Journal of Computer Science and Cybernetics, V.37, N.1 (2021), 43–56
DOI 10.15625/1813-9663/37/1/15396

WEIGHTED STRUCTURAL SUPPORT VECTOR MACHINE

NGUYEN THE CUONG[1], HUYNH THE PHUNG[2,*]

[1] *Faculty of Basic, Telecommunications University, 101 Mai Xuan Thuong Street, Vinh Hoa Ward, Nha Trang City, Khanh Hoa Province, Viet Nam*
[2] *Department of Mathematics, College of Sciences, Hue University, 77 Nguyen Hue Street, Hue City, Thua Thien - Hue Province, Viet Nam*

Abstract. In binary classification problems, two classes of data seem to be different from each other. It is expected to be more complicated due to the clusters in each class also tend to be different. Traditional algorithms as Support Vector Machine (SVM) or Twin Support Vector Machine (TWSVM) cannot sufficiently exploit structural information with cluster granularity of the data, cause limitation on the capability of simulation of data trends. Structural Twin Support Vector Machine (S-TWSVM) sufficiently exploits structural information with cluster granularity for learning a represented hyperplane. Therefore, the capability of S-TWSVM's data simulation is better than that of TWSVM. However, for the datasets where each class consists of clusters of different trends, the S-TWSVM's data simulation capability seems restricted. Besides, the training time of S-TWSVM has not been improved compared to TWSVM. This paper proposes a new Weighted Structural - Support Vector Machine (called WS-SVM) for binary classification problems with a class-vs-clusters strategy. Experimental results show that WS-SVM could describe the tendency of the distribution of cluster information. Furthermore, both the theory and experiment show that the training time of the WS-SVM for classification problem has significantly improved compared to S-TWSVM.

Keywords. Support vector machine, twin support vector machine, structural twin support vector machine, weighted structural - support vector machine.

1. INTRODUCTION

In the early years of the 20th century, the Support Vector Machine (SVM) [1, 2] was a popular binary classification algorithm applied to many different fields in practice [3, 4, 5, 6, 7]. The SVM seeks a hyperplane separating two classes so that the margin between them is largest. Actual data is often distributed with different structures and tendencies, but SVM does not fully exploit structural information of data, so its ability to simulate data is limited.

Nowadays, with rapid development, datasets are increasing in number and diversifying in structure. This fact requires classification algorithms to guarantee accuracy and improve the speed and ability to simulate data distribution. Many variants of SVM have been recently proposed to improve the speed and other task of standard SVM [4, 8, 9, 7, 10]. Two typical innovations of SVM are Twin Support Vector Machine (TWSVM) [11] and Structural Twin Support Vector Machine (S-TWSVM) [12]. The main idea of TWSVM is to seek two hyperplanes such that each hyperplane is closer to one class and far away from the other by solving two Quadratic Programming Problems (QPPs) whose size are smaller than the QPP

[*] Corresponding author.

E-mail addresses: nckcbnckcb@gmail.com (N.T.Cuong); huynhthephung@gmail.com (H.T.Phung).

in SVM. Despite having to solve two QPPs, the speed of TWSVM is approximately four times faster than standard SVM. S-TWSVM has the same strategy as TWSVM. Besides, S-TWSVM fully exploits structural information with cluster granularity into learning the model to build a more reasonable classifier. There is a reason for handling SVM with structure. In some practical binary classification problems, each class will consist of more than one cluster. For example, we consider the problem of classifying fruits with data including five categories: Mango, Jackfruit, Pineapple, Apples, Grapes, but the fruits will be only be classified according to the criteria "smooth skin" or "rough skin". Obviously, data in the "smooth skin" class will form 3 clusters, corresponding to Pineapple, Apples, and Grapes, while data in the "rough skin" class will distribute into 2 clusters, corresponding to Jackfruit and Pineapple. S-TWSVM proved to be quite effective in the simple case, where each class consists of clusters with a similar distribution trend. However, for more complex data types, where each class contains clusters of different trends, S-TWSVM was not efficient at simulating data trends.

Based on the strategy of TWSVM and S-TWSVM, we propose a new binary classification model: Weighted Structural - Support Vector Machine (called WS-SVM) with a class-vs-clusters strategy. Instead of solving two QPPs as in S-TWSVM, WS-SVM will solve $(l+k)$ QPPs, where k and l are the number of clusters in class $\{+\}$ and class $\{-\}$, respectively. This method allows WS-SVM to effectively simulate the distribution trends for complex data types while also improving computation speed.

The paper is organized as follows. Section 2 briefly introduces the background of SVM, TWSVM and S-TWSVM; Section 3 is devoted to a detailed description of WS-SVM along with the algorithms and discussions; All experimental results are presented in Section 4, together with the comparative evaluation; The conclusion is given in Section 5. All algorithms are settled by version 3.8.3 of Python Programming Language.

2. BACKGROUND

2.1. Structural granularity

Consider a binary classification problem with the dataset, denoted by matrix C, consisting of m points (each point is a row of C) $\mathbf{x}_j^T \in \mathbb{R}^n,\ j = 1, \dots,\ m$. We also write $\mathbf{x}_j \in C$ to indicate that $\mathbf{x}_j$ is a row of C. Suppose that $y_j \in \{-1, 1\}$ is the $j-$th data point label. Class $\{+\}$ consists of m_A points denoted by a matrix $A \subset \mathbb{R}^{m_A \times n}$, class $\{-\}$ consists of m_B points denoted by a matrix $B \subset \mathbb{R}^{m_B \times n}$. There are k clusters in class $\{+\}$, whose $i-$th cluster consists of m_{Ai} points and is denoted by matrix $A_i \subset \mathbb{R}^{m_{Ai} \times n}$. Also, there are l clusters in class $\{-\}$, whose $i-$th cluster consists of m_{Bi} points and is denoted by matrix $B_i \subset \mathbb{R}^{m_{Bi} \times n}$. A, B, A_i, B_i are called structural granularity [13]. We are interested in the following quantities of structural granularity.

- Class granularity: $\varSigma_A = \dfrac{1}{m_A} \displaystyle\sum_{\mathbf{x}_j \in A} (\mathbf{x}_j - \boldsymbol{\mu}_A)(\mathbf{x}_j - \boldsymbol{\mu}_A)^T,$

$$\varSigma_B = \frac{1}{m_B} \sum_{\mathbf{x}_j \in B} (\mathbf{x}_j - \boldsymbol{\mu}_B)(\mathbf{x}_j - \boldsymbol{\mu}_B)^T.$$

- Cluster granularity: $\Sigma_{i+} = \dfrac{1}{m_{Ai}} \sum_{\mathbf{x}_j \in A_i} (\mathbf{x}_j - \boldsymbol{\mu}_{Ai})(\mathbf{x}_j - \boldsymbol{\mu}_{Ai})^T,$

$$\Sigma_{i-} = \frac{1}{m_{Bi}} \sum_{\mathbf{x}_j \in B_i} (\mathbf{x}_j - \boldsymbol{\mu}_{Bi})(\mathbf{x}_j - \boldsymbol{\mu}_{Bi})^T,$$

here, $\boldsymbol{\mu}_X$ denotes the average vector of the dataset X.

