 1, 2, \ldots, \ell. \end{aligned} \tag{1.4.3}$$

Algorithm 2 *Nonlinear Support Vector Machine* [7]

- Input the training set $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i)\}$, where $x_i \in \mathbb{R}^n$, $y_i \in Y = \{-1, 1\}$, $i = 1, 2, \ldots, \ell$.
- Choose an appropriate map $\phi : \mathbb{R}^\ell \to$ *Hilbert space* where data points are linearly separable and a penalty parameter $c > 0$.
- Formulate the convex quadratic programming problem:

$$\begin{aligned} \max_{\alpha} \ & \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_i) \\ s.t. \ & \sum_{i=1}^{\ell} \alpha_i y_i = 0, \ 0 \leq \alpha_i \leq c, \ i = 1, 2, \ldots, \ell, \end{aligned}$$

  and obtain the optimal solution $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)^T$.
- Compute (w, b) by $w = \sum_{i=1}^{\ell} \alpha_i y_i \phi(x_i)$ and $b = y_j - \sum_{i=1}^{\ell} \alpha_i y_i \phi(x_i)^T \phi(x_j)$, j is the j^{th} component of α such that $\alpha_j > 0$.
- Construct the separating hyperplane $w^T \phi(x) + b = 0$ and its associated decision function $h(x) = sgn(f(x))$ where

$$f(x) = w^T \phi(x) + b = \sum_{i=1}^{\ell} \alpha_i y_i \phi(x_i)^T \phi(x) + b. \tag{1.4.4}$$

Remark: Only difference between the algorithm 1 and 2 is that we replace inner product $(x_i^T x_j)$ and $(x_i^T x)$ by inner product in feature space $(\phi(x_i)^T \phi(x_j))$ and $(\phi(x_i)^T \phi(x))$ respectively.

Chapter 2

Binary-class Classifiers

2.1 Twin Support Vector Machine (TSVM)

SVM, being computationally powerful tool for supervised learning, it is widely used for classification. The conventional SVM has drawback such as high computation complexity, approximately of order $\mathcal{O}(\ell^3)$, where ℓ is number of data points. To resolve this challenge, Jayadeva et al. [14] proposed a variant of SVM termed as twin support vector machine (TSVM). The formulation of SVM requires all data points but in TSVM they are distributed in such a way one class gives the constraints to the other class and vice verse. So, TSVM solves two smaller size QPPs rather than one large size QPP. Consider a binary classification problem which classify data points belonging to classes +1 and -1 are represented by matrices A and B respectively. Let data points belongs to class +1 and -1 are ℓ_1 and ℓ_2 respectively in the n-dimensional real space $\mathbb{R}^n$.

2.1.1 Linear TSVM

Linear TSVM seeks for a pair of non-parallel hyperplanes

$$f^+(x) = w_+^T x + b_+ = 0 \text{ and } f^-(x) = w_-^T x + b_- = 0, \tag{2.1.1}$$

such that each hyperplane is close to one class and far from the other class, where $w_+ \in \mathbb{R}^n$, $w_- \in \mathbb{R}^n$, $b_+ \in \mathbb{R}$ and $b_- \in \mathbb{R}$. The formulation of linear TSVM can be written as follows [14]:

$$\begin{aligned} \min_{w_+, b_+, \xi_1} \ & \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 \\ s.t. \quad & -(Bw_+ + e_2 b_+) + \xi_1 \geq e_2, \ \xi_1 \geq 0 \end{aligned} \tag{2.1.2}$$

and

$$\begin{aligned} \min_{w_-, b_-, \xi_2} \ & \frac{1}{2}\|Bw_- + e_2 b_-\|^2 + c_2 e_1^T \xi_2 \\ s.t. \quad & (Aw_- + e_1 b_-) + \xi_2 \geq e_1, \ \xi_2 \geq 0, \end{aligned} \tag{2.1.3}$$

where c_1, c_2 are positive parameters, e_1, e_2 are standard unit vectors of appropriate dimensions and ξ_1, ξ_2 are slack variables. We observe that TSVM is four times faster than the conventional support vector machine. This is because time complexity of usual support vector machine is approximately $\mathcal{O}(\ell^3)$, where ℓ is the total number of data points, and TSVM solves two QPPs (2.1.2) and (2.1.3) each is roughly of size $\ell/2$, then ratio of run times is

$$\left[(\ell^3)/\left(2 \times \left(\frac{\ell}{2}\right)^3\right)\right] = 4.$$

The Lagrange function corresponding to the problem (2.1.2) is given by :

$$\begin{aligned} L(w_+, b_+, \xi_1, \alpha, \beta) = & \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 - \beta^T \xi_1 \\ & - \alpha^T(-(Bw_+ + e_2 b_+) + \xi_1 - e_2), \end{aligned} \tag{2.1.4}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are Lagrange multipliers. Using the K.K.T. optimality conditions we obtain:

$$A^T(Aw_+ + e_1 b_+) + B^T \alpha = 0, \ e_1^T(Aw_+ + e_1 b_+) + e_2^T \alpha = 0 \text{ and } c_1 e_2 - \alpha - \beta = 0. \tag{2.1.5}$$

By using equation (2.1.5), we obtain:

$$H^T H z_+ + G^T \alpha = 0, \;\; i.e. \;\; z_+ = -(H^T H) G^T \alpha, \tag{2.1.6}$$

where $H = [A \; e_1], \; G = [B \; e_2]$ and $z_+ = [w_+ \; b_+]^T$.

Although $H^T H$ is always positive semi-definite, it is possible that it may not be well conditioned in some situations. So, introduce a regularization term [21] δI, $\delta > 0$, to take care of problems due to ill-conditioning of $H^T H$. Here, I is an identity matrix of appropriate dimension. Therefore (2.1.6) is modified to

$$z_+ = -(H^T H + \delta I)^{-1} G^T \alpha. \tag{2.1.7}$$

By using (2.1.4) and (2.1.5), the dual formulation of (2.1.2) is given by:

$$\begin{aligned} &\max_{\alpha} \; e_2^T \alpha - \frac{1}{2} \alpha^T G (H^T H)^{-1} G^T \alpha \\ &s.t. \;\; 0 \leq \alpha \leq c_1. \end{aligned} \tag{2.1.8}$$

Similarly, the dual formulation of equation (2.1.3) is given by:

$$\begin{aligned} &\max_{\beta} \; e_1^T \beta - \frac{1}{2} \beta^T H (G^T G)^{-1} H^T \beta \\ &s.t. \;\; 0 \leq \beta \leq c_2. \end{aligned} \tag{2.1.9}$$

A new data point $x \in \mathbb{R}^n$ is assigned to class $i (i = +1, -1)$ depending on which of the hyperplanes in (2.1.1) is closer to x, i.e.,

$$class(i) = \text{sign}\Big(\frac{{w_+}^T x + b_+}{\|w_+\|} + \frac{{w_-}^T x + b_-}{\|w_-\|} \Big). \tag{2.1.10}$$

2.1.2 Nonlinear TSVM

We can extend linear TSVM to the nonlinear TSVM by considering the following kernel generated surfaces:

$$K(x^T, D^T)u_+ + b_+ = 0 \text{ and } K(x^T, D^T)u_- + b_- = 0, \tag{2.1.11}$$

where $D = [A; B]$; u_+, $u_- \in \mathbb{R}^n$ and K is an arbitrary kernel function. Formulation of the nonlinear TSVM is as follows [14]:

$$\begin{aligned} \min_{u_+, b_+, \xi_1} \ & \frac{1}{2}\|K(A, D^T)u_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 \\ s.t. \quad & -(K(B, D^T)u_+ + e_2 b_+) + \xi_1 \geq e_2, \ \xi_1 \geq 0, \end{aligned} \tag{2.1.12}$$

$$\begin{aligned} \min_{u_-, b_-, \xi_2} \ & \frac{1}{2}\|K(B, D^T)u_- + e_2 b_-\|^2 + c_2 e_1^T \xi_2 \\ s.t. \quad & (K(A, D^T)u_- + e_1 b_-) + \xi_2 \geq e_1, \ \xi_2 \geq 0. \end{aligned} \tag{2.1.13}$$

Similar to the linear TSVM by introducing the Lagrange multipliers α, β and using the K.K.T. conditions we can derive dual of (2.1.12) and (2.1.13) as follows:

$$\begin{aligned} \max_{\alpha} \ & e_2^T \alpha - \frac{1}{2}\alpha^T Q(P^T P)^{-1} Q^T \beta \\ s.t. \ & 0 \leq \alpha \leq c_1 \end{aligned} \tag{2.1.14}$$

and

$$\begin{aligned} \max_{\beta} \ & e_1^T \beta - \frac{1}{2}\beta^T P(Q^T Q)^{-1} P^T \beta \\ s.t. \ & 0 \leq \beta \leq c_2, \end{aligned} \tag{2.1.15}$$

where $P = [K(A,\, D^T)\; e_1]$ and $Q = [K(B,\, D^T)\; e_2]$. Finally solutions of (2.1.14) and (2.1.15) are given by

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -(P^T P + \delta I)^{-1} Q^T \alpha \text{ and } \begin{bmatrix} u_- \\ b_- \end{bmatrix} = (Q^T Q + \delta I)^{-1} P^T \beta, \tag{2.1.16}$$

where δI, $\delta > 0$ is a regularization term used to avoid the ill-conditioning of matrices $P^T P$ and $Q^T Q$.

A new data point $x \in \mathbb{R}^n$ is assigned to class $i (i = +1, -1)$ depending on which of the kernel generate surface in (2.1.11) is closer to x, i.e.,

$$class(i) = \text{sign}\Big(\frac{K(x^T, D^T)u_+ + b_+}{\|u_+\|} + \frac{K(x^T, D^T)u_- + b_-}{\|u_-\|}\Big). \tag{2.1.17}$$

2.2 Least Squares Twin Support Vector Machine (LSTSVM)

The idea of TSVM is extended to LSTSVM [17] by replacing the inequality constraints in TSVM with equality constraints and taking the squares of 2-norm of slack variables instead of 1-norm in TSVM. As a result solution of LSTSVM follows from solving a system of linear equations which makes the algorithm simple and fast.

2.2.1 Linear LSTSVM

Linear LSTSVM seeks for a pair of non-parallel hyperplanes given in equation (2.1.1). Formulation of the linear LSTSVM is given by [17]:

$$\begin{aligned} \min_{w_+, b_+, \xi_1} \; & \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + \frac{c_1}{2}\|\xi_1\|^2 \\ s.t. \quad & -(Bw_+ + e_2 b_+) + \xi_1 = e_2 \end{aligned} \tag{2.2.2}$$

and

$$\min_{w_-, b_-, \xi_2} \frac{1}{2}\|Bw_- + e_2 b_-\|^2 + \frac{c_2}{2}\|\xi_2\|^2$$
$$s.t. \quad (Aw_- + e_1 b_-) + \xi_2 = e_1. \tag{2.2.3}$$

After substituting the value of ξ_1 into the objective function of (2.2.2) leads to

$$\min_{w_+, b_+} \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + \frac{c_1}{2}\|e_2 + Bw_+ + e_2 b_+\|^2. \tag{2.2.4}$$

Using the K.K.T. conditions, the solution of (2.2.2) is given by

$$\begin{bmatrix} w_+ \\ b_+ \end{bmatrix} = -\Big(\frac{1}{c_1} H^T H + G^T G\Big)^{-1} G^T e_1. \tag{2.2.5}$$

Similarly, solution of (2.2.3) is given by

$$\begin{bmatrix} w_- \\ b_- \end{bmatrix} = \Big(\frac{1}{c_2} G^T G + H^T H\Big)^{-1} H^T e_2. \tag{2.2.6}$$

Thus the linear LSTSVM completely solves the classification problem with inverse of two matrices of order $(n+1) \times (n+1)$ where $n << \ell$. A new data point $x \in \mathbb{R}^n$ is assigned to class $i (i = +1, -1)$ according to the equation (2.1.10).

2.2.2 Nonlinear LSTSVM

Linear LSTSVM can be extended to the nonlinear LSTSVM by considering two kernel generated surfaces given in (2.1.11). Formulation of the nonlinear LSTSVM is given by [17]:

$$\min_{u_+, b_+, \xi_1} \frac{1}{2}\|K(A, D^T)u_+ + e_1 b_+\|^2 + \frac{c_1}{2}\|\xi_1\|^2$$
$$s.t. \quad -(K(B, D^T)u_+ + e_2 b_+) + \xi_1 = e_2 \tag{2.2.7}$$

and

$$\min_{u_-,b_-,\xi_2} \frac{1}{2}\|K(B,D^T)u_- + e_2b_-\|^2 + \frac{c_2}{2}\|\xi_2\|^2$$
$$s.t. \quad (K(A,D^T)u_- + e_1b_-) + \xi_2 = e_1. \tag{2.2.8}$$

Similar to the linear LSTSVM, solution of QPPs (2.2.7) and (2.2.8) can be derived as follows:

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -(Q^TQ + \frac{1}{c_1}P^TP)^{-1}Q^Te_2 \text{ and } \begin{bmatrix} u_- \\ b_- \end{bmatrix} = (P^TP + \frac{1}{c_2}Q^TQ)^{-1}P^Te_1.$$

It is observed that the solution of the nonlinear LSTSVM requires the inverse of matrix of order $(\ell+1)\times(\ell+1)$. However using Sherman-Morrison-Woodbury (SMW) formula [10], solution of nonlinear LSTSVM given by using the inverse of three matrices of smaller order rather than $(\ell+1)\times(\ell+1)$. The class $i(i=+1,-1)$ is assigned to a new data point x according to the equation (2.1.17).

2.3 Twin Support Vector Machine With Pinball Loss (Pin-TSVM)

2.3.1 Loss Functions

- Consider the training set $T = \{(x_1,y_1),(x_2,y_2),\ldots,(x_\ell,y_\ell)\}$, where $x_i \in \mathbb{R}^n$ are inputs and $y_i \in Y = \{-1,1\}$ are corresponding outputs for $i = 1,2,\ldots,\ell$.
- Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a map from $x_i \in \mathbb{R}^n$ to $y_i \in \{-1,1\}$. Loss function represents the price paid for inaccuracy of precision in classification problem.
- **0-1 Loss Function**: It takes the value zero if the predicted class is same as the true class and one if the predicted class does not match the true class and it is defined by

$$L(x,y,f(x)) = I(y \neq f(x)), \tag{2.3.1}$$

where I is the indicator function.

- **Hinge Loss Function** [13]: The hinge loss function is defined by

$$L_{hinge}(x, y, f(x)) = \max(0, 1 - yf(x)). \tag{2.3.2}$$

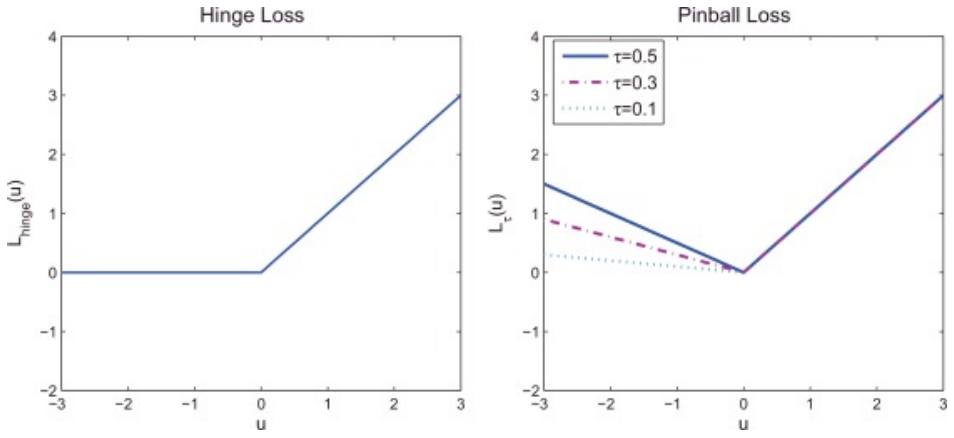

FIGURE 2.1: Hinge and pinball loss funtion.

In all the algorithms discussed in previous chapters, we use hinge loss function. Hinge loss function is noise sensitive. To resolve this problem Suykens et al. [13] use pinball loss function in support vector machines.

- **Pinball Loss Function** [13, 15]: The pinball loss function is defined by

$$L_\tau(x, y, f(x)) = \begin{cases} -yf(x), & -yf(x) \geq 0, \\ -\tau(-yf(x)), & -yf(x) < 0, \end{cases} \tag{2.3.3}$$

where $\tau \in [0, 1]$. Pinball loss gives an additional penalty to the correctly classified points.

2.3.2 Linear Pin-TSVM

Twin parametric-margin support vector machine (TPMSVM) [19] is an efficient classifier but it is noise sensitive. To further improve the generalization performance, Xu

et al. [26] introduced twin support vector machine with pinball loss (Pin-TSVM), especially for noise-corrupted data. Similar to TPMSVM, Pin-TSVM also derives a pair of non-parallel hyperplanes in input space ($\mathbb{R}^n$) given in equation (2.1.1). Formulation of the linear Pin-TSVM is given by [26]:

$$\begin{aligned} \min_{w_+, b_+, \xi_1} \ & \frac{1}{2}\|w_+\|^2 + \frac{\nu_1}{\ell_2} e_2^T (Bw_+ + e_2 b_+) + \frac{c_1}{\ell_1} e_1^T \xi_1 \\ s.t. \quad & (Aw_+ + e_1 b_+) \geq -\xi_1, \ (Aw_+ + e_1 b_+) \leq \frac{\xi_1}{\tau_1} \end{aligned} \tag{2.3.4}$$

and

$$\begin{aligned} \min_{w_-, b_-, \xi_2} \ & \frac{1}{2}\|w_-\|^2 - \frac{\nu_2}{\ell_1} e_1^T (Aw_- + e_1 b_-) + \frac{c_2}{\ell_2} e_2^T \xi_2 \\ s.t. \quad & -(Bw_- + e_2 b_-) \geq -\xi_2, \ -(Bw_- + e_2 b_-) \leq \frac{\xi_2}{\tau_2}, \end{aligned} \tag{2.3.5}$$

where $\nu_1, \ \nu_2 > 0$ are margin parameters and $\tau_1, \ \tau_2 \in [0, 1]$ are pinball loss function parameters. When τ_1 and τ_2 are zero then QPPs (2.3.4) and (2.3.5) are converted into the QPPs of TPMSVM. The Lagrange function corresponding to the problem (2.3.4) is given by:

$$\begin{aligned} L(w_+, \, b_+, \, \xi_1, \, \alpha, \, \beta) = & \frac{1}{2}\|w_+\|^2 + \frac{\nu_1}{\ell_2} e_2^T (Bw_+ + e_2 b_+) + \frac{c_1}{\ell_1} e_1^T \xi_1 \\ & - \alpha^T (Aw_+ + e_1 b_+ + \xi_1) + \beta^T (Aw_+ + e_1 b_+ - \frac{\xi_1}{\tau_1}). \end{aligned} \tag{2.3.6}$$

After using the K.K.T. conditions we obtain:

$$w_+ + \frac{\nu_1}{\ell_2} B^T e_2 - A^T \alpha + A^T \beta = 0, \ \frac{\nu_1}{\ell_2} e_2^T e_2 - e_1^T \alpha + e_1^T \beta = 0 \text{ and } \frac{c_1}{\ell_1} e_1 - \alpha + \frac{\beta}{\tau_1} = 0. \tag{2.3.7}$$

Using equation (2.3.6) and (2.3.7), the dual of (2.3.4) is given by:

$$\begin{aligned} \max_{\alpha, \beta} \ & \frac{\nu_1}{\ell_2} e_2^T B A^T (\alpha - \beta) - \frac{1}{2} (\alpha - \beta)^T A A^T (\alpha - \beta) \\ s.t. \ & e_1^T (\alpha - \beta) = \nu_1, \ \alpha + \frac{\beta}{\tau_1} = \frac{c_1}{\ell_1} e_1, \\ & \alpha \geq 0, \quad \beta \geq 0. \end{aligned} \tag{2.3.8}$$

Value of the bias term (b_+) is given by:

$$O_+ = \{i : \alpha_i > 0 \text{ and } \beta_i > 0\},\ b_+ = -\frac{1}{|O_+|}\sum_{i \in O_+} w_+^T x_i.$$

Similarly, we can obtain the dual of QPP (2.3.5)

$$\begin{aligned} \max_{\gamma,\sigma}\ & \frac{\nu_2}{\ell_1} e_1^T AB^T(\gamma - \sigma) - \frac{1}{2}(\gamma - \sigma)^T BB^T(\gamma - \sigma) \\ s.t.\ & e_2^T(\gamma - \sigma) = \nu_2,\ \gamma + \frac{\sigma}{\tau_2} = \frac{c_2}{\ell_2} e_2, \\ & \gamma \geq 0, \quad \sigma \geq 0, \end{aligned} \tag{2.3.9}$$

where γ and σ are Lagrange multipliers.
Value of bias term (b_-) is given by

$$O_- = \{i : \gamma_i > 0 \text{ and } \sigma_i > 0\},\ b_- = -\frac{1}{|O_-|}\sum_{i \in O_-} w_-^T x_i.$$

A new data point $x \in \mathbb{R}^n$ is assigned to class $i (i = +1, -1)$ according to the equation (2.1.10).

2.3.3 Nonlinear Pin-TSVM

Linear Pin-TSVM can be extended to the nonlinear Pin-TSVM by considering the kernel generated surfaces given in equation (2.1.11). Formulation of the nonlinear Pin-TSVM is given by [26]:

$$\begin{aligned} \min_{u_+, b_+, \xi_1}\ & \frac{1}{2}\|u_+\|^2 + \frac{\nu_1}{\ell_2} e_2^T(K(B,\, D^T)u_+ + e_2 b_+) + \frac{c_1}{\ell_1} e_1^T \xi_1 \\ s.t.\ \quad & (K(A,\, D^T)u_+ + e_1 b_+) \geq -\xi_1, \\ & (K(A,\, D^T)u_+ + e_1 b_+) \leq \frac{\xi_1}{\tau_1} \end{aligned} \tag{2.3.10}$$

and

$$\begin{aligned}\min_{u_-,b_-,\xi_2} \frac{1}{2}\|u_-\|^2 - \frac{\nu_2}{\ell_1}e_1^T(K(A,\, D^T)u_- + e_1b_-) + \frac{c_2}{\ell_2}e_2^T\xi_2 \\ s.t. \quad -(K(B,\, D^T)u_- + e_2b_-) \geq -\xi_2, \\ -(K(B,\, D^T)u_- + e_2b_-) \leq \frac{\xi_2}{\tau_2}. \end{aligned} \tag{2.3.11}$$

Similar to linear Pin-TSVM we can find the dual formulation of (2.3.10) and (2.3.11) by introducing the Lagrange function and using the K.K.T. conditions as follows:

$$\begin{aligned}\max_{\alpha,\beta}\ & \frac{\nu_1}{\ell_2}e_2^TK(B,\, D^T)K(A,\, D^T)^T(\alpha-\beta) - \frac{1}{2}(\alpha-\beta)^TK(A,\, D^T)K(A,\, D^T)^T(\alpha-\beta) \\ s.t.\ & e_1^T(\alpha-\beta) = \nu_1,\ \alpha + \frac{\beta}{\tau_1} = \frac{c_1}{\ell_1}e_1, \\ & \alpha \geq 0, \quad \beta \geq 0, \end{aligned} \tag{2.3.12}$$

$$\begin{aligned}\max_{\gamma,\sigma}\ & \frac{\nu_2}{\ell_1}e_1^TK(A,\, D^T)K(B,\, D^T)^T(\gamma-\sigma) - \frac{1}{2}(\gamma-\sigma)^TK(B,\, D^T)K(B,\, D^T)^T(\gamma-\sigma) \\ s.t.\ & e_2^T(\gamma-\sigma) = \nu_2,\ \gamma + \frac{\sigma}{\tau_2} = \frac{c_2}{\ell_2}e_2, \\ & \gamma \geq 0, \quad \sigma \geq 0. \end{aligned} \tag{2.3.13}$$

Similar to linear Pin-TSVM we can obtain the value b_+ and b_-. A new data point $x \in \mathbb{R}^n$ is assigned to class $i (i = +1, -1)$ by using the equation (2.1.17).

Chapter 3

Multi-class Classifiers

3.1 Multi-class Classification Problems

Multi-class classification problem are more applicable in real life situation. In literature there are two approaches for dealing with multi-class classification problems.

ONE-VERSUS-ONE [12]: If there are k classes then we consider two particular classes as focused classes and classify them by any binary classifier, where it construct $\frac{k(k-1)}{2}$ possible binary classifiers. There only two kinds of samples are involved for each classifier, and no information is given for rest of the samples, therefore we receive unfavorable outputs.

ONE-VERSUS-ALL [12]: In this approach we fix one class and consider rest $(k-1)$ classes as another class which can be executed as binary classification problem giving us k classifiers. This approach leads to the class imbalance problem and produces a bad performance.

3.2 Twin Multi-class Support Vector Machine (Twin-KSVC)

To resolve the drawbacks of above approaches, a new multi-class classification algorithm, called K-SVCR [2], was proposed by Cecilio et al. It produces better forecasting results for multi-class classification problem, as it evaluates all the training points into the "1-vs-1-vs-rest" structure with ternary output $\{-1, 0, 1\}$. By integrating both the structural advantage of K-SVCR and the speed advantage of TSVM, Xu et al. [24] proposed a novel algorithm called Twin-KSVC. It finds two non-parallel planes

$$f^+(x) = w_+^T x + b_+ = 0 \text{ and } f^-(x) = w_-^T x + b_- = 0 \tag{3.2.1}$$

for two focused classes selected from k classes, where $w_+ \in \mathbb{R}^n$, $w_- \in \mathbb{R}^n$, $b_+ \in \mathbb{R}$ and $b_- \in \mathbb{R}$. The rest $(k-2)$ classes are mapped into a region between hyperplanes given in (3.2.1) which satisfies $(w_+^T x + b_+) \geq 1 - \epsilon$ and $(w_-^T x + b_-) \geq 1 - \epsilon$, where ϵ is a positive real number.