2.2. Support vector machine

Standard SVM [2] seeks a hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ separating class $\{+\}$ and class $\{-\}$ such that the margin $\frac{2}{\|\mathbf{w}\|}$ between two classes is largest. However, Standard SVM is only available when the data is linearly separable. In the case when the data is not linearly separable, soft SVM [2] is recommended with the more loser constraints

$$\begin{cases} \min\limits_{\mathbf{w},b,\boldsymbol{\xi}} & c\mathbf{e}^T\boldsymbol{\xi} + \frac{1}{2}\|\mathbf{w}\|_2^2, \\ \text{s.t.} & D(C\mathbf{w} + \mathbf{e}b) + \boldsymbol{\xi} \geq \mathbf{e}, \ \boldsymbol{\xi} \geq \mathbf{0}, \end{cases} \tag{1}$$

where $D \in \mathbb{R}^{m\times m}$ is the diagonal matrix with $D_{jj} = y_j$; $\forall j$, $\boldsymbol{\xi} \in \mathbb{R}^m$ is the vector of slack variables, $c \in \mathbb{R}$ is the penalty coefficient, appropriately selected to adjust the role between terms in the objective function, $\mathbf{e} \in \mathbb{R}^m$ is the vector of ones. A new data point $\mathbf{x}$ will be classified in class $\{+\}$ if $\text{sgn}(f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b) > 0$ and in class $\{-\}$ if $\text{sgn}(f(\mathbf{x})) < 0$. SVM does not effectively exploit structural information of the data, resulting in simulating the distribution structure of the two classes being the same (see Figure 1.a).

2.3. Twin support vector machine

Based on the strategy of multi-surface proximal support vector classification via generalized eigenvalues (GEPSVM) [14]. The main idea of TWSVM [11] for binary classification problem is to seek two hyperplanes:

- $f_+(\mathbf{x}) = \mathbf{w}_+^T\mathbf{x} + b_+ = 0$ is closer to class $\{+\}$ and far away from class $\{-\}$,
- $f_-(\mathbf{x}) = \mathbf{w}_-^T\mathbf{x} + b_- = 0$ is closer to class $\{-\}$ and far away from class $\{+\}$

by solving two QPPs as follows

$$\text{(TWSVM1)} \begin{cases} \min\limits_{\mathbf{w}_+,b_+,\boldsymbol{\xi}} & \frac{1}{2}||A\mathbf{w}_+ + \mathbf{e}_+b_+||_2^2 + c_1\mathbf{e}_-^T\boldsymbol{\xi}, \\ \text{s.t.} & -(B\mathbf{w}_+ + \mathbf{e}_-b_+) + \boldsymbol{\xi} \geq \mathbf{e}_-, \ \boldsymbol{\xi} \geq \mathbf{0}, \end{cases} \tag{2}$$

$$\text{(TWSVM2)} \begin{cases} \min\limits_{\mathbf{w}_-,b_-,\boldsymbol{\eta}} & \frac{1}{2}||B\mathbf{w}_- + \mathbf{e}_-b_-||_2^2 + c_2\mathbf{e}_+^T\boldsymbol{\eta}, \\ \text{s.t.} & (A\mathbf{w}_+ + \mathbf{e}_+b_+) + \boldsymbol{\eta} \geq \mathbf{e}_+, \ \boldsymbol{\eta} \geq \mathbf{0}, \end{cases} \tag{3}$$

where c_1, c_2 are penalty coefficients to adjust the role between terms in the objective functions, $\mathbf{e}_+ \in \mathbb{R}^{m_A}$, $\mathbf{e}_- \in \mathbb{R}^{m_B}$ are vectors of ones, $\boldsymbol{\xi} \in \mathbb{R}^{m_B}$, $\boldsymbol{\eta} \in \mathbb{R}^{m_A}$ are vectors of slack variables. A new data $\mathbf{x}$ is classified into class $\{+\}$ or class $\{-\}$ depending on whether it is closer to the hyperplane $f_+(\mathbf{x}) = 0$ or the hyperplane $f_-(\mathbf{x}) = 0$. TWSVM simulates the structural distribution of the two classes independently, but it does not actually simulate the distribution trend of the data within each class (see Figure 1.b). It has been shown in [11] that the training time of TWSVM is approximately four times faster than that of standard SVM (see Table 1).

2.4. Structural twin support vector machine

S-TWSVM [12] has two steps: The first step is to extract the structural information within classes by Ward's linkage clustering method [15, 13]; The second step is the model learning. Suppose that there are k clusters $A_1, \dots, A_k$ in class A, each cluster A_i consists of m_{Ai} data points, and there are l clusters $B_1, \dots, B_l$ in class B, each cluster B_i consists of m_{Bi} data points. S-TWSVM determines two hyperplanes

$$f_+(\mathbf{x}) = \mathbf{w}_+^T\mathbf{x} + b_+ = 0;\ f_-(\mathbf{x}) = \mathbf{w}_-^T\mathbf{x} + b_- = 0, \tag{4}$$

by solving two QPPs as follows

$$\begin{cases} \min\limits_{\mathbf{w}_+,\, b_+,\, \boldsymbol{\xi}} & \frac{1}{2}\|A\mathbf{w}_+ + \mathbf{e}_+b_+\|_2^2 + c_1\mathbf{e}_-^T\boldsymbol{\xi} + \frac{1}{2}c_2(\|\mathbf{w}_+\|_2^2 + b_+^2) + \frac{1}{2}c_3\mathbf{w}_+^T\Sigma_+\mathbf{w}_+, \\ \text{s.t.} & -(B\mathbf{w}_+ + \mathbf{e}_-b_+) + \boldsymbol{\xi} \geq \mathbf{e}_-,\ \boldsymbol{\xi} \geq \mathbf{0}, \end{cases} \tag{5}$$