3.2.1 Linear Twin-KSVC

Similar to TSVM [14], the Twin-KSVC generates two non-parallel hyperplanes for the two focused classes such that each hyperplane is closer to one of the class and as far as possible from the other. The remaining classes are mapped into a region between hyperplanes which satisfies the following constraints:

$$Cw_+ + e_3 b_+ + \eta_1 \geq e_3(1-\epsilon) \text{ and } Cw_- + e_3 b_- + \eta_2 \geq e_3(1-\epsilon),$$

where ϵ is a positive parameter and non-parallel separating hyperplanes are given by (3.2.1). Let matrix $A \in \mathbb{R}^{\ell_1 \times n}$ represents the data points which belongs to the class $+1$, $B \in \mathbb{R}^{\ell_2 \times n}$ represents the data points which belongs to the class -1 and $C \in \mathbb{R}^{\ell_3 \times n}$ indicates the rest of data points which are labeled 0. Then hyperplanes in equation

(3.2.1) are obtained by following QPPs [24]:

$$\begin{aligned}\min_{w_+, b_+, \xi_1, \eta_1} \quad & \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 + c_2 e_3^T \eta_1 \\ s.t. \quad & -(Bw_+ + e_2 b_+) + \xi_1 \geq e_2, \\ & -(Cw_+ + e_3 b_+) + \eta_1 \geq e_3(1-\epsilon),\ \xi_1 \geq 0,\ \eta_1 \geq 0 \end{aligned} \tag{3.2.2}$$

and

$$\begin{aligned}\min_{w_-, b_-, \xi_2, \eta_2} \quad & \frac{1}{2}\|Bw_- + e_2 b_-\|^2 + c_3 e_1^T \xi_2 + c_4 e_3^T \eta_2 \\ s.t. \quad & (Aw_- + e_1 b_-) + \xi_2 \geq e_1, \\ & (Cw_- + e_3 b_-) + \eta_2 \geq e_3(1-\epsilon),\ \xi_2 \geq 0,\ \eta_2 \geq 0, \end{aligned} \tag{3.2.3}$$

where $c_i (i = 1, 2, 3, 4)$ are positive penalty parameters, $e_i (i = 1, 2, 3)$ are standard unit vectors of appropriate dimensions and ξ_i, $\eta_i (i = 1, 2)$ are slack variables.

The Lagrange function corresponding to the problem (3.2.2) is given by:

$$\begin{aligned} L(w_+,\, b_+,\, \xi_1,\, \eta_1,\, \alpha,\, \beta) =& \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 + c_2 e_3^T \eta_1 \\ & - \alpha^T(-(Bw_+ + e_2 b_+) + \xi_1 - e_2) - \gamma^T \xi_1 \\ & - \beta^T(-(Cw_+ + e_3 b_+) + \eta_1 - e_3(1-\epsilon)) - \sigma^T \eta_1, \end{aligned} \tag{3.2.4}$$

where $\alpha \geq 0$, $\beta \geq 0$, $\gamma \geq 0$ and $\sigma \geq 0$ are Lagrange multipliers. Define $H = [A\ e_1]$, $G = [B\ e_2]$, $T = [C\ e_3]$ and $z_+ = [w_+\ b_+]$.
By using the K.K.T. optimality conditions, the dual formulation of (3.2.2) is given by:

$$\begin{aligned}\max_{\rho} \quad & e_4^T \rho - \frac{1}{2}\rho^T E_1 (H^T H)^{-1} E_1^T \rho \\ s.t. \quad & 0 \leq \rho \leq K_1, \end{aligned} \tag{3.2.5}$$

where $E_1 = [G;\ T];\ K_1 = [c_1 e_2;\ c_2 e_3];\ \rho = [\alpha; \beta]$ and $e_4 = [e_2; e_3(1-\epsilon)]$.
Similarly, the dual of (3.2.3) is given by

$$\begin{aligned} \max_{\zeta}\ & e_5^T\zeta - \frac{1}{2}\zeta^T E_2 (G^T G)^{-1} E_2^T \zeta \\ s.t.\ & 0 \le \zeta \le K_2, \end{aligned} \tag{3.2.6}$$

where $E_2 = [H;\ T];\ K_2 = [c_3 e_1;\ c_4 e_3];\ \zeta = [\gamma; \sigma]$ and $e_5 = [e_1; e_3(1-\epsilon)]$.

3.2.2 Nonlinear Twin-KSVC

Linear Twin-KSVC can be extended to the nonlinear Twin-KSVC by considering the following kernel generated surfaces

$$K(x^T, D_*^T)u_+ + b_+ = 0 \text{ and } K(x^T, D_*^T)u_- + b_- = 0, \tag{3.2.7}$$

where $D_* = [A;\ B;\ C]$; $u_+,\ u_- \in \mathbb{R}^n$ and K is an arbitrary kernel function. They can be obtained by resolving the following pair of QPPs:

$$\begin{aligned} \min_{u_+, b_+, \xi_1, \eta_1}\ & \frac{1}{2}\|K(A, D_*^T)u_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 + c_2 e_3^T \eta_1 \\ s.t.\ & -(K(B, D_*^T)u_+ + e_2 b_+) + \xi_1 \ge e_2, \\ & -(K(C, D_*^T)u_+ + e_3 b_+) + \eta_1 \ge e_3(1-\epsilon), \\ & \xi_1 \ge 0,\ \eta_1 \ge 0 \end{aligned} \tag{3.2.8}$$

and

$$\begin{aligned} \min_{u_-, b_-, \xi_2, \eta_2}\ & \frac{1}{2}\|K(B, D_*^T)u_- + e_2 b_-\|^2 + c_3 e_1^T \xi_2 + c_4 e_3^T \eta_2 \\ s.t.\ & (K(A, D_*^T)u_- + e_1 b_-) + \xi_2 \ge e_1, \\ & (K(C, D_*^T)u_- + e_3 b_-) + \eta_2 \ge e_3(1-\epsilon), \\ & \xi_2 \ge 0,\ \eta_2 \ge 0. \end{aligned} \tag{3.2.9}$$

The Lagrange function corresponding to the problem (3.2.8) is given by:

$$\begin{aligned} L(u_+,\, b_+,\, \xi_1,\, \eta_1,\, \alpha,\, \beta) =& \frac{1}{2}\|K(A, D_*^T)u_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 + c_2 e_3^T \eta_1 \\ & - \alpha^T(-(K(B, D_*^T)u_+ + e_2 b_+) + \xi_1 - e_2) - \gamma^T \xi_1 \\ & - \beta^T(-(K(C, D_*^T)u_+ + e_3 b_+) + \eta_1 - e_3(1-\epsilon)) - \sigma^T \eta_1. \end{aligned} \tag{3.2.10}$$

Define $R = [K(A,\, D_*^T)\; e_1],\; S = [K(B,\, D_*^T)\; e_2]$ and $M = [K(C,\, D_*^T)\; e_3]$. By using the K.K.T. optimality conditions, the dual formulation of (3.2.8) is given by

$$\begin{aligned} \max_{\rho}\;\; & e_4^T \rho - \frac{1}{2}\rho^T N_1 (R^T R)^{-1} N_1^T \rho \\ s.t.\;\; & 0 \le \rho \le K_1, \end{aligned} \tag{3.2.11}$$

where $N_1 = [S;\, M]$. Similarly, the dual formulation of (3.2.9) is given by:

$$\begin{aligned} \max_{\zeta}\;\; & e_5^T \zeta - \frac{1}{2}\zeta^T N_2 (S^T S)^{-1} N_2^T \zeta \\ s.t.\;\; & 0 \le \zeta \le K_2, \end{aligned} \tag{3.2.12}$$

where $N_2 = [R;\, M]$. The solution of QPPs (3.2.11) and (3.2.12) are given by:

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -[R^T R + \delta I]^{-1}[S^T \alpha + M^T \beta] \text{ and } \begin{bmatrix} u_- \\ b_- \end{bmatrix} = [S^T S + \delta I]^{-1}[R^T \gamma + M^T \sigma],$$

where δI, $\delta > 0$ is a regularization term used to avoid the ill-conditioning of matrices $R^T R$ and $S^T S$.

3.2.3 Decision Rule of Twin-KSVC

For a new testing point x, Twin-KSVC determines its class label by the following decision function in linear case [24]:

$$f(x) = \begin{cases} 1, & w_+^T x + e_1 b_+ > -1 + \epsilon \\ -1, & w_-^T x + e_2 b_- < 1 - \epsilon \\ 0, & \text{otherwise.} \end{cases} \tag{3.2.13}$$

In case of nonlinear Twin-KSVC decision function is given by [24]:

$$f(x) = \begin{cases} 1, & K(x^T, D^T) u_+ + e_1 b_+ > -1 + \epsilon \\ -1, & K(x^T, D^T) u_- + e_2 b_- < 1 - \epsilon \\ 0, & \text{otherwise.} \end{cases} \tag{3.2.14}$$

In this way Twin-KSVC constructs $k(k-1)/2$ classifiers for k-classes. For a new point x, a vote is given to one of the focused class based on the condition satisfied by it. Finally, the given point x is assigned to the class that gets highest votes.

3.3 Least Squares Twin Multi-class Classification Support Vector Machine (LST-KSVC)

In LST-KSVC [18] primal QPPs of Twin-KSVC are modified to least squares sense same as in case of LSTSVM [17], inequality constraint replaced with equality constraint and 1-norm of slack variables are replaced with 2-norm. As a result solution of LST-KSVC follows from solving a system of linear equations which makes the algorithm simple and fast.

3.3.1 Linear LST-KSVC

Linear LST-KSVC seeks for two non-parallel hyperplanes given in equation (3.2.1) to classify the focused classes. Formulation of the linear LST-KSVC is given by [18]:

$$\begin{aligned} \min_{w_+, b_+, \xi_1, \eta_1} \quad & \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + \frac{c_1}{2}\|\xi_1\|^2 + \frac{c_2}{2}\|\eta_1\|^2 \\ s.t. \quad & -(Bw_+ + e_2 b_+) + \xi_1 = e_2, \\ & -(Cw_+ + e_3 b_+) + \eta_1 = e_3(1-\epsilon) \end{aligned} \tag{3.3.1}$$

and

$$\begin{aligned} \min_{w_-, b_-, \xi_2, \eta_2} \quad & \frac{1}{2}\|Bw_- + e_2 b_-\|^2 + \frac{c_3}{2}\|\xi_2\|^2 + \frac{c_4}{2}\|\eta_2\|^2 \\ s.t. \quad & (Aw_- + e_1 b_-) + \xi_2 = e_1, \\ & (Cw_- + e_3 b_-) + \eta_2 = e_3(1-\epsilon). \end{aligned} \tag{3.3.2}$$

The square of 2-norm of slack variables ξ_1, ξ_2, η_1 and η_2 with weights $\frac{c_1}{2}$, $\frac{c_2}{2}$, $\frac{c_3}{2}$ and $\frac{c_4}{2}$ instead of 1-norm of ξ_1, ξ_2, η_1 and η_2 with weights c_1, c_2, c_3 and c_4 are used in (3.3.1) and (3.3.2) which makes $\xi_1 \geq 0$, $\eta_1 \geq 0$, $\xi_2 \geq 0$ and $\eta_2 \geq 0$ redundant. By substituting the value of slack variables ξ_1 and η_1 into the objective functions of (3.3.1) we obtain

$$\min_{w_+, b_+} \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + \frac{c_1}{2}\|Bw_+ + e_2 b_+ + e_2\|^2 + \frac{c_2}{2}\|Cw_+ + e_3 b_+ + e_3(1-\epsilon)\|^2. \tag{3.3.3}$$

Using K.K.T. optimality conditions, the solution of (3.3.1) is given by:

$$\begin{bmatrix} w_+ \\ b_+ \end{bmatrix} = -(H^T H + c_1 G^T G + c_2 T^T T)^{-1}(c_1 G^T e_2 + c_2 T^T e_3(1-\epsilon)). \tag{3.3.4}$$

Similarly, the solution of equation (3.3.2) is given by

$$\begin{bmatrix} w_- \\ b_- \end{bmatrix} = (c_3 H^T H + G^T G + c_4 T^T T)^{-1}(c_3 H^T e_1 + c_4 T^T e_3(1-\epsilon)), \tag{3.3.5}$$

where $e_i (i = 1, 2, 3)$ are standard unit vectors of appropriate dimensions. The LST-KSVC solves the classification problem with inverse of two matrices of smaller dimension rather than solving QPPs.

3.3.2 Nonlinear LST-KSVC

Linear LST-KSVC can be extended to the nonlinear LST-KSVC [18] by considering the kernel generated surfaces given in equation (3.2.7). These surfaces can be obtained by solving the following pair of QPPs [18]:

$$\begin{aligned} \min_{u_+, b_+, \xi_1, \eta_1} \ & \frac{1}{2}\|K(A, D_*^T)u_+ + e_1 b_+\|^2 + \frac{c_1}{2}\|\xi_1\|^2 + \frac{c_2}{2}\|\eta_1\|^2 \\ s.t. \quad & -(K(B, D_*^T)u_+ + e_2 b_+) + \xi_1 = e_2, \\ & -(K(C, D_*^T)u_+ + e_3 b_+) + \eta_1 = e_3(1-\epsilon) \end{aligned} \tag{3.3.6}$$

and

$$\begin{aligned} \min_{u_-, b_-, \xi_2, \eta_2} \ & \frac{1}{2}\|K(B, D_*^T)u_- + e_2 b_-\|^2 + \frac{c_3}{2}\|\xi_2\|^2 + \frac{c_4}{2}\|\eta_2\|^2 \\ s.t. \quad & (K(A, D_*^T)u_- + e_1 b_-) + \xi_2 = e_1, \\ & (K(C, D_*^T)u_- + e_3 b_-) + \eta_2 = e_3(1-\epsilon). \end{aligned} \tag{3.3.7}$$

By substituting the value of slack variables into the objective functions of (3.3.6) and (3.3.7) it gives

$$\begin{aligned} \min_{u_+, b_+} \ & \frac{1}{2}\|K(A, D_*^T)u_+ + e_1 b_+\|^2 + \frac{c_1}{2}\|K(B, D_*^T)u_+ + e_2 b_+ \\ & + e_2\|^2 + \frac{c_2}{2}\|K(C, D_*^T)u_+ + e_3 b_+ + e_3(1-\epsilon)\|^2 \end{aligned} \tag{3.3.8}$$

and

$$\begin{aligned} \min_{u_-, b_-} \ & \frac{1}{2}\|K(B, D_*^T)u_- + e_2 b_-\|^2 + \frac{c_3}{2}\|e_1 - K(A, D_*^T)u_- \\ & - e_1 b_-\|^2 + \frac{c_4}{2}\|e_3(1-\epsilon) - K(C, D_*^T)u_- - e_3 b_-\|^2. \end{aligned} \tag{3.3.9}$$

Using the K.K.T. optimality conditions, solution of (3.3.8) and (3.3.9) are given by:

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -\Big(c_1 S^T S + R^T R + c_2 M^T M\Big)^{-1}\Big(c_1 S^T e_2 + c_2 M^T e_3(1-\epsilon)\Big) \tag{3.3.10}$$

and

$$\begin{bmatrix} u_- \\ b_- \end{bmatrix} = \Big(c_3 R^T R + S^T S + c_4 M^T M\Big)^{-1}\Big(c_3 R^T e_2 + c_4 M^T e_3(1-\epsilon)\Big). \tag{3.3.11}$$

The nonlinear LST-KSVC requires inverse of two matrices of size $(\ell+1)\times(\ell+1)$. However using Sherman-Morrison-Woodbury (SMW) formula [10], the nonlinear LST-KSVC can be solved by using the inverse of three matrices of smaller size rather than $(\ell+1)\times(\ell+1)$.
The decision function of LST-KSVC is same as the decision function of Twin-KSVC.

3.4 K-nearest Neighbor-based Weighted Multi-class Twin Support Vector Machine

In Twin-KSVC and LST-KSVC samples have same weights when constructing the hyperplanes. So that local information of samples is omitted and inter-class information is also not exploited. However, they have different effects on the hyperplanes. In K-nearest neighbor-based weighted multi-class twin support vector machine (KWMTSVM) [23] in order to obtain information of intra-class and inter-class, K-nearest neighbor method [5] is employed.

For each sample x_k in class +1, define two sets: $Neb_s(x_k)$ and $Neb_d(x_k)$, where $Neb_s(x_k)$ contains its neighbors in class +1, while $Neb_d(x_k)$ contains its neighbors in class -1.

$$Neb_s(x_k) = \{x_k^j | \text{ if } x_k^j \text{ and } x_k \text{ belong to the same class}, 0 \le j \le m_1\} \tag{3.4.1}$$

and

$$Neb_d(x_k) = \{x_k^j | \text{ if } x_k^j \text{ and } x_k \text{ belong to the different class}, 0 \leq j \leq m_2\}. \quad (3.4.2)$$

Neb_s denotes a set of m_1-nearest neighbors of x_k in class +1, and Neb_d denotes a set of m_2-nearest neighbors x_k in class -1. For two adjacent matrices for class +1, define M_s and M_d as follows [23]:

$$M_{s,ij} = \begin{cases} 1, & \text{if } x_j \in Neb_s(x_i) \text{ or } x_i \in Neb_s(x_j) \\ 0, & \text{otherwise} \end{cases} \quad (3.4.3)$$

and

$$M_{d,ij} = \begin{cases} 1, & \text{if } x_j \in Neb_d(x_i) \text{ or } x_i \in Neb_d(x_j) \\ 0, & \text{otherwise.} \end{cases} \quad (3.4.4)$$

When $M_{s,ij} = 1$ or $M_{d,ij} = 1$, an undirectional edge between two points. For the reduction of the training points, redefine the weight matrix $M_{d,ij}$ as follows:

$$f_j = \begin{cases} 1, & \text{if } \exists\, i,\ M_{d,ij} \neq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (3.4.5)$$

3.4.1 Linear KWMTSVM

Linear KWMTSVM seeks for two non-parallel hyperplanes given in equation (3.2.1) to classify the focused classes. Formulation of the linear KWMTSVM is given by [23]:

$$\begin{aligned} \min_{w_+, b_+, \xi_1, \eta_1} \quad & \frac{1}{2}\|D_1(Aw_+ + e_1 b_+)\|^2 + c_1 e_2^T \xi_1 + c_2 e_3^T \eta_1 \\ s.t. \quad & -F_1(Bw_+ + e_2 b_+) + \xi_1 \geq F_{v_1}, \\ & -H_1(Cw_+ + e_3 b_+) + \eta_1 \geq (1-\epsilon) H_v, \\ & \xi_1 \geq 0,\ \eta_1 \geq 0 \end{aligned} \quad (3.4.6)$$

and

$$\begin{aligned}\min_{w_-,\, b_-,\, \xi_2,\, \eta_2} \quad & \frac{1}{2}\|D_2(Bw_- + e_2b_-)\|^2 + c_3e_1^T\xi_2 + c_4e_3^T\eta_2 \\ s.t. \quad & F_2(Aw_- + e_1b_-) + \xi_2 \geq F_{v_2}, \\ & H_2(Cw_- + e_3b_-) + \eta_2 \geq (1-\epsilon)H_v, \\ & \xi_2 \geq 0,\ \eta_2 \geq 0, \end{aligned} \tag{3.4.7}$$

where $F_1 = diag(f_1, f_2, \ldots, f_{\ell_2})$; $H_1 = diag(h_1, h_2, \ldots, h_{\ell_3})$; $F_2 = diag(f_1, f_2, \ldots, f_{\ell_1})$; $H_2 = diag(h_1, h_2, \ldots, h_{\ell_3})$; $D_1 = diag(d_1, d_2, \ldots, d_{\ell_1})$ and $d_j = \sum_{i=1}^{\ell_1} M_{s,ij}$. F_{v_1} denotes the vector of diagonal elements of F_1, similarly F_{v_2} and H_v are defined.

The Lagrange function corresponding to the problem (3.4.7) is given by :

$$\begin{aligned} L(w_+,\, b_+,\, \xi_1,\, \eta_1,\, \alpha,\, \beta) =& \frac{1}{2}\|D_1(Aw_+ + e_1b_+)\|^2 + c_1e_2^T\xi_1 + c_2e_3^T\eta_1 \\ & - \alpha^T(-F_1(Bw_+ + e_2b_+) + \xi_1 - F_{v_1}) - \gamma^T\xi_1 \\ & - \beta^T(-H_1(Cw_+ + e_3b_+) + \eta_1 - (1-\epsilon)H_v) - \sigma^T\eta_1. \end{aligned} \tag{3.4.8}$$

Using K.K.T. optimality conditions, the dual formulation of (3.4.6) is given by:

$$\begin{aligned}\max_{\rho} \quad & e_4^T\rho - \frac{1}{2}\rho^T L_1^T(H^T D_1 H)^{-1}L_1\rho \\ s.t. \quad & 0 \leq \rho \leq K_1, \end{aligned} \tag{3.4.9}$$

where $\rho = [\alpha; \beta]$, $L_1 = [G^T F_1\ T^T H_1]$ and $e_4 = [F_1^T e_2; (1-\epsilon)H_1^T e_3]$.
Similarly, the dual formulation of (3.4.7) is given by:

$$\begin{aligned}\max_{\zeta} \quad & e_5^T\zeta - \frac{1}{2}\zeta^T L_2^T(G^T D_2 G)^{-1}L_2\zeta \\ s.t. \quad & 0 \leq \zeta \leq K_2, \end{aligned} \tag{3.4.10}$$

where $\zeta = [\gamma;\ \sigma]$, $L_2 = [H^T F_2\ T^T H_2]$ and $e_4 = [F_2^T e_1;\ (1-\epsilon)H_2^T e_3]$.
Solution of QPPs (3.4.9) and (3.4.10) are given by:

$$\begin{bmatrix} w_+ \\ b_+ \end{bmatrix} = -(H^T D_1 H + \delta I)^{-1}(G^T F_1 \alpha + T^T H_1 \beta)$$

and

$$\begin{bmatrix} w_- \\ b_- \end{bmatrix} = (G^T D_2 G + \delta I)^{-1}(H^T F_2 \gamma + T^T H_2 \sigma), \tag{3.4.11}$$

where δI, $\delta > 0$ is a regularization term used to avoid the ill-conditioning of matrices $H^T D_1 H$ and $G^T D_2 G$.

3.4.2 Nonlinear KWMTSVM

Nonlinear KWMTSVM seeks for two kernel generated surfaces given in equation (3.2.7). Formulation of the nonlinear KWMTSVM is given by [23]:

$$\begin{aligned} \min_{u_+, b_+, \xi_1, \eta_1} \quad & \frac{1}{2}\|D_1(K(A, D_*^T)u_+ + e_1 b_+)\|^2 + c_1 e_2^T \xi_1 + c_2 e_3^T \eta_1 \\ s.t. \quad & -F_1(K(B, D_*^T)u_+ + e_2 b_+) + \xi_1 \geq F_{v_1}, \\ & -H_1(K(C, D_*^T)u_+ + e_3 b_+) + \eta_1 \geq (1-\epsilon)H_v, \\ & \xi_1 \geq 0,\ \eta_1 \geq 0 \end{aligned} \tag{3.4.12}$$

and

$$\begin{aligned} \min_{u_-, b_-, \xi_2, \eta_2} \quad & \frac{1}{2}\|(D_2 K(B, D_*^T)u_- + e_2 b_-)\|^2 + c_3 e_1^T \xi_2 + c_4 e_3^T \eta_2 \\ s.t. \quad & F_2(K(A, D_*^T)u_- + e_1 b_-) + \xi_2 \geq F_{v_2}, \\ & H_2(K(C, D_*^T)u_- + e_3 b_-) + \eta_2 \geq (1-\epsilon)H_v, \\ & \xi_2 \geq 0,\ \eta_2 \geq 0. \end{aligned} \tag{3.4.13}$$

By introducing the Lagrange multipliers α, β, γ and σ, the dual formulation of (3.4.12) and (3.4.13) are as follows:

$$\begin{aligned} \max_{\rho} \ & e_4^T\rho - \frac{1}{2}\rho^T M_1^T (R^T D_1 R)^{-1} M_1 \rho \\ s.t. \ & 0 \leq \rho \leq K_1, \end{aligned} \tag{3.4.14}$$

where $M_1 = [S^T F_1 \ M^T H_1]$ and $e_4 = [F_1^T e_2; (1-\epsilon) H_1^T e_3]$.

$$\begin{aligned} \max_{\zeta} \ & e_5^T\zeta - \frac{1}{2}\zeta^T M_2^T (S^T D_2 S)^{-1} M_2 \zeta \\ s.t. \ & 0 \leq \zeta \leq K_2, \end{aligned} \tag{3.4.15}$$

where $M_2 = [R^T F_2 \ M^T H_2]$ and $e_5 = [F_2^T e_1; (1-\epsilon) H_2^T e_3]$.
The decision function of KWMTSVM is same as the decision function of Twin-KSVC.

Chapter 4

Proposed Algorithm 1

4.1 Least Squares K-Nearest Neighbor-based Weighted Multi-class Twin Support Vector Machine

In this chapter, we introduce a novel algorithm termed as least squares K-nearest neighbor-based weighted multi-class twin support vector machine (LS-KWMTSVM). We modify the primal problems of KWMTSVM in least square sense, by replacing the inequality constraints with equality constraints and slack variables are replaced by 2-norm with weights $\frac{c_1}{2}$, $\frac{c_2}{2}$ instead of c_1 and c_2. As a result solution of LS-KWMTSVM follows from solving a system of linear equations which makes the algorithm simple and fast.

4.2 Linear LS-KWMTSVM

Let matrix $A \in \mathbb{R}^{\ell_1 \times n}$ represents the data points of class +1, $B \in \mathbb{R}^{\ell_2 \times n}$ represents the data points of class -1 and $C \in \mathbb{R}^{\ell_3 \times n}$ represent the rest data points which are labeled 0. To classify the focused classes, we are trying to find two non-parallel hyperplanes

defined as follows:

$$f^+(x) = w_+^T x + b_+ = 0 \text{ and } f^-(x) = w_-^T x + b_- = 0, \tag{4.2.1}$$

where w_+, $w_- \in \mathbb{R}^n$ and b_+, $b_- \in \mathbb{R}$. Formulation of the LS-KWMTSVM is given as follows:

$$\begin{aligned} \min_{w_+, b_+ \xi_1, \eta_1} \ & \frac{1}{2}\|D_1(Aw_+ + e_1 b_+)\|^2 + \frac{c_1}{2}\|\xi_1\|^2 + \frac{c_2}{2}\|\eta_1\|^2 \\ s.t. \quad & -F_1(Bw_+ + e_2 b_+) + \xi_1 = F_{v_1}, \\ & -H_1(Cw_+ + e_3 b_+) + \eta_1 = (1-\epsilon)H_v \end{aligned} \tag{4.2.2}$$

and

$$\begin{aligned} \min_{w_-, b_- \xi_2, \eta_2} \ & \frac{1}{2}\|D_2(Bw_- + e_2 b_-)\|^2 + \frac{c_3}{2}\|\xi_2\|^2 + \frac{c_4}{2}\|\eta_2\|^2 \\ s.t. \quad & F_2(Aw_- + e_1 b_-) + \xi_2 = F_{v_2}, \\ & H_2(Cw_- + e_3 b_-) + \eta_2 = (1-\epsilon)H_v, \end{aligned} \tag{4.2.3}$$

where D_1, D_2, H_1, H_2, F_1, F_2, F_{v_2}, F_{v_1} and H_v are same as defined in KWMTSVM. ξ_1, η_1, ξ_2, η_2 are slack variables. By substituting the value of ξ_1 and η_1 into the objective function of QPP (4.2.2) we obtain:

$$\begin{aligned} \min_{w_+, b_+} \ & \frac{1}{2}\|D_1(Aw_+ + e_1 b_+)\|^2 + \frac{c_1}{2}\|F_{v_1} + F_1(Bw_+ + e_2 b_+)\|^2 \\ & + \frac{c_2}{2}\|(1-\epsilon)H_v + H_1(Cw_+ + e_3 b_+)\|^2. \end{aligned} \tag{4.2.4}$$