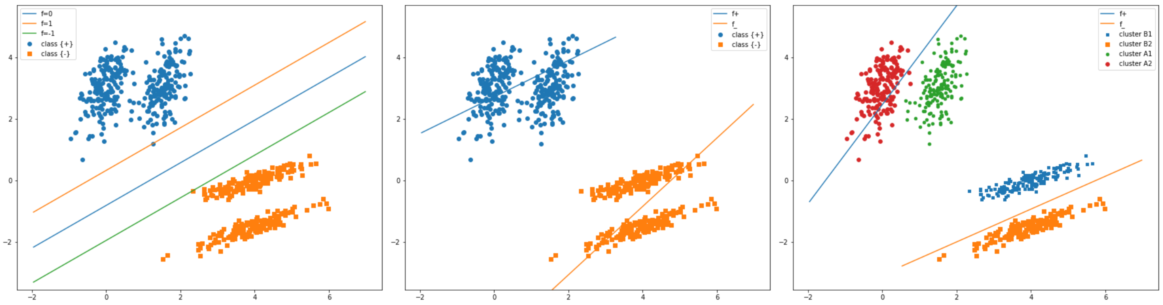
$$\begin{cases} \min\limits_{\mathbf{w}_-,\, b_-,\, \boldsymbol{\eta}} & \frac{1}{2}\|B\mathbf{w}_- + \mathbf{e}_-b_-\|_2^2 + c_4\mathbf{e}_+^T\boldsymbol{\eta} + \frac{1}{2}c_5(\|\mathbf{w}_-\|_2^2 + b_-^2) + \frac{1}{2}c_6\mathbf{w}_-^T\Sigma_-\mathbf{w}_-, \\ \text{s.t.} & (A\mathbf{w}_- + \mathbf{e}_+b_-) + \boldsymbol{\eta} \geq \mathbf{e}_+,\ \boldsymbol{\eta} \geq \mathbf{0}, \end{cases} \tag{6}$$

where $c_1, \dots, c_6 \geq 0$ are penalty coefficients to adjust the role between terms in the objective functions, $\boldsymbol{\xi} \in \mathbb{R}^{m_B}, \boldsymbol{\eta} \in \mathbb{R}^{m_A}$ are vectors of slack variables. $\Sigma_+ = \Sigma_{1+} + \Sigma_{2+} + \cdots + \Sigma_{k+}$, $\Sigma_- = \Sigma_{1-} + \Sigma_{2-} + \cdots + \Sigma_{l-}$, Σ_{i+} and Σ_{i-} are respectively the covariance matrices corresponding to the clusters A_i and B_i, $\mathbf{e}_+ \in \mathbb{R}^{m_A}$, $\mathbf{e}_- \in \mathbb{R}^{m_B}$ are vectors of ones. A new data point is assigned into class $\{+\}$ or $\{-\}$ in the same manner as in TWSVM. In the problem (5), $\frac{1}{2}\|A\mathbf{w}_+ + \mathbf{e}_+b_+\|_2^2$ is the sum of the squares of the distances from data points in class $\{+\}$ to the hyperplane $\{f_+(\mathbf{x}) = 0\}$. $c_1\mathbf{e}_-^T\boldsymbol{\xi}$ is the sum of errors, $\frac{1}{2}c_2(\|\mathbf{w}_+\|_2^2 + b_+^2)$ is the regularization, $\frac{1}{2}c_3\mathbf{w}_+^T\Sigma_+\mathbf{w}_+$ is the sum of covariance matrices with the cluster granularity of class $\{+\}$ projected onto vector $\mathbf{w}_+$. The constraints of (5) is defined by the points of class $\{-\}$. The problem (6) is similarly established for class $\{-\}$ with the constraints defined by class $\{+\}$. Thus, S-TWSVM exploits structural information with cluster granularity of one class in each problem. Therefore, the data simulation capability of S-TWSVM is more accurate than that of TWSVM (see Figure 1.c). However, as the data become more complex, this ability of the S-TWSVM remains limited (see Figure 2.c).

By using sufficient information about the cluster granularity of the classes, S-TWSVM finds the hyperplanes that represent the classes better than TWSVM. However, due to the calculation of all the covariance matrices Σ_{i+} and Σ_{i-} of both classes, the training speed of S-TWSVM is not improved compared to that of TWSVM (see Table 1).

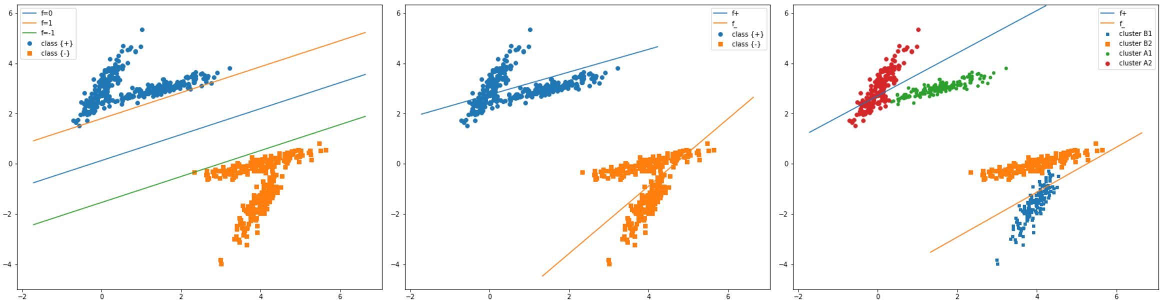
3. WEIGHTED STRUCTURAL - SUPPORT VECTOR MACHINE

In this section, we describe a new classification algorithm: Weighted Structural - Support Vector Machine (called WS-SVM). Both theoretically and experimentally, we show that WS-SVM overcomes the S-TWSVM in data simulation and training speed.



a *b* *c*

Figure 1: For datasets in which the same trend constitutes clusters in each class, the SVM (a) simulates the distribution trend of the two classes being the same, TWSVM (b) simulates the distribution trend of the two classes as different, but does not actually simulate the distribution trend of data in each class, S-TWSVM (c) simulates the distribution trend in each class quite accurately.



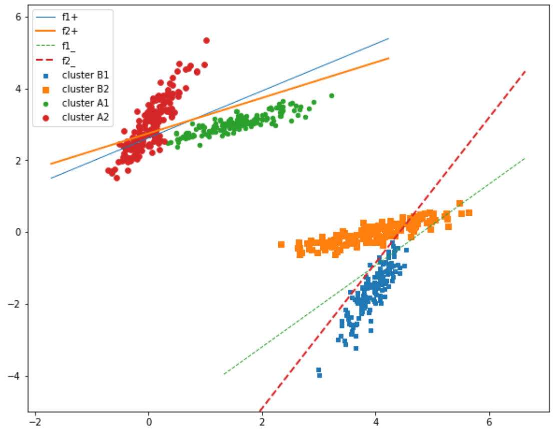
a *b* *c*

Figure 2: For datasets in which clusters in each class are constructed according to different distribution trends, SVM (a), TWSVM (b), and S-TWSVM (c) all have difficulty in simulating the distribution trend of the data.

Table 1: Training time (seconds), with the dataset shown in Figure 2

Algorithms	1200 data points	1600 data points	2000 data points	2400 data points
SVM	2.143	5.308	6.754	11.512
TWSVM	0.351	0.700	1.193	1.805
S-TWSVM	0.388	0.853	1.434	2.158

Figure 3: (WS-SVM) f_{1+} (thin solid line) shows that cluster B_1 (small square) tends to deviate vertically, f_{2+} (bold solid line) shows that cluster B_2 (large square) tends to deviate horizontally. f_{1-} (thin dashed line) shows that cluster A_1 (small circle) tends to deviate horizontally, f_{2-} (bold dashed line) shows that cluster A_2 (large circle) tends to deviate vertically.