Using the K.K.T. optimality conditions [16] we obtain:

$$\begin{aligned} & A^T D_1^T D_1(Aw_+ + e_1 b_+) + c_1 B^T F_1^T(F_{v_1} + F_1(Bw_+ + e_2 b_+)) \\ & + c_2 C^T H_1^T((1-\epsilon)H_v + H_1(Cw_+ + e_3 b_+)) = 0 \end{aligned} \tag{4.2.5}$$

and

$$e_1^T D_1^T D_1(Aw_+ + e_1 b_+) + c_1 e_2^T F_1^T (F_{v_1} + F_1(Bw_+ + e_2 b_+)) + c_2 e_3^T H_1^T ((1-\epsilon)H_v + H_1(Cw_+ + e_3 b_+)) = 0. \tag{4.2.6}$$

Arranging equation (4.2.5) and (4.2.6) in matrix form we get

$$\begin{bmatrix} A^T \\ e_1^T \end{bmatrix} D_1^T D_1 [A \ e_1] \begin{bmatrix} w_+ \\ b_+ \end{bmatrix} + c_1 \begin{bmatrix} B^T \\ e_2^T \end{bmatrix} F_1^T \Big(F_{v_1} + F_1 [B \ e_2] \begin{bmatrix} w_+ \\ b_+ \end{bmatrix} \Big) + c_2 \begin{bmatrix} C^T \\ e_3^T \end{bmatrix} H_1^T \Big(H_v(1-\epsilon) + H_1 [C \ e_3] \begin{bmatrix} w_+ \\ b_+ \end{bmatrix} \Big) = 0 \tag{4.2.7}$$

$$H^T D_1^T D_1 H z_+ + c_1 G^T F_1^T [F_{v_1} + F_1 G z_+] + c_2 T^T H_1^T [H_v(1-\epsilon) + H_1 T z_+] = 0,$$

where $H = [A \ e_1]$, $G = [B \ e_2]$, $T = [C \ e_3]$ and $z_+ = \begin{bmatrix} w_+ \\ b_+ \end{bmatrix}$.

$$z_+ = -(H^T D_1^T D_1 H + c_1 G^T F_1^T F_1 G + c_2 T^T H_1^T H_1 T)^{-1} (c_1 G^T F_1^T F_{v_1} + c_2 T^T H_1^T H_v(1-\epsilon)). \tag{4.2.8}$$

Similarly, the solution of (4.2.3) is given by:

$$\begin{bmatrix} w_- \\ b_- \end{bmatrix} = (G^T D_2^T D_2 G + c_3 H^T F_2^T F_2 H + c_4 T^T H_1^T H_1 T)^{-1} (c_3 H^T F_2^T F_{v_2} + c_4 T^T H_1^T H_v(1-\epsilon)). \tag{4.2.9}$$

In this regard, optimal hyperplanes of (4.2.1) are obtained. LS-KWMTSVM solves the classification problem with just inverse of two matrices of smaller size rather than solving two QPPs in KWMTSVM.

4.3 Nonlinear LS-KWMTSVM

We can extend the linear LS-KWMTSVM to the nonlinear LS-KWMTSVM by considering the following kernel generated surfaces:

$$K(x^T, D_*^T)u_+ + b_+ = 0 \text{ and } K(x^T, D_*^T)u_- + b_- = 0, \tag{4.3.1}$$

where $D_* = [A;\ B;\ C]$; $u_+,\ u_- \in \mathbb{R}^n$ and K is an arbitrary kernel function. Formulation of the nonlinear LS-KWMTSVM is given by:

$$\begin{aligned} \min_{u_+, b_+, \xi_1, \eta_1} \ & \frac{1}{2}\|D_1(K(A, D_*^T)u_+ + e_1 b_+)\|^2 + \frac{c_1}{2}\|\xi_1\|^2 + \frac{c_2}{2}\|\eta_1\|^2 \\ s.t. \quad & -F_1(K(B, D_*^T)u_+ + e_2 b_+) + \xi_1 = F_{v_1}, \\ & -H_1(K(C, D_*^T)u_+ + e_3 b_+) + \eta_1 = (1-\epsilon)H_v \end{aligned} \tag{4.3.2}$$

and

$$\begin{aligned} \min_{u_-, b_-, \xi_2, \eta_2} \ & \frac{1}{2}\|D_2(K(B, D_*^T)u_- + e_2 b_-)\|^2 + \frac{c_3}{2}\|\xi_2\|^2 + \frac{c_4}{2}\|\eta_2\|^2 \\ s.t. \quad & F_2(K(A, D_*^T)u_- + e_1 b_-) + \xi_1 = F_{v_2}, \\ & H_1(K(C, D_*^T)u_- + e_3 b_-) + \eta_1 = (1-\epsilon)H_v. \end{aligned} \tag{4.3.3}$$

Similar to linear case, we obtain the solution of (4.3.2) and (4.3.3) as follows:

$$\begin{aligned} \begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = & -(R^T D_1^T D_1 R + c_1 S^T F_1^T F_1 S + c_2 M^T H_1^T H_1 M)^{-1} \\ & (c_1 S^T F_1^T F_{v_1} + c_2 M^T H_1^T H_v(1-\epsilon)) \end{aligned} \tag{4.3.4}$$

and

$$\begin{aligned} \begin{bmatrix} u_- \\ b_- \end{bmatrix} = & (S^T D_2^T D_2 S + c_3 R^T F_2^T F_2 R + c_4 M^T H_1^T H_1 M)^{-1} \\ & (c_3 R^T F_2^T F_{v_2} + c_4 M^T H_1^T H_v(1-\epsilon)), \end{aligned} \tag{4.3.5}$$

where $R = [K(A, D_*^T)\ e_1]$, $S = [K(B, D_*^T)\ e_2]$, $M = [K(C, D_*^T)\ e_3]$ and ϵ is a real positive parameter. We notice that for the solution of nonlinear case we require inverse of two matrices of size $(\ell + 1) \times (\ell + 1)$. Now to reduce the computation cost, we use *Sherman-Morrison-Woodbury* (SMW) formula [10] to recast (4.3.4) and (4.3.5)

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -\Big(Z - Z(F_1S)^T\Big(\frac{I}{c_1} + (F_1S)^T Z(F_1S)^T\Big)^{-1} F_1SZ\Big) \Big(c_1S^TF_1F_{v_1} + c_2M^TH_1^TH_v(1-\epsilon)\Big), \tag{4.3.6}$$

$$\begin{bmatrix} u_- \\ b_- \end{bmatrix} = \Big(F_* - F_*(F_2R)^T\Big(\frac{I}{c_3} + (F_2R)^T F_*(F_2R)^T\Big)^{-1} F_2RF_*\Big) \Big(c_1R^TF_2F_{v_2} + c_2M^TH_1^TH_v(1-\epsilon)\Big), \tag{4.3.7}$$

where $Z = (R^TD_1^TD_1R + c_2M^TH_1^TH_1M)^{-1}$ and $F_* = (S^TD_2^TD_2S + c_4M^TH_1^TH_1M)^{-1}$. Again by using the SMW formula we obtain:

$$Z = \frac{1}{c_2}\Big(Y - Y(F_2R)^T(c_2I + F_2RY(F_2R)^T)^{-1}F_2RY\Big), \tag{4.3.8}$$

$$F_* = \frac{1}{c_4}\Big(Y - Y(F_1S)^T(c_4I + F_1SY(F_1S)^T)^{-1}F_1SY\Big), \tag{4.3.9}$$

where $Y = (M^TH_1^TH_1M)^{-1}$. To avoid the case when Y is ill-conditioned we add a regularization term δI, where $\delta > 0$ then

$$Y = \frac{1}{\delta}(I - (H_1M)^T(\delta I + M^TH_1^TH_1M)^{-1}H_1M). \tag{4.3.10}$$

Advantage of SMW formula is that earlier we have to compute the inverse of matrix with size $(\ell + 1) \times (\ell + 1)$ and after using SMW we can compute it by using inverse of three smaller dimension matrices of size $(\ell_1 \times \ell_1)$, $(\ell_2 \times \ell_2)$ and $(\ell_3 \times \ell_3)$ respectively.

4.4 Decision Function

For a new testing point x, we decide the class label by using the following decision function in the linear case:

$$f(x) = \begin{cases} 1, & w_+^T x + e_1 b_+ > -1 + \epsilon \\ -1, & w_-^T x + e_2 b_- < 1 - \epsilon \\ 0, & \text{otherwise.} \end{cases} \tag{4.4.1}$$

In case of nonlinear, decision function is given by:

$$f(x) = \begin{cases} 1, & K(x, D^T) u_+ + e_1 b_+ > -1 + \epsilon \\ -1, & K(x, D^T) u_- + e_2 b_- < 1 - \epsilon \\ 0, & \text{otherwise.} \end{cases} \tag{4.4.2}$$

In our proposed algorithm we construct $k(k-1)/2$ classifiers for k-classes. For a new data point x, a vote is given to one of the focused class based on condition it satisfies. Finally, the given data point x is assigned to the class that gets highest votes.

4.5 Algorithm Analysis

Our proposed algorithm seeks for two non-parallel hyperplanes by solving a system of linear equations.

- We give different weights to the data points of the focused class +1 in (4.2.2) by using K-nearest neighbor (KNN) graph [25]. If a data point in the focused class +1 has more KNNs, then we give more weight to it.
- **Constraint Reduction**: We introduce $F_1 = diag(f_1, f_2, \ldots, f_{l_2})$ where $f_i = 1$ or 0 in (4.2.2). Similarly, $H_1 = diag(h_1, h_2, \ldots, h_{l_3})$ where $h_i = 1$ or 0. If any

data point of focused classes belongs to the KNN of another focused class i.e., $f_i = 1$ or $h_i = 1$ otherwise the corresponding constraint is redundant.

- Our proposed algorithm exploits the local information of intra-class and inter-class by using the K-nearest neighbor method and imbalance problem is removed by using the "1-versus-1-versus -rest" approach.

- Our proposed algorithm is an extension of LST-KSVC [18]. In equation (4.2.2) if all components of $M_{s,ij} = 1$, $f_i = 1$ and $h_i = 1 \ \forall \ i, j$ then it reduces to LST-KSVC.

- **Computation Complexity**: It is well known that computational complexity of SVM is $\mathcal{O}(\ell^3)$ [4] where ℓ is the number of data points. Computation complexity of TSVM is $\mathcal{O}(2 \times (\frac{\ell}{2})^3)$ because TSVM divides the data into roughly equal size matrices of order $(\frac{\ell}{2} \times n)$.

  In a 3-class classification problem, assume each class have approximately $\ell/3$ data points. The data points of 3^{rd} class involved twice in the constraints of the K-SVCR, thus there are $\frac{4\ell}{3}$ constraints. Therefore, computational complexity of K-SVCR is $\mathcal{O}(\frac{4\ell}{3})^3$. In Twin-KSVC, the data points of 3^{rd} class are used only once hence computational complexity of Twin-KSVC is $\mathcal{O}(2 \times (\frac{2\ell}{3})^3)$.

  If LS-KWMTSVM have no redundant constraint then LS-KWMTSVM has almost same construction as Twin-KSVC, so they have same computational complexity. If LS-KWMTSVM have some redundant constraint then computational complexity of LS-KWMTSVM is less than $\mathcal{O}(\frac{16\ell^3}{27})$. In LS-KWMTSVM, KNN-graph needs $\ell^2(\log(\ell))$ steps to compute the weight matrices for each data point. Thus, total computational complexity is approximately $\mathcal{O}(\frac{16\ell^3}{27} + \ell^2(\log(\ell))$.

4.6 Numerical Experiments

In this section, we demonstrate the performance of the four algorithms i.e., KWMTSVM [25], LST-KSVC [18], Twin K-SVC [24] and our proposed algorithm. We conduct experiments on ten benchmark datasets taken from UCI machine learning repository [3]

and KEEL repository. The datasets are iris, teaching evaluation, wine, hayes-roth, glass, lenses, contraceptive, zoo, cleave land and tae. In Table 4.1, total number of samples, attributes and number of classes are denoted by sign " $\cdot \times \cdot \times \cdot$ " below the dataset name. For example Iris dataset contains 150 samples and each sample consist of the four attribute classified to three classes is denoted by "$150 \times 4 \times 3$". We test the performance of the proposed algorithm in classification accuracy and running time aspects. In our experiments, we use 10-fold cross-validation to compare the performance of the four algorithms. In 10-fold cross validation the dataset randomly splits into ten subsets, nine of them are used for training and one is used for testing. This process is repeated ten times and performance measure is taken as the average of ten tested results. All the algorithm are implemented by MATLAB R2010b on Windows 10 Education on a PC with system configuration Intel (R) Core (TM) i7-6700 CPU @ 3.40 GHZ with 8 GB of RAM. Gaussian kernel function $K(x, y) = \exp^{-(\|x-y\|^2/\mu^2)}$ is considered on benchmark datasets, as it is often applied and yields great generalization performance, where μ is a parameter.

4.6.1 Parameter Selection

It is clear that the performance of the algorithms depend on the choices of parameters. In our experiments, optimal parameters are obtained by the grid search method [12]. For all the algorithms, penalty parameters $c_i (i = 1, 2, 3, 4)$ are selected from the set $\{2^j \,|\, j = -5, -4, \cdots, 4, 5\}$. The Gaussian kernel parameter μ is selected over the range $\{2^j \,|\, j = -10, -9, \cdots, 9, 10\}$. Parameter ϵ is set to a small value 0.2. To reduce the computational cost of parameter selection, we set $c_1 = c_3$ and $c_2 = c_4$ for all the algorithms.