Similar to S-TWSVM, WS-SVM also has two steps. The first step is to extract structural information within classes by Ward's linkage clustering method [15, 12, 13]; The second step is the model learning. Suppose that there are k clusters $A_1, \dots, A_k$ in class $\{+\}$ and l clusters $B_1, \dots, B_l$ in class $\{-\}$. WS-SVM uses a class-vs-clusters strategy to determine $(l+k)$ hyperplanes such that each of which is closer to one class and far away from one cluster in the other class. Specifically, the method need to find l hyperplanes such that the $i-$th hyperplane, $f_{i+}(\mathbf{x}) = \mathbf{w}_{i+}^T\mathbf{x} + b_{i+} = 0$, is closer to class $\{+\}$ and far away from cluster B_i of class $\{-\}$; Also, It need to find k hyperplanes such that the $i-$th hyperplane, $f_{i-}(\mathbf{x}) = \mathbf{w}_{i-}^T\mathbf{x} + b_{i-} = 0$, is closer to class $\{-\}$ and far away from cluster A_i of class $\{+\}$ (see Figure 3). Here $\mathbf{w}_{i+}, \mathbf{w}_{i-} \in \mathbb{R}^n, b_{i+}, b_{i-} \in \mathbb{R}$.

The classifier is now selected as

$$f(\mathbf{x}) = \underset{+,\ -}{\operatorname{argmin}}(f_+(\mathbf{x}),\ f_-(\mathbf{x})), \tag{7}$$

with
$$f_+(\mathbf{x}) = \sum_{i=1}^{l} \frac{m_{Bi}}{m_B}|f_{i+}(\mathbf{x})|, \quad f_-(\mathbf{x}) = \sum_{i=1}^{k} \frac{m_{Ai}}{m_A}|f_{i-}(\mathbf{x})|. \tag{8}$$

From the definition, we see that $f_+(\mathbf{x})$ is the average, taking into account the weights, distances from $\mathbf{x}$ to the hyperplanes $\{f_{i+}(\mathbf{x}) = 0\}$. The $i-$th hyperplane's weight is proportional to m_{Bi} - the number of data points in the cluster Bi. Similarly, $f_-(\mathbf{x})$ is the weighted average of distances from $\mathbf{x}$ to the hyperplanes $\{f_{i-}(\mathbf{x}) = 0\}$. By virtue of (7), a new data point $\mathbf{x}$ is classified into class $\{+\}$ or class $\{-\}$ depending on whether $f_+(\mathbf{x})$ is less than or greater than $f_-(\mathbf{x})$.

3.1. The linear case

WS-SVM determines $(l+k)$ hyperplanes by solving $(l+k)$ QPPs as follows

$$(\text{WS-SVM}_i^+)\begin{cases} \min\limits_{\mathbf{w}_{i+}, b_{i+}, \boldsymbol{\xi}_i} \frac{1}{2}\|A\mathbf{w}_{i+} + \mathbf{e}_{m_A} b_{i+}\|_2^2 + c_+\mathbf{e}_{m_{Bi}}^T\boldsymbol{\xi}_i + \frac{\mu_+}{2}(\|\mathbf{w}_{i+}\|_2^2 + b_{i+}^2) + \frac{\lambda_+}{2}\mathbf{w}_{i+}^T\mathit{\Sigma}_+\mathbf{w}_{i+}, \\ \text{s.t.} \ -(B_i\mathbf{w}_{i+} + \mathbf{e}_{m_{Bi}} b_{i+}) + \boldsymbol{\xi}_i \geq \mathbf{e}_{m_{Bi}};\ \boldsymbol{\xi}_i \geq \mathbf{0}, \end{cases}$$

i= $1, \dots, l$, and

$$(\text{WS-SVM}_i^-)\begin{cases} \min\limits_{\mathbf{w}_{i-}, b_{i-}, \boldsymbol{\eta}_i} \frac{1}{2}\|B\mathbf{w}_{i-} + \mathbf{e}_{m_B} b_{i-}\|_2^2 + c_-\mathbf{e}_{m_{Ai}}^T\boldsymbol{\eta}_i + \frac{\mu_-}{2}(\|\mathbf{w}_{i-}\|_2^2 + b_{i-}^2) + \frac{\lambda_-}{2}\mathbf{w}_{i-}^T\mathit{\Sigma}_-\mathbf{w}_{i-}, \\ \text{s.t.} \quad (A_i\mathbf{w}_{i-} + \mathbf{e}_{m_{Ai}} b_{i-}) + \boldsymbol{\eta}_i \geq \mathbf{e}_{m_{Ai}};\ \boldsymbol{\eta}_i \geq \mathbf{0}, \end{cases}$$

$i = 1, \dots, k$. Here, $\mathbf{e}_{m_A} \in \mathbb{R}^{m_A \times 1}$, $\mathbf{e}_{m_B} \in \mathbb{R}^{m_B \times 1}$, $\mathbf{e}_{m_{Ai}} \in \mathbb{R}^{m_{Ai} \times 1}$, $\mathbf{e}_{m_{Bi}} \in \mathbb{R}^{m_{Bi} \times 1}$ are vectors of ones; $\boldsymbol{\eta}_i \in \mathbb{R}^{m_{Ai} \times 1}$, $\boldsymbol{\xi}_i \in \mathbb{R}^{m_{Bi} \times 1}$ are vectors of slack variables; $\mathit{\Sigma}_+ = \Sigma_{1+} + \cdots + \Sigma_{k+}$, $\mathit{\Sigma}_- = \Sigma_{1-} + \cdots + \Sigma_{l-}$, $\mathit{\Sigma}_{i+}$ and $\mathit{\Sigma}_{i-}$ are the covariance matrices corresponding to the clusters A_i and B_i; c_+, c_-, λ_+, λ_-, μ_+, μ_- are penalty coefficients to adjust the role between terms in the objective functions; $\mathbf{w}_{i+}^T\mathit{\Sigma}_+\mathbf{w}_{i+}$ is the sum of covariance matrices with cluster granularity of class $\{+\}$ projected onto to $\mathbf{w}_{i+}$ and $\mathbf{w}_{i-}^T\mathit{\Sigma}_-\mathbf{w}_{i-}$ is the sum of covariance matrices with cluster granularity of class $\{-\}$ projected onto $\mathbf{w}_{i-}$. The constraints in problem (WS-SVM_i^+) is defined by the points of cluster B_i and the constraints in problem (WS-SVM_i^-) is defined by cluster A_i.

The Lagrangian of (WS-SVM$_i^+$) is given by

$$\begin{aligned}L_i(\mathbf{w}_{i+}, b_{i+}, \boldsymbol{\xi}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) =& \frac{1}{2}\|A\mathbf{w}_{i+} + \mathbf{e}_{m_A} b_{i+}\|_2^2 + c_+ \mathbf{e}_{m_{Bi}}^T \boldsymbol{\xi}_i + \frac{1}{2}\mu_+(\|\mathbf{w}_{i+}\|_2^2 + b_{i+}^2)\\ &+ \frac{1}{2}\lambda_+ \mathbf{w}_{i+}^T \Sigma_+ \mathbf{w}_{i+} - \boldsymbol{\alpha}_i^T(-(B_i\mathbf{w}_{i+} + \mathbf{e}_{m_{Bi}} b_{i+}) + \boldsymbol{\xi}_i - \mathbf{e}_{m_{Bi}}) - \boldsymbol{\beta}_i^T \boldsymbol{\xi}_i.\end{aligned}$$

Therefore, the KKT conditions of problem (WS-SVM$_i^+$) are

$$A^T(A\mathbf{w}_{i+} + \mathbf{e}_{m_A} b_{i+}) + \mu_+ \mathbf{w}_{i+} + \lambda_+ \Sigma_+ \mathbf{w}_{i+} + B_i^T \boldsymbol{\alpha}_i = \mathbf{0}, \tag{9}$$

$$\mathbf{e}_{m_A}^T(A\mathbf{w}_{i+} + \mathbf{e}_{m_A} b_{i+}) + \mu_+ b_{i+} + \mathbf{e}_{m_{Bi}}^T \boldsymbol{\alpha}_i = 0, \tag{10}$$

$$\mathbf{e}_{m_{Bi}}^T c_+ - \boldsymbol{\alpha}_i - \boldsymbol{\beta}_i = \mathbf{0}, \tag{11}$$

$$\boldsymbol{\alpha}_i^T(-(B_i\mathbf{w}_{i+} + \mathbf{e}_{m_{Bi}} b_{i+}) + \boldsymbol{\xi}_i - \mathbf{e}_{m_{Bi}}) = 0, \tag{12}$$

$$\boldsymbol{\beta}_i^T \boldsymbol{\xi}_i = 0. \tag{13}$$

By defining $H = [A\ \mathbf{e}_{m_A}]$, $\mathbf{u}_{i+} = \begin{bmatrix}\mathbf{w}_{i+}\\ b_{i+}\end{bmatrix}$, $F_+ = \begin{bmatrix}\Sigma_+ & \mathbf{0}\\ \mathbf{0} & 0\end{bmatrix}$, $G_i = [B_i\ \mathbf{e}_{m_{Bi}}]$, and I being the identity matrix of order $(n+1)$, it follows from (9) and (10) that

$$[H^T H + \mu_+ I + \lambda_+ F_+]\mathbf{u}_{i+} + G_i^T \boldsymbol{\alpha}_i = \mathbf{0},$$

which implies

$$\mathbf{u}_{i+} = -[H^T H + \mu_+ I + \lambda_+ F_+]^{-1} G_i^T \boldsymbol{\alpha}_i,\ i = 1, \ldots, l. \tag{14}$$

Substituting (14) into the Lagrangian, and combined with the conditions (11), (12), and (13) we have the dual problem of (WS-SVM$_i^+$) as follows

$$(\text{DWS-SVM}_i^+)\begin{cases}\max\limits_{\boldsymbol{\alpha}_i} & \mathbf{e}_{m_{Bi}}^T \boldsymbol{\alpha}_i - \frac{1}{2}\boldsymbol{\alpha}_i^T G_i [H^T H + \mu_+ I + \lambda_+ F_+]^{-1} G_i^T \boldsymbol{\alpha}_i,\\ \text{s.t.} & \mathbf{0} \le \boldsymbol{\alpha}_i \le c_+ \mathbf{e}_{m_{Bi}}.\end{cases} \tag{15}$$

In the same way, we also obtain the dual problem of (WS-SVM$_i^-$)

$$(\text{DWS-SVM}_i^-)\begin{cases}\max\limits_{\boldsymbol{\gamma}_i} & \mathbf{e}_{m_{Ai}}^T \boldsymbol{\gamma}_i - \tfrac{1}{2}\boldsymbol{\gamma}_i^T H_i (G^T G + \mu_- I + \lambda_- F_-)^{-1} H_i^T \boldsymbol{\gamma}_i,\\ \text{s.t.} & \mathbf{0} \le \boldsymbol{\gamma}_i \le c_- \mathbf{e}_{m_{Ai}},\end{cases} \tag{16}$$

where $G = [B\ \mathbf{e}_{m_B}]$, $F_- = \begin{bmatrix}\Sigma_- & \mathbf{0}\\ \mathbf{0} & 0\end{bmatrix}$, $H_i = [A_i\ \mathbf{e}_{m_{Ai}}]$. The augmented vectors $\mathbf{u}_{i-} = \begin{bmatrix}\boldsymbol{w}_{i-}\\ b_{i-}\end{bmatrix}$ are also given by

$$\mathbf{u}_{i-} = [G^T G + \mu_- I + \lambda_- F_-]^{-1} H_i^T \boldsymbol{\gamma_i},\ i = 1, \ldots, k. \tag{17}$$

Algorithm 1. [Linear WS-SVM]

Give m data points in $\mathbb{R}^n$ represented by a $m \times n$ matrix C. Class $\{+\}$ includes m_A points represented by a $m_A \times n$ matrix A, class $\{-\}$ includes m_B points represented by a $m_B \times n$ matrix B. We generate the linear classifier $f(\mathbf{x})$ as follows:

(i) Clustering dataset by using Ward's linkage [12]. Suppose that, there are k clusters in class $\{+\}$, each cluster consists of m_{Ai} points and is represented by matrix $A_i \subset \mathbb{R}^{m_{Ai}\times n}$, there are l clusters in class $\{-\}$, each cluster consists of m_{Bi} points and is represented by matrix $B_i \subset \mathbb{R}^{m_{Bi}\times n}$.

(ii) Solving the problems (15), (16) to obtain $\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_l$; $\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_k$.

(iii) Determining $(\mathbf{w}_{1+}, b_{1+}), \dots, (\mathbf{w}_{l+}, b_{l+})$ and $(\mathbf{w}_{1-}, b_{1-}), \dots, (\mathbf{w}_{k-}, b_{k-})$ via (14), (17).

(iv) Classifying a new data $\mathbf{x}$ by using (7) and (8).

Remark 1. In Linear WS-SVM, we have to solve $(l+k)$ problems of the form (15) or (16). These are QPPs with the number of decisive variables being m_{Bi} or m_{Ai}. If the number of data samples in each cluster is approximately equal to $\dfrac{m}{k+l}$, then the complexity of the algorithm is $O\big((k+l)(\dfrac{m}{k+l})^3\big) = O(\dfrac{m^3}{(k+l)^2})$. While the complexity of SVM and TWSVM are $O(m^3)$ and $O(\dfrac{m^3}{4})$, respectively. From the above formula, it is easy to see that the more clusters in the data set, the less the runtime of WS-SVM. This is also clearly demonstrated in experimentation.

Remark 2. There are many different clustering methods that we can use to implement for Step (i) of Algorithm 1. Here we choose Ward's clustering algorithm (with k and l being chosen independently via elbow method) for convenience in comparison with S-TWSVM because the authors have also used this technique in the experiment in [12]. When processing actual data we can choose another clustering algorithm that is more efficient.