4.6.2 Results Comparison and Discussion

We compare our proposed algorithm with Twin-KSVC [24], LST-KSVC [18] and KWMTSVM [25]. The experimental results are given in Table 4.1. In the perspective of prediction accuracy, we find out that our proposed algorithm LS-KWMTSVM

outperforms on most of the datasets. In our algorithm, we solve the system of linear equation, which makes the computation speed fast. LST-KSVC also solves system of linear equations and even faster than proposed algorithm. Figure 4.1 shows the influence of penalty parameters (c_1, c_2) on the performance of LS-KWMTSVM with the optimal value of kernel parameter for Wine dataset. It is observed from Figure 4.1 that c_1 have more impact on the predictive accuracy of proposed algorithm as compared to c_2. As value of c_1 increase accuracy also increase linearly. Figure 4.2 shows the influence of penalty parameters (c_1, c_2) on the performance of LS-KWMTSVM with the optimal value of kernel parameter for Teaching dataset. It is observed from Figure 4.2 that variation in the value of c_1 and c_2 does not much effect the variation of predictive accuracy. Predicative accuracy of proposed algorithm is approximately constant. Figure 4.3 shows the influence of penalty parameters (c_1, c_2) on the performance of LS-KWMTSVM with the optimal value of kernel parameter for Iris dataset. It is observed from Figure 4.3 that c_1 have more impact on the predictive accuracy as compared to c_2. For large value of c_1, the performance of LS-KWMTSVM suddenly degrades.

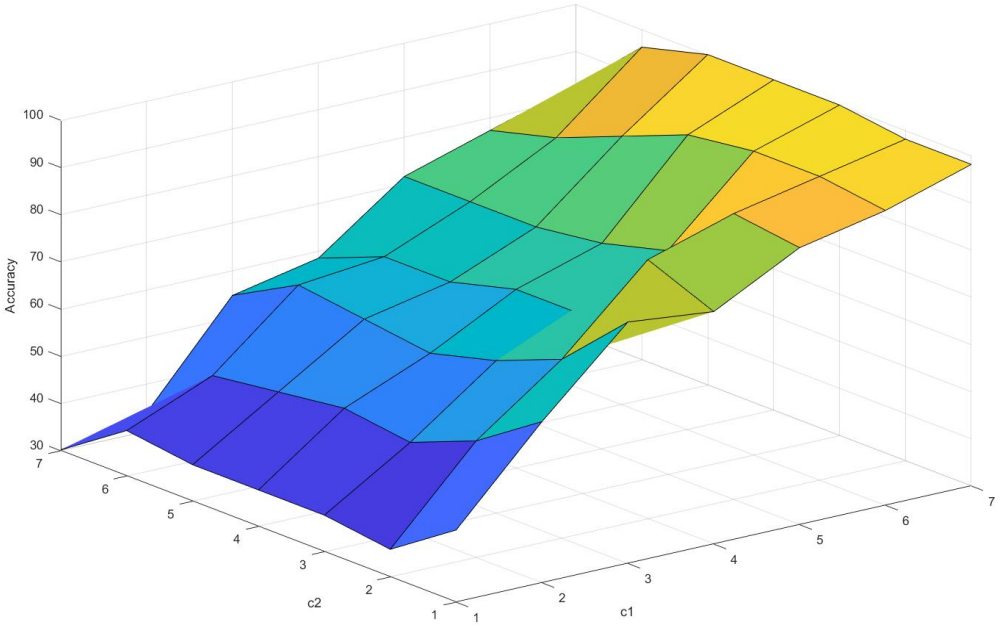

FIGURE 4.1: Wine

TABLE 4.1: Performance comparsion of multi-class algorithm with Gaussian kernel

Dataset	Twin-KSVC		LST-KSVC		KWMTSVM		Proposed LS-KWMTSVM	
	Accuracy (%) Parameters (c_1=c_3, c_2=c_4, μ)	Time (s)	Accuracy (%) Parameters (c_1=c_3, c_2=c_4, μ)	Time (s)	Accuracy (%) Parameters (c_1=c_3, c_2=c_4, μ)	Time (s)	Accuracy (%) Parameters (c_1=c_3, c_2=c_4, μ)	Time (s)
Iris (150×4×3)	**88.89** (0.031,2,8)	0.31	86.66 (8,0.031,0.5)	0.08	82.22 (0.125,0.5,2,0.2)	0.33	**88.89** (8,0.031,0.25)	0.130
Teaching (151×5×3)	43.47 (0.125,0.31,8)	0.30	**60.86** (32,8,0.062)	0.08	54.34 (0.5,0.125,0.125)	0.34	58.69 (8,2,0.0625)	0.132
Wine (178×13×3)	92.45 (0.031,8,1024)	0.34	**94.33** (32,1,64)	0.09	92.45 (2,0.031,1024)	0.36	93.33 (32,1,128)	0.157
Hayes (132×5×3)	66.67 (0.031,0.5,2)	0.30	**76.19** (32,0.031,1)	0.07	66.67 (0.125,0.031,4)	0.31	**76.19** (32,0.031,1)	0.11
Glass (214×9×6)	57.97 (0.031,1,16)	0.95	60 (8,0.125,0.0625)	0.45	59.42 (2,0.031,8)	1.14	**65.21** (32,0.125,0.125)	0.67
Lense (24×4×3)	75 (0.031,0.031,16)	0.26	75 (32,0.031,8)	0.04	75 (0.031,0.031,16)	0.28	**87.5** (8,8,4)	0.07
Contraceptive (210×9×3)	**40** (32,32,1024)	0.41	32.30 (2,2,2)	0.115	35 (1,0.031,4)	0.43	**40** (2,0.031,1)	0.22
Zoo (100×16×7)	61.29 (0.031,0.031,2)	0.66	93.54 (0.031,0.5,2)	0.19	61.29 (32,32,1024)	0.71	**96.77** (1,0.125,0.7)	0.37
Cleveland (297×13×5)	52.87 (0.125,0.125,256)	1.11	50.57 (32,0.031,2)	0.46	50.57 (0.5,0.031,256)	1.22	**55.17** (8,0.031,8)	0.72
Tae (150×5×3)	44.44 (0.5,0.5,0.5)	0.32	48.89 (32,0.031,0.004)	0.08	53.33 (0.125,8,0.125)	0.35	**57.77** (8,0.125,0.0078)	0.14

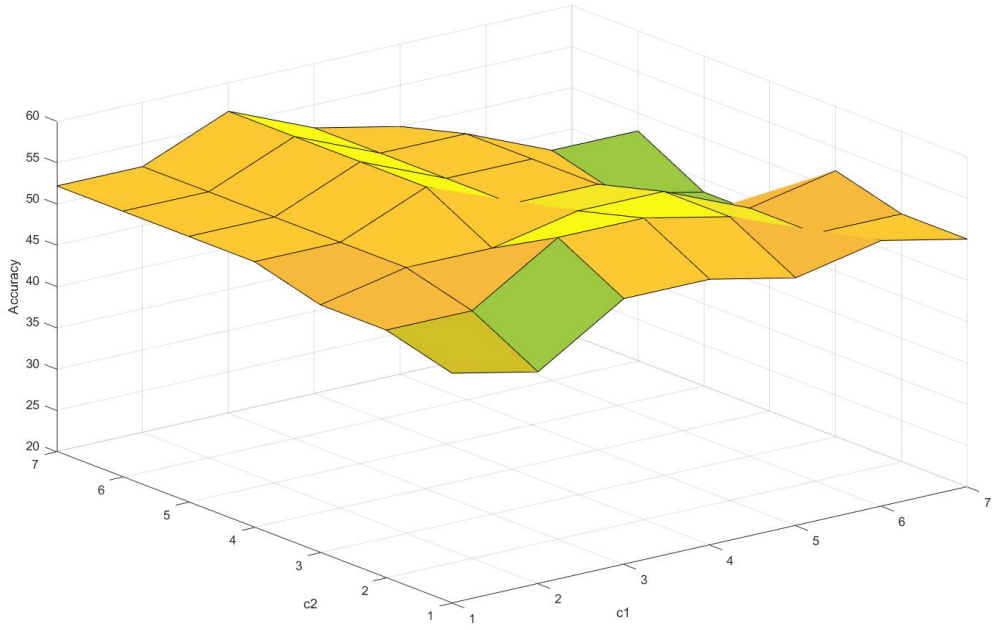

FIGURE 4.2: Teaching

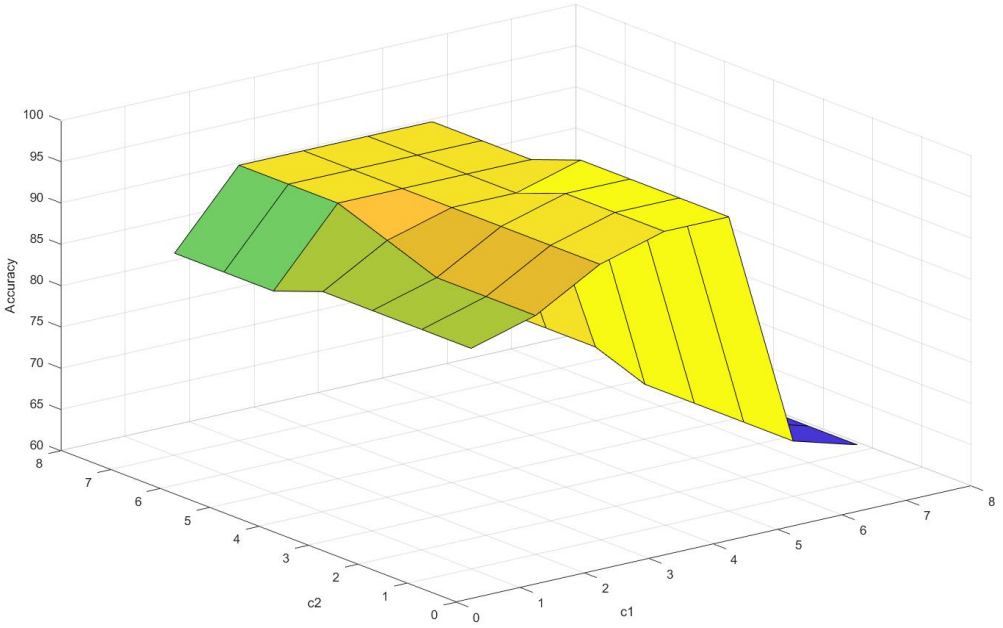

FIGURE 4.3: Iris

4.7 Statistical Analysis

In Table 4.1 we observe that our algorithm does not outperform for all the datasets. To analyze the statistical significance of our proposed algorithm LS-KWMTSVM in comparison of Twin-KSVC [24], LST-KSVC [18], and KWMTSVM [23]. We use Friedman test [6, 9] with corresponding post hoc tests. Friedman test is considered to be simple, robust, non-parametric and safe test for comparison of different classifiers over multiple datasets. It ranks the algorithm for each dataset separately, the best performing algorithm getting the rank 1, second one rank 2 and so on. In the case of ties average ranks are assigned. The average ranks of all algorithms on the accuracy with Gaussian kernel function are computed and listed in Table 4.2.

TABLE 4.2: Average rank on accuracy of four algorithms on ten benchmark datasets

Dataset	Twin-KSVC	LSTKSVC	KWMTSVM	LSKWMTSVM
Iris	1.5	3	4	1.5
Teaching	4	1	3	2
Wine	3.5	1	3.5	2
Hayes	3.5	1.5	3.5	1.5
Glass	4	2	3	1
Lenses	3	3	3	1
Contraceptive	1.5	4	3	1.5
Zoo	3.5	2	3.5	1
Cleveland	2	3.5	3.5	1
Tae	4	3	2	1
Average Rank	3.05	2.4	3.2	1.35

Under the null hypothesis, the Friedman statistics is distributed according to $\mathcal{X}_F^2$ with $(k-1)$ degree of freedom as follows [6]:

$$\mathcal{X}_F^2 = \frac{12N}{k(k+1)}\Big[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\Big] \text{ and } F_F = \frac{(N-1)\mathcal{X}_F^2}{N(k-1)-\mathcal{X}_F^2}. \tag{4.7.1}$$

$$\mathcal{X}_F^2 = \frac{12\times 10}{4(4+1)}\Big[3.05^2 + 2.4^2 + 3.2^2 + 1.35^2 - \frac{4\times 5^2}{4}\Big] = 12.75. \tag{4.7.2}$$

$$F_F = \frac{(10-1) \times 12.75}{10 \times (4-1) - 12.75} = 6.65, \tag{4.7.3}$$

where $R_j = \frac{1}{N} \sum_j r_i^j$ and r_i^j denotes the rank of j^{th} algorithm on the i^{th} dataset out of N datasets and F_F is F-distribution with degree of freedom $(k-1)(N-1)$, where k is the number of algorithms and N number of datasets. The critical values of F(3, 27) at significance level (α = 0.025, 0.05, 0.1) are 3.65, 2.96, 2.29 respectively. Since F_F value 6.65 of our algorithm is larger than the critical values i.e., the average rank of our proposed algorithm is much lower than other algorithms. One can conclude that our proposed LS-KWMTSVM is significantly better than Twin-KSVC, LST-KSVC and KWMTSVM.

Chapter 5

Proposed Algorithm 2

5.1 General Twin Support Vector Machine With Pinball Loss (Pin-GTSVM)

Usual TSVM affiliates hinge loss function which is sensitive to noise and unstable for re-sampling. To elevate the performance of TSVM [14] similar to Pin-TSVM [26] we also introduce pinball loss [13] in usual TSVM and propose a novel algorithm termed as general twin support vector machine with pinball loss. Pin-GTSVM deals with quantile distance [15] which makes it less sensitive to noise.