3.2. The nonlinear case

Let $\Phi : \mathbb{R}^n \to \mathbb{H}$ be a nonlinear mapping, where $\mathbb{H}$ is a Hilbert space whose dimension is not less than n (maybe infinite-dimensional). Since $\mathbb{S} = \operatorname{span}(\Phi(C^T))$ is a subspace of $\mathbb{H}$ whose dimension does not exceed m, we can consider $\mathbb{S}$ as an Euclidean space and $\Phi : \mathbb{R}^n \to \mathbb{S}$. Suppose that after the clustering step on space $\mathbb{S}$ we obtain k clusters $\Phi(A_1), \dots, \Phi(A_k)$ in the class $\Phi(A)$, each cluster $\Phi(A_i)$ consists of m_{Ai} data points; and l clusters $\Phi(B_1), \dots, \Phi(B_l)$ in the class $\Phi(B)$, each cluster $\Phi(B_i)$ consists of m_{Bi} data points. In space $\mathbb{S}$, a hyperplane $\Phi(\mathbf{x}^T)\mathbf{h}+b = 0$ (with $\mathbf{h} \in \mathbb{S}$ being the normal vector) can be rewritten as $\Phi(\mathbf{x}^T)\Phi(C^T)\mathbf{u}+b = 0$ for some vector $\mathbf{u} \in \mathbb{R}^m$. Therefore, by defining $\Phi(\mathbf{x}^T)\Phi(C^T) = K(\mathbf{x}^T, C^T)$, the hyperplane has the form $K(\mathbf{x}^T, C^T)\mathbf{u} + b = 0$, where K is a predefined kernel [16].

WS-SVM determines l hyperplanes such that the $i-$th one $K(\mathbf{x}^T,\ C^T)\mathbf{u}_{i+} + b_{i+} = 0$ is closer to class $\Phi(A)$ and far away from cluster $\Phi(B_i)$. It also determines k hyperplanes such that the $i-$th one $K(\mathbf{x}^T,\ C^T)\mathbf{u}_{i-} + b_{i-} = 0$ is closer to class $\Phi(B)$ and far away from cluster $\Phi(A_i)$. Specifically, we have $(l+k)$ QPPs as follows

$$\begin{cases} \min\limits_{\mathbf{u}_{i+}, b_{i+}, \boldsymbol{\xi}_i} \dfrac{1}{2}\|K(A, C^T)\mathbf{u}_{i+} + \mathbf{e}_{m_A} b_{i+}\|_2^2 + c_+ \mathbf{e}_{m_{Bi}}^T \boldsymbol{\xi}_i + \dfrac{\mu_+}{2}(\|\mathbf{u}_{i+}\|_2^2 + b_{i+}^2) + \dfrac{\lambda_+}{2}\mathbf{u}_{i+}^T \Phi(C) \varSigma_+^{\Phi} \Phi(C)^T \mathbf{u}_{i+}, \\ \text{s.t.} \quad -(K(B_i,\ C^T)\mathbf{u}_{i+} + \mathbf{e}_{m_{Bi}} b_{i+}) + \boldsymbol{\xi}_i \geq \mathbf{e}_{m_{Bi}},\ \boldsymbol{\xi}_i \geq \mathbf{0}, \end{cases} \tag{18}$$

$i = 1, \dots, l$, and

$$\begin{cases} \min\limits_{\mathbf{u}_{i-}, b_{i-}, \boldsymbol{\eta}_i} \frac{1}{2}\|K(B, C^T)\mathbf{u}_{i-} + \mathbf{e}_{m_B} b_{i-}\|_2^2 + c_- \mathbf{e}_{m_{Ai}}^T \boldsymbol{\eta}_i + \frac{\mu_-}{2}(\|\mathbf{u}_{i-}\|_2^2 + b_{i-}^2) + \frac{\lambda_-}{2}\mathbf{u}_{i-}^T \Phi(C) \Sigma_-^{\Phi} \Phi(C)^T \mathbf{u}_{i-}, \\ \text{s.t.} \quad (K(A_i,\ C^T)\mathbf{u}_{i-} + \mathbf{e}_{m_{Ai}} b_{i-}) + \boldsymbol{\eta}_i \geq \mathbf{e}_{m_{Ai}},\ \boldsymbol{\eta}_i \geq \mathbf{0}, \end{cases} \tag{19}$$

$i = 1, \dots, k$. Here, $\Sigma_+^{\Phi} = \Sigma_{1+}^{\Phi} + \dots + \Sigma_{k+}^{\Phi}$, $\Sigma_-^{\Phi} = \Sigma_{1-}^{\Phi} + \dots + \Sigma_{l-}^{\Phi}$, Σ_{i+}^{Φ}, Σ_{i-}^{Φ} are respectively the covariance matrices corresponding to clusters $\Phi(A_i)$ and $\Phi(B_i)$.

The classification function in the nonlinear case is selected as

$$f(\mathbf{x}) = \underset{+,\ -}{\operatorname{argmin}}(f_+(\mathbf{x}),\ f_-(\mathbf{x})), \tag{20}$$

$$\text{with } f_+(\mathbf{x}) = \sum_{i=1}^{l} \frac{m_{Bi}}{m_B}|K(\mathbf{x}^T, C^T)\mathbf{u}_{i+} + b_{i+}|,\ f_-(\mathbf{x}) = \sum_{i=1}^{k} \frac{m_{Ai}}{m_A}|K(\mathbf{x}^T, C^T)\mathbf{u}_{i-} + b_{i-}|. \tag{21}$$

The dual problem of (18)

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}_i} \quad \mathbf{e}_{m_{Bi}}^T \boldsymbol{\alpha}_i - \frac{1}{2}\boldsymbol{\alpha}_i^T G_i [H^T H + \mu_+ I + \lambda_+ F_+]^{-1} G_i^T \boldsymbol{\alpha}_i, \\ \text{s.t.} \quad \mathbf{0} \leq \boldsymbol{\alpha}_i \leq c_+ \mathbf{e}_{m_{Bi}}, \end{cases} \tag{22}$$

where $H = [K(A, C^T)\ \mathbf{e}_{m_A}]$, $F_+ = \begin{bmatrix} \Phi(C)\Sigma_+^{\Phi}\Phi(C)^T & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$, $G_i = [K(B_i, C^T)\ \mathbf{e}_{m_{Bi}}]$, I is the identity matrix of order $(m+1)$, and the augmented vectors

$$\bar{\mathbf{u}}_{i+} = \begin{bmatrix} \mathbf{u}_{i+} \\ b_{i+} \end{bmatrix} = -[H^T H + \mu_+ I + \lambda_+ F_+]^{-1} G_i^T \boldsymbol{\alpha}_i. \tag{23}$$

The dual problem of (19)

$$\begin{cases} \max\limits_{\boldsymbol{\gamma}_i} \quad \mathbf{e}_{m_{Ai}}^T \boldsymbol{\gamma}_i - \frac{1}{2}\boldsymbol{\gamma}_i^T H_i (G^T G + \mu_- I + \lambda_- F_-)^{-1} H_i^T \boldsymbol{\gamma}_i, \\ \text{s.t.} \quad \mathbf{0} \leq \boldsymbol{\gamma}_i \leq c_- \mathbf{e}_{m_{Ai}}, \end{cases} \tag{24}$$

where $G = [K(B, C^T)\ \mathbf{e}_{m_B}]$, $F_- = \begin{bmatrix} \Phi(C)\Sigma_-^{\Phi}\Phi(C)^T & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$, $H_i = [K(A_i,\ C^T)\ \mathbf{e}_{m_{Ai}}]$, and

$$\bar{\mathbf{u}}_{i-} = \begin{bmatrix} \mathbf{u}_{i-} \\ b_{i-} \end{bmatrix} = (G^T G + \mu_- I + \lambda_- F_-)^{-1} H_i^T \boldsymbol{\gamma}_i. \tag{25}$$

Remark 3. We can calculate the matrix F_+ as follows. For each $A_i \in \mathbb{R}^{m_{Ai} \times n}$ we denote by $MA_i \in \mathbb{R}^{m_{Ai} \times n}$ the average matrix of A_i (i.e., all rows of MA_i are the same and equal to the average vector of A_i). Therefore, $\Sigma_{i+}^{\Phi} = \frac{1}{m_{A_i}}(\Phi(A_i) - \Phi(MA_i))^T(\Phi(A_i) - \Phi(MA_i))$,

$$\begin{aligned} \Phi(C)\Sigma_{i+}^{\Phi}\Phi(C)^T &= \Big[\frac{1}{\sqrt{m_{A_i}}}(\Phi(A_i) - \Phi(MA_i))\Phi(C)^T\Big]^T \Big[\frac{1}{\sqrt{m_{A_i}}}(\Phi(A_i) - \Phi(MA_i))\Phi(C)^T\Big] \\ &= \Big[\frac{1}{\sqrt{m_{A_i}}}(K(A_i, C^T) - K(MA_i, C^T))\Big]^T \Big[\frac{1}{\sqrt{m_{A_i}}}(K(A_i, C^T) - K(MA_i, C^T))\Big]. \end{aligned}$$

Similarly, we can use the following formula to calculate the matrix F_{-}

$$\begin{aligned}\Phi(C)\Sigma^{\Phi}_{i-}\Phi(C)^T &= \Big[\frac{1}{\sqrt{m_{B_i}}}(\Phi(B_i)-\Phi(MB_i))\Phi(C)^T\Big]^T\Big[\frac{1}{\sqrt{m_{B_i}}}(\Phi(B_i)-\Phi(MB_i))\Phi(C)^T\Big]\\ &= \Big[\frac{1}{\sqrt{m_{B_i}}}(K(B_i,C^T)-K(MB_i,C^T))\Big]^T\Big[\frac{1}{\sqrt{m_{B_i}}}(K(B_i,C^T)-K(MB_i,C^T))\Big],\end{aligned}$$

where $MB_i \in \mathbb{R}^{m_{Bi}\times n}$ is the average matrix of B_i.

Remark 4. In (22), (24) we need to compute the inverse of square matrices in order $(m+1)$. This work will become difficult when m is large. So it is necessary to reduce the size of those matrices. This problem can be solved by using the Sherman-Morrison-Woodbury (SMW) formula [17] as in [11].

Algorithm 2. [Nonlinear WS-SVM]

Give m data points in $\mathbb{R}^n$ represented by the $m \times n$ matrix C. Class $\{+\}$ includes m_A points represented by the $m_A \times n$ matrix A, class $\{-\}$ includes m_B points represented by the $m_B \times n$ matrix B. We generate the nonlinear classifier $f(\mathbf{x})$ as follows:

(i) Choosing a kernel function $K(\mathbf{x}^T, C^T)$, typically the Gaussian kernel [16].

(ii) Clustering the dataset by using Ward's linkage [12]. Suppose that, there are k clusters in class $\{+\}$, each cluster consists of m_{Ai} points and represented by matrix $\Phi(A_i)$; and there are l clusters in class $\{-\}$, each cluster consists of m_{Bi} points and represented by matrix $\Phi(B_i)$.

(iii) Solving the problems (22) and (24) to obtain $\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_l$; $\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_k$.

(iv) Determining $(\mathbf{u}_{1+}, b_{1+}), \dots, (\mathbf{u}_{l+}, b_{l+})$ and $(\mathbf{u}_{1-}, b_{1-}), \dots, (\mathbf{u}_{k-}, b_{k-})$ via (23), (25).

(v) Classifying a new data $\mathbf{x}$ by using (20) and (21).

Remark 5. As usual, the linear algorithm will be applied when the training data is almost linearly separable. If the data overlap occurs too seriously, the linear algorithm will not work effectively, and the accuracy is not high. In that situation, the nonlinear algorithm should be used for better accuracy.

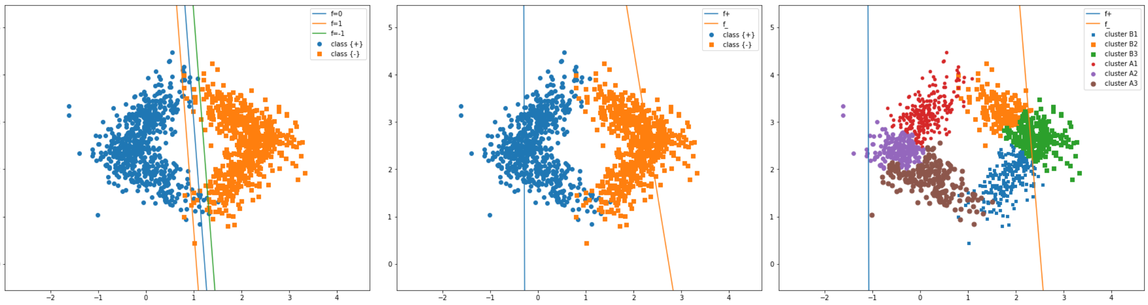
4. EXPERIMENTS

In this section, we compare the WS-SVM against S-TWSVM [12] and TWSVM [11] on various datasets. All algorithms (our model, S-TWSVM as in [12], TWSVM as in [11]) are settled by version 3.8.3 of Python programming language, and run on a Laptop with an AMD Ryzen 5 with 8GB RAM. We use the following libraries: "scipy.cluster.hierarchy" to cluster the data, "cvxopt" to solve the QPP, "matplotlib" to show figure, "panda" and "numpy" to process data, 'sklearn' to evaluate and adjust hyperparameters of all models.

For simplicity, let $c_+ = c_-$, $\mu_+ = \mu_-$, $\lambda_+ = \lambda_-$ in WS-SVM, $c_1 = c_4, c_2 = c_5, c_3 = c_6$ as in S-TWSVM [12], all hyperparameters belong to the set $\{0.0001,\ 0.001,\ 0.1,\ 1,\ 10,\ 100,\ 1000\}$ and are obtained by using Grid-search technique. All settings are uploaded to [18].