5.2 Linear Pin-GTSVM

The linear Pin-GTSVM obtains two non-parallel hyperplanes

$$f^+(x) = w_+^T x + b_+ = 0 \text{ and } f^-(x) = w_-^T x + b_- = 0, \tag{5.1.1}$$

where w_+, $w_- \in \mathbb{R}^n$ and b_+, $b_- \in \mathbb{R}$. Introducing the pinball loss into the TSVM, we obtain following QPPs:

$$\min_{w_+, b_+} \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + c_1 \sum_{j=1}^{\ell_2} L_{\tau_2}(x_j^-, y_j, f^+(x_j^-)) \tag{5.2.1}$$

and

$$\min_{w_-, b_-} \frac{1}{2}\|Bw_- + e_2 b_-\|^2 + c_2 \sum_{i=1}^{\ell_1} L_{\tau_1}(x_i^+, y_i, f^-(x_i^+)), \tag{5.2.2}$$

where ℓ_1 and ℓ_2 denotes the number of data points in positive and negative class respectively. Data point x_j^- corresponds to the negative class and x_i^+ corresponds to the positive class respectively. $L_{\tau_1}(\cdot)$ and $L_{\tau_2}(\cdot)$ are pinball loss functions with parameter τ_1, $\tau_2 \in [0,\, 1]$ respectively.

Substituting the pinball loss in (5.2.1) and (5.2.2) we obtain the following QPPs:

$$\begin{aligned} \min_{w_+, b_+, \xi_1} & \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 \\ s.t. \quad & -(Bw_+ + e_2 b_+) + \xi_1 \geq e_2, \\ & -(Bw_+ + e_2 b_+) - \frac{\xi_1}{\tau_2} \leq e_2 \end{aligned} \tag{5.2.3}$$

and

$$\begin{aligned} \min_{w_-, b_-, \xi_2} & \frac{1}{2}\|Bw_- + e_2 b_-\|^2 + c_2 e_1^T \xi_2 \\ s.t. \quad & (Aw_- + e_1 b_-) + \xi_2 \geq e_1, \\ & (Aw_- + e_1 b_-) - \frac{\xi_2}{\tau_1} \leq e_1, \end{aligned} \tag{5.2.4}$$

where c_1, c_2 are positive penalty parameters and e_1, e_2 are standard unit vectors of appropriate dimensions and ξ_1, ξ_2 are slack variables.

Remark: We observe that when τ_1 and τ_2 tends to zero then QPPs (5.2.3) and (5.2.4) are reduced to QPPs of TSVM.

To obtain the solution of (5.2.3), we introduce the corresponding Lagrange function with Lagrange multipliers $\alpha \geq 0$ and $\beta \geq 0$

$$L(w_+,\, b_+,\, \xi_1,\, \alpha,\, \beta) = \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 - \alpha^T(-(Bw_+ + e_2 b_+) + \xi_1 - e_2) + \beta^T(-(Bw_+ + e_1 b_+) - \frac{\xi_1}{\tau_2} - e_2). \tag{5.2.5}$$

Using K.K.T. optimality condition [16] we obtain:

$$A^T(Aw_+ + e_1 b_+) + B^T\alpha - B^T\beta = 0 \tag{5.2.6}$$

$$e_1^T(Aw_+ + e_1 b_+) + e_2^T\alpha - e_2\beta = 0 \tag{5.2.7}$$

$$c_1 e_2 - \alpha - \frac{\beta}{\tau_2} = 0. \tag{5.2.8}$$

By using equation (5.2.8) and $\alpha \geq 0$, we obtain $-\tau_2 c_1 e_2 \leq (\alpha - \beta)$.
Combining equation (5.2.6) and (5.2.7) leads to

$$\begin{bmatrix} A^T \\ e_1^T \end{bmatrix} [A\ e_1] \begin{bmatrix} w_+ \\ b_+ \end{bmatrix} + \begin{bmatrix} B^T \\ e_2^T \end{bmatrix} (\alpha - \beta) = 0. \tag{5.2.9}$$

Define $H = [A\ e_1],\ G = [B\ e_2]$ and $z_+ = \begin{bmatrix} w_+ \\ b_+ \end{bmatrix}$. With these notations (5.2.9) can be rewritten as follows :

$$H^T H z_+ + G^T(\alpha - \beta) = 0,\ \ i.e.,\ \ z_+ = -(H^T H)^{-1} G^T(\alpha - \beta). \tag{5.2.10}$$

Although $H^T H$ is always positive semi-definite, it is possible that it may not be well conditioned in some situations. So, we introduce regularization term [21] δI, $\delta > 0$, to take care of problems due to possible ill-conditioning of $H^T H$. Here, I is an identity

matrix of appropriate dimensions. Therefore, equation (5.2.10)

$$z_+ = -(H^T H + \delta I)^{-1} G^T (\alpha - \beta). \tag{5.2.11}$$

Using (5.2.5) and above K.K.T. conditions, we get the dual of (5.2.3) as follows:

$$\begin{aligned} \max_{(\alpha-\beta)} \; & e_2^T(\alpha-\beta) - \frac{1}{2}(\alpha-\beta)^T G(H^T H)^{-1} G^T(\alpha-\beta) \\ s.t. \quad & -\tau_2 c_1 e_2 \leq (\alpha-\beta). \end{aligned} \tag{5.2.12}$$

Similarly, we can obtain the dual of QPP (5.2.4) as follows:

$$\begin{aligned} \max_{(\gamma-\sigma)} \; & e_1^T(\gamma-\sigma) - \frac{1}{2}(\gamma-\sigma)^T H(G^T G)^{-1} H^T(\gamma-\sigma) \\ s.t. \quad & (\gamma-\sigma) \geq -\tau_1 c_2 e_1, \end{aligned} \tag{5.2.13}$$

where $\gamma \geq 0$ and $\sigma \geq 0$ are Lagrange multipliers. Finally, optimal separating hyperplanes are obtained by:

$$\begin{bmatrix} w_+ \\ b_+ \end{bmatrix} = -(H^T H + \delta I)^{-1} G^T(\alpha - \beta) \text{ and } \begin{bmatrix} w_- \\ b_- \end{bmatrix} = (G^T G + \delta I)^{-1} H^T(\gamma - \sigma).$$

A new data point $x \in \mathbb{R}^n$ is assigned to class $i (i = +1, -1)$ depending on which of the two hyperplanes in (5.2.1) is closer to x, i.e.,

$$class(i) = \text{sign}\Big(\frac{{w_+}^T x + b_+}{\|w_+\|} + \frac{{w_-}^T x + b_-}{\|w_-\|}\Big). \tag{5.2.14}$$

5.3 Nonlinear Pin-GTSVM

In order to extend the linear Pin-GTSVM to the nonlinear Pin-GTSVM we consider the following kernel generated surfaces:

$$K(x^T, D^T)u_+ + b_+ = 0 \text{ and } K(x^T, D^T)u_- + b_- = 0, \tag{5.3.1}$$

where $D = [A\,;\, B]$; $u_+,\ u_- \in \mathbb{R}^n$ and K is an arbitrary kernel function. Similar to linear Pin-GTSVM we construct the following optimization problems:

$$\begin{aligned} \min_{u_+, b_+, \xi_1} \ & \frac{1}{2}\|K(A,\, D^T)u_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 \\ s.t. \quad & -(K(B,\, D^T)u_+ + e_2 b_+) + \xi_1 \geq e_2, \\ & -(K(B,\, D^T)u_+ + e_2 b_+) - \frac{\xi_1}{\tau_2} \leq e_2 \end{aligned} \tag{5.3.2}$$

and

$$\begin{aligned} \min_{u_-, b_-, \xi_2} \ & \frac{1}{2}\|K(B,\, D^T)u_+ + e_2 b_+\|^2 + c_2 e_1^T \xi_2 \\ s.t. \quad & (K(A,\, D^T)u_- + e_1 b_-) + \xi_2 \geq e_1, \\ & (K(A,\, D^T)u_- + e_1 b_-) - \frac{\xi_2}{\tau_1} \leq e_1, \end{aligned} \tag{5.3.3}$$

where $\xi_1,\ \xi_2$ are slack vectors. $e_i (i = 1,\, 2)$ are standard unit vectors of appropriate dimensions. By introducing the Lagrange function and applying the K.K.T. optimality conditions, the dual of (5.3.2) is given by:

$$\begin{aligned} \max_{(\alpha-\beta)} \ & e_2^T(\alpha - \beta) - \frac{1}{2}(\alpha - \beta)^T Q (P^T P)^{-1} Q^T (\alpha - \beta) \\ s.t. \quad & -\tau_2 c_1 e_2 \leq (\alpha - \beta). \end{aligned} \tag{5.3.4}$$

Similarly, we can obtain the dual of equation (5.3.3)

$$\begin{aligned} \max_{(\gamma-\sigma)} \ & e_1^T(\gamma - \sigma) - \frac{1}{2}(\gamma - \sigma)^T P (Q^T Q)^{-1} P^T (\gamma - \sigma) \\ s.t. \quad & (\gamma - \sigma) \geq -\tau_1 c_2 e_1, \end{aligned} \tag{5.3.5}$$

where $P = [K(A,\, D^T)\ e_1]$ and $Q = [K(B,\, D^T)\ e_2]$. $\alpha,\ \beta,\ \gamma,$ and $\sigma \geq 0$ are Lagrange multipliers. Finally, optimal separating hyperplanes are given by:

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -(P^T P + \delta I)^{-1} Q^T (\alpha - \beta) \text{ and } \begin{bmatrix} u_- \\ b_- \end{bmatrix} = (Q^T Q + \delta I)^{-1} P^T (\gamma - \sigma). \tag{5.3.6}$$

It is possible that P^TP and Q^TQ may not be well conditioned in some situations. So, we introduce a regularization term [21] δI, $\delta > 0$, to take care of problems due to possible ill-conditioning of P^TP and Q^TQ. Here, I is an identity matrix of appropriate dimensions. A new point $x \in \mathbb{R}^n$ is assigned to class $i(i = +1, -1)$ depending on which of the two kernel generated surface in (5.3.1) is closer to x, i.e.

$$class(i) = \text{sign}\Big(\frac{K(x^T, D^T)u_+ + b_+}{\|u_+\|} + \frac{K(x^T, D^T)u_- + b_-}{\|u_-\|}\Big), \tag{5.3.7}$$

5.4 Algorithm Analysis

Similar to TSVM our proposed algorithm Pin-GTSVM explores two non-parallel hyperplanes by solving a pair of smaller sizes QPPs. Twin parametric-margin support vector machine (TPMSVM) is a improved version of TSVM in which Peng et al. [19] determine the parametric-margin hyperplanes similar to par-ν-SVM [11]. Xu et al. [26] introduce pinball loss function in TPMSVM and propose a algorithm Pin-TSVM. We ventilate the pinball loss function in classical TSVM which is more general than Pin-TSVM [26].

It is well known that computational complexity of SVM is $\mathcal{O}(\ell^3)$ [4] where ℓ is the number of data points. Computation complexity of TSVM is $\mathcal{O}(2 \times (\frac{\ell}{2})^3)$ because TSVM divides the data into roughly equal size matrices of order $(\frac{\ell}{2} \times n)$. Computational complexity of Pin-GTSVM and Pin-TSVM is same as of TSVM. The major advantage of our proposed algorithm is that Pin-GTSVM is noise insensitive, especially for the feature noise around the decision boundary. Pin-GTSVM is more stable than TSVM for re-sampling. The re-sampling stability and noise insensitive are sustained by numerical experiments. In term of computation time, Pin-GTSVM costs nearly the same time as TSVM i.e., pinball loss function does not increase the computation time of Pin-GTSVM. From Table 5.1 we observe that as we increase the value of pinball loss parameter $\tau_i (i = 1, 2) \in [0,\ 1]$ it gives more better results.

Table 5.1: Performance comparison of binary-class algorithms with Gaussian kernel

Dataset	TSVM		LSTSVM		Pin-GTSVM($\tau = 0.5$)		Pin-GTSVM($\tau = 0.8$)	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
Fertility (r = 0) (100×9×2)	**96.66**	0.15	**96.66**	0.009	**96.66**	0.152	**96.66**	0.149
r=0.05	**96.66**	0.149	**96.66**	0.0094	**96.66**	0.151	**96.66**	0.148
r=0.1	**96.66**	0.155	**96.66**	0.0108	**96.66**	0.151	**96.66**	0.154
Banknote (r = 0) (1372×5×2)	**42.47**	0.26	**42.47**	0.188	41.26	0.216	41.74	0.21
r=0.05	**42.47**	0.247	**42.47**	0.189	42.23	0.23	**42.47**	0.21
r=0.1	**42.47**	0.243	42.33	0.19	**42.47**	0.212	**42.47**	0.25
Votes (r = 0) (435×16×2)	96.89	0.022	95.34	0.010	96.89	0.020	**97.67**	0.021
r=0.05	96.12	0.025	**96.8**	0.0108	**96.8**	0.02	**96.8**	0.026
r=0.1	96.89	0.0227	95.34	0.010	**97.69**	0.021	96.12	0.026
NDC1100 (r = 0) (1110×33×2)	95.75	0.12	96.67	0.097	**96.96**	0.110	96.36	0.11
r=0.05	95.15	0.22	**96.06**	0.10	**96.06**	0.113	**96.06**	0.115
r=0.1	93.33	0.217	95.75	0.091	94.84	0.115	**96.06**	0.121
Transfusion (r = 0) (748×4×3)	85.51	0.0736	89.86	0.03	**90.54**	0.07	86.48	0.074
r=0.05	85.13	0.119	**88.51**	0.0038	**88.51**	0.068	82.43	0.072
r=0.1	87.83	0.118	78.37	0.0035	87.16	0.074	**88.51**	0.078
WDBC (r = 0) (569×30×2)	95.65	0.052	**97.1**	0.028	94.2	0.045	95.65	0.047
r=0.05	**95.65**	0.06	91.3	0.036	92.75	0.049	**95.65**	0.045
r=0.1	91.3	0.00723	91.3	0.045	94.2	0.05	**97.10**	0.064

Dataset	TSVM		LSTSVM		Pin-GTSVM($\tau = 0.5$)		Pin-GTSVM($\tau = 0.8$)	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
Splice (r = 0) (767×60×2)	88.37	0.072	**88.89**	0.058	**88.89**	0.06	**88.89**	0.066
r=0.05	88.37	0.068	87.92	0.026	87.73	0.064	**88.74**	0.068
r=0.1	88	0.049	**88.71**	0.032	87.43	0.070	**88.71**	0.07
Monk2 (r = 0) (601×8×3)	83.79	0.011	**85.41**	0.003	81.94	0.011	84.72	0.11
r=0.05	83.79	0.011	72.45	0.0037	83.33	0.01	**84.02**	0.0118
r=0.1	78.47	0.009	**88.49**	0.004	82.4	0.009	81.94	0.008
Monk1 (r = 0) (556×8×2)	78.7	0.0089	**79.16**	0.004	74.10	0.0091	74.53	0.0089
r=0.05	78.7	0.0092	75.92	0.0042	75.23	0.0097	**79.16**	0.0096
r=0.1	75	0.0098	**82.14**	0.0044	79.86	0.0094	78.70	0.0095
CMC (r = 0) (1473×9×2)	**74.63**	0.2	**74.63**	0.16	**74.63**	0.20	**74.63**	0.21
r=0.05	74.2	0.26	**74.63**	0.151	**74.63**	0.221	**74.63**	0.218
r=0.1	74.2	0.242	73.78	0.142	74.2	0.21	**74.63**	0.221
Crossplane150 (r = 0) (150×2×2)	74.28	0.072	62.85	0.0012	80	0.007	**81.42**	0.006
r=0.05	**98.57**	0.066	94.28	0.0012	95.71	0.007	**76.28**	0.0068
r=0.1	95.71	0.0072	95.71	0.0014	94.28	0.0073	94.28	0.0071
NDC550 (r = 0) (550×33×2)	90	0.468	**94**	0.029	**94**	0.042	**94**	0.046
r=0.05	90	0.045	**98**	0.031	94	0.044	96	0.043
r=0.1	90	0.052	**94**	0.032	**94**	0.045	**94**	0.047

5.5 Numerical Experiments

In this section, we exhibit the performance of three algorithms i.e, TSVM [14], LSTSVM [17] and our proposed Pin-GTSVM. In our proposed algorithm Pin-GTSVM we perform experiments for different values of $\tau_i (i = 1, 2) = 0.5, 0.8$ and investigate the variation in accuracy. We conduct experiments on twelve benchmark datasets taken from UCI machine learning repository [3] and KEEL repository. The datasets are Fertility, Iono, Banknote, Votes, and so on which are given in Table 5.1. In Table 5.1, total number of samples, attributes and number of classes are denoted by sign " $\cdot \times \cdot \times \cdot$ " below the dataset name. For example Fertility dataset contains 100 samples and each sample consists of the nine features classified in two classes, it is denoted by "$100 \times 9 \times 2$". We test the performance of the proposed algorithm in classification accuracy and running time aspects.

In our experiments, we normalize the datasets before training and testing. We incorporate the feature noise [13] in datasets with zero-mean Gaussian noise. For each feature standard deviation is denoted by r. The value of r is fixed $r = 0$(i.e, noise free), 0.05 and 0.1. The testing and training sets are aggravated by the same noise.

In our experiments, we use 10-fold cross-validation to compare the performance of the three algorithms. In 10-fold cross-validation the dataset randomly splits into ten subsets, nine of them are used for training and one is used for testing. This process is repeated ten times and performance measure is taken as the average of ten tested results. All the algorithm are implemented by MATLAB R2010b on Windows 10 Education on a PC with system configuration Intel (R) Core (TM) i7-6700 CPU @ 3.40 GHZ with 8 GB of RAM. Gaussian kernel function $K(x, y) = \exp^{-(\|x-y\|^2/\mu^2)}$ is considered on benchmark datasets, as it is often applied and yields great generalization performance, where μ is a parameter.

5.5.1 Parameter Selection

It is clear that the performance of different algorithms depends on the choices of parameters. In our experiments optimal value of parameters are found by the grid search

method [12]. In all the algorithms we have to choose three parameters i.e, penalty parameter c_1 and c_2, Gaussian kernel parameter μ. We use the fix value of τ_1, τ_2 0, 0.5 and 0.8. The optimal value for c_1 and c_2 parameter are selected from the set $\{2^j \mid j = -5, -4, \cdots, 4, 5\}$ and μ is selected over the range $\{2^j \mid j = -10, -9, \cdots, 9, 10\}$. To reduce the computation cost of parameter selection, we set $\tau_1 = \tau_2$.

5.5.2 Results Comparison and Discussion

We compare our proposed algorithm Pin-GTSVM with TSVM [14], LSTSVM [17]. The experimental results are presented in Table 5.1. From the perspective of prediction accuracy, we find out that our proposed algorithm Pin-GTSVM outperforms on 8 data sets out of 12. Pin-GTSVM attains best prediction accuracy in 24, 15 cases out of 36, when $\tau_i = 0.8$ and 0.5 respectively. Pin-GTSVM gives better results for noise corrupted datasets due to use of pinball loss function. From the perspective of time, we observed that Pin-GTSVM cost nearly the same computation time as TSVM. LSTSVM solves the system of linear equation, so it is much faster than TSVM and Pin-GTSVM.

5.6 Noise Insensitivity

The main advantage of pinball loss minimization enjoys insensitivity with respect to noise around the optimal separating hyperplanes. For easy comprehension, we focus on the linear case. Consider the pinball loss function as follows:

$$L_\tau(x, y, f(x)) = \begin{cases} -yf(x), & -yf(x) \geq 0, \\ -\tau(-yf(x)), & -yf(x) < 0, \end{cases} \tag{5.6.1}$$

where $\tau \in [0, 1]$. Sub-gradient of the pinball loss $L_\tau(x, y, f(x))$ is given by sign function $sgn_\tau(x, y, f(x))$ given below:

$$sgn_\tau(x, y, f(x)) = \begin{cases} 1, & -yf(x) > 0, \\ [-\tau, 1], & -yf(x) = 0, \\ -\tau, & -yf(x) < 0. \end{cases} \tag{5.6.2}$$

Using the K.K.T. optimality condition for QPP (5.2.1) we obtain:

$$A^T Aw_+ + A^T e_1 b_+ + c_1 e_1 \sum_{j=1}^{\ell_2} sgn_{\tau_2}(x_j^-,\, y_j,\, f^+(x_j^-))x_j^- = 0. \tag{5.6.3}$$

For given w_+ and b_+ the whole index set can be separated into three distinct subsets as follows:

$$S_1 = \{j : w_+^T x_j + b_+ < 0\},$$

$$S_2 = \{j : w_+^T x_j + b_+ > 0\},$$

$$S_3 = \{j : w_+^T x_j + b_+ = 0\}.$$

Using the notation S_1, S_2 and S_3, equation (5.6.3) can be recast as follows:

$$A^T Aw_+ + A^T e_1 b_+ - c_1 e_1 \tau_2 \sum_{j \in S_1} x_j^- + c_1 e_1 \sum_{j \in S_2} x_j^- + c_1 e_1 \sum_{j \in S_3} \zeta_j^+ x_j^- = 0,$$

where $j = 1, 2, \cdots, \ell_2$ and $\zeta_j^+ \in [-\tau_2, 1]$. We perceive that τ_2 controls the numbers of points in S_1, S_2, and S_3. When τ_2 is small enough, there are a lot of points in set S_2 and hence the amount of points in S_1 and S_2 will decrease and when τ_2 becomes large, all of the three sets contain many points, and hence the result is less sensitive.

Chapter 6

Conclusion and Future Work

A novel multi-class algorithm, i.e., least squares K-nearest neighbor-based weighted multi-class twin support vector machine (LS-KWMTSVM), is proposed in chapter 4 of this thesis. In this algorithm local information of samples is exploited. For the information of intra-class we introduce different weights D_1 and D_2 in the objective function of QPPs and for inter-class information weight vectors F_v and H_v are involved in the constraint. If any component of $F_v = 0$ or $H_v = 0$ is zero, it implies that the corresponding constraint is redundant so it can be ignored. LS-KWMTSVM solves a system of linear equations which make it simple and fast so that we can use it to classify the large datasets in shorter time. Experimental results on ten UCI datasets authenticates that our proposed algorithm LS-KWMTSVM outperforms existing algorithms on most of the datasets.

We also proposed another algorithm, general twin support vector machine with pinball loss. Pinball loss function is widely used in regression problems since there is a strong relation between quantile regression and pinball loss function. Here we use pinball loss function in usual TSVM instead of hinge loss. Some properties like noise insensitivity is investigated from both theoretical and experimental aspects. In addition, we compare our proposed algorithm with TSVM and LSTSVM. We also investigate the effect of value $\tau_i (i = 1, 2)$ in our algorithm. Pin-GTSVM is more stable for noise corrupted data than usual TSVM, it is sustained by numerical experiments.

In future, we will inspect the techniques of data compressing and re-sampling so that our Pin-GTSVM can be applied to large-scale noise corrupted datasets for classification. Further study on this topic will also include many application of Pin-GTSVM in real life classification with noise. Another interesting topic would be to design fast algorithm for our Pin-GTSVM and introduce pinball loss function in other variants of TSVM. It should be identify that there are several parameters in our proposed algorithms, so parameter selection is a interesting problem and we will need to address in future. The selection of an appropriate kernel function is very important for performance improvement.

Bibliography

[1] Abe S. (2010), Support Vector Machines for Pattern Classification, second ed., Springer London Dordrecht Heidelberg, New York.

[2] Angulo C., Parra X., Catal A. (2003), K-SVCR: A support vector machine for multi-class classification, Neurocomputing, vol. 55, pp. 57-77.

[3] Blake C.L., Merz C.J. (1998), UCI Repository for Machine Learning Databases, Dept. of Information and Computer Sciences, Univ. of California, Irvine, http://www.ics.uci.edu/ mlearn/MLRepository.html.

[4] Cortes C., Vapnik V. (1995), Support vector networks, Machine Learning, vol. 20, pp. 273-297.

[5] Cover T.M., Hart P.E. (1967), Nearest neighbor pattern classification, IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27.

[6] Demšar J. (2006), Statistical comparisons of classifiers over multiple datasets, Journal of Machine Learning Research, vol. 7, pp. 1–30.

[7] Deng N., Tian Y., Zhang C. (2013), Support Vector Machines Optimization Based Theory, Algorithms, and Extensions, CRC Press Taylor & Francis Group.

[8] Fukunaga K. (1990), Introduction to Statistical Pattern Recognition, Second Edition. Academic Press, San Diego.

[9] García S., Fernández A., Luengo J., Herrera F. (2010), Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining experimental analysis of power, Information Sciences, vol. 180, pp. 2044–2064.

[10] Golub G.H., Van C.F. (2012), Matrix Computations, 3rd ed., John Hopkins University Press, Baltimore, London.

[11] Hao P.Y. (2010), New support vector algorithms with parametric insensitive/margin model, Neural Networks, vol. 23, no. 1, pp. 60-73.

[12] Hsu C., Lin C. (2002), A comparison of methods for multiclass support vector machine, IEEE Transactions on Neural Networks, vol. 13, pp. 415–425.

[13] Huang X., Shi L., Suykens J.A.K. (2014), Support vector machine classifier with pinball loss, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 5, pp. 984-997.

[14] Jayadeva, Khemchandani R., Chandra S. (2007), Twin support vector machine for pattern classification, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 5, pp. 905-910.

[15] Koenker R. (2005), Quantile Regression, Cambridge, U.K.: Cambridge University Press.

[16] Kuhn, H.W., Tucker, A.W. (1951), Nonlinear programming, Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press, pp. 481–492.

[17] Kumar M.A., Gopal M. (2009), Least squares twin support vector machines for pattern classification, Expert Systems with Applications, vol. 36, 7535-7543.

[18] Nasiri J.A., Charkari N.M., Jalili S. (2015), Least squares twin multi-class classification support vector machine, Pattern Recognition, vol 48 pp. 984-992.

[19] Peng X. (2011), A novel twin parametric-margin support vector machine for pattern recognition, Pattern Recognition, vol. 44, nos. 10-11, pp. 2678-2692.

[20] Steinwart I., Christmann A. (2007), How SVMs can estimate quantiles and the median, Eletronic Proceedings of Neural Information Processing Systems, pp. 305–312.

[21] Tikhonov A.N., Arsenin V.Y. (1977), Solution of Ill Posed Problems., John Wiley and Sons.

[22] Vapnik V.N. (1998), Statistical Learning Theory. John Wiley & Sons, New York.

[23] Xu Y. (2016), K-nearest neighbor-based weighted multi-class twin support vector machine, Neurocomputing, vol. 205, pp. 430-438.

[24] Xu Y., Guo R., Wang L. (2013), A twin multi-class classification support vector machine, Cognitive Computation, vol. 5, pp. 580-588.

[25] Xu Y., Wang L. (2014), K-nearest neighbor-based weighted twin support vector regression, Applied Intelligence, vol. 41, pp. 299-309.

[26] Xu Y., Yang Z., Pan X. (2017), A novel twin support vector machine with pinball loss, IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 2, pp. 359-370.