4.1. Toy data

We first experiment with 2-D toy data. Note that the training time of WS-SVM and S-TWSVM includes clustering time. The 2-D toy data consists of 800 points belonging to class $\{+\}$ and 800 points belonging to class $\{-\}$, randomly generated according to two Gaussian distributions in each class. The dataset is scaled 90/10, which means 90% of the data was used for training and the rest of the data for testing. We use standard 10-fold cross validation (CV) to evaluate the testing accuracy of all models. Implementing SVM, TWSVM, S-TWSVM, we obtained the results as shown in Figure 4, and WS-SVM, we obtained the results as shown in Figure 5.



a *b* *c*

Figure 4: SVM (a): The run-time: 10.677 (s); The CV accuracy: 98.516 +/- 0.649. TWSVM (b): The run-time: 0.510 (s); The CV accuracy: 98.438 +/- 0.988. S-TWSVM (c): The run-time: 0.562 (s); The CV accuracy: 98.594 +/- 0.911.

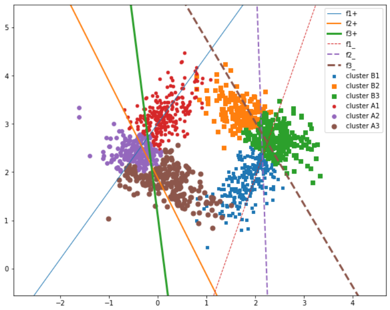


Figure 5: WS-SVM: The run-time: 0.142 (s); The CV accuracy: 98.359 +/- 0.547

4.2. UCI datasets

Next, we implement these algorithms on the UCI datasets [19] which have been experimented in [12] and [11]. We randomly select 90% of each extracted dataset for training and 10% for testing. We also use 10-fold cross validation to evaluate the accuracy of all algorithms. All hyperparameters belong to the set $\{0.0001, 0.001, 0.1, 1, 10, 100, 1000\}$ and are obtained by using Grid-search technique. The results are shown in Table 3, Table 4 (by applying Algorithm 1), and Table 5 (by applying Algorithm 2).

Table 3 shows that the training time of WS-SVM is better than that of S-TWSVM and TWSVM, while Table 4 shows that the accuracy in data classification is not much different between methods. WS-SVM training time is even better when the data size is large and the data is constructed in many clusters. That is shown in Table 2.

Table 2: Training time (s) with the increase in the size of the data set. The computational complexity of WS-SVM is clearly less than that of S-TWSVM and TWSVM.

Algorithms	4000 data points	6000 data points	8000 data points	10000 data points
TWSVM	7.267	21.872	58.101	104.355
S-TWSVM	5.614	16.244	44.397	75.549
WS-SVM	1.33	6.251	8.675	16.863

Table 3: Test training time (s) with a Linear Kernel (Algorithm 1)

Dataset	Number of clusters	WS-SVM	S-TWSVM	TWSVM
Hepatitis(155 x 19)	(5×3)	0.001	0.005	0.004
Australian(690 x 14)	(4×5)	0.033	0.089	0.043
BUPA-liver(345 x 6)	(4×2)	0.015	0.020	0.014
CMC (844 x 9)	(3×5)	0.047	0.099	0.083
Credit (690 x 19)	(2×3)	0.037	0.055	0.050
Diabetes (768 x 8)	(5×2)	0.099	0.166	0.117
Flare-solar(1066 x 10)	(2×2)	0.117	0.330	0.278
German (1000 x 20)	(3×3)	0.065	0.180	0.166
Heart-statlog (270 x 13)	(3×4)	0.011	0.011	0.011
Image (2310 x 18)	(3×2)	0.356	1.279	1.444
Ionosphere (350 x 34)	(3×2)	0.012	0.017	0.014
Spect (265 x 22)	(3×3)	0.012	0.016	0.012
Sonar (208 x 60)	(6×3)	0.008	0.009	0.008
Heart-c (303 x 13)	(8×2)	0.013	0.019	0.010

Table 4: Test set accuracy (%) with a Linear Kernel (Algorithm 1)

Dataset	WS-SVM	S-TWSVM	TWSVM
Hepatitis(155 x 19)	83.516 +/- 10.078	87.088 +/- 8.888	84.176 +/- 8.910
Australian(690 x 14)	86.452 +/- 4.903	86.613 +/- 4.507	85.968 +/- 4.565
BUPA-liver(345 x 6)	68.065 +/- 7.133	67.097 +/- 11.613	65.806 +/- 5.983
CMC (844 x 9)	65.223 +/- 3.637	65.221 +/- 5.010	64.821 +/- 5.204
Credit (690 x 19)	86.157 +/- 3.127	86.964 +/- 4.444	85.192 +/- 4.223
Diabetes (768 x 8)	75.118 +/- 4.993	77.000 +/- 5.759	77.205 +/- 4.605
Flare-solar(1066 x 10)	81.760 +/- 4.127	81.969 +/- 4.200	81.761 +/- 4.306
German (1000 x 24)	69.375 +/- 5.356	71.667 +/- 5.449	71.042 +/- 7.184
Heart-statlog (270 x 13)	84.850 +/- 7.223	86.483 +/- 5.367	85.650 +/- 5.776
Image (2310 x 18)	86.196 +/- 1.619	84.415 +/- 1.244	85.089 +/- 1.745
Ionosphere (350 x 34)	92.056 +/- 6.574	91.744 +/- 4.982	90.504 +/- 6.104
Spect (265 x 22)	81.449 +/- 6.503	84.801 +/- 4.992	83.134 +/- 6.120
Sonar (208 x 60)	80.263 +/- 11.966	78.099 +/- 8.244	78.129 +/- 10.286
Heart-c (303 x 13)	84.577 +/- 5.116	83.836 +/- 4.668	83.823 +/- 4.435

Table 5: Test set accuracy (%) with an RBF Kernel (Algorithm 2)

Dataset(mxn)	WS-SVM	S-TWSVM	TWSVM
Hepatitis(155 x 19)	84.835 +/- 9.939	83.407 +/- 12.052	83.462 +/- 11.958
BUPA-liver(345 x 6)	70.000 +/- 6.297	69.032 +/- 3.290	72.258 +/- 3.592
Votes (303 x 13)	95.154 +/- 2.626	95.667 +/- 2.519	95.917 +/- 2.321
WPBC (198x35)	81.669 +/- 8.879	79.412 +/- 9.727	80.588 +/- 10.156

5. CONCLUSIONS

This paper has proposed a new Weighted Structural - Support Vector Machine (known as WS-SVM) for classification problems with a class-vs-clusters strategy. This algorithm is performed in two steps: The first step is to extract structural information of the data using Ward's linkage clustering method; The second step is to apply structural information with cluster granularity to the learning model. The classifier is based on the weighted average distances from the data point to the class representative hyperplanes. Both theory and experiment show that training time of WS-SVM is better than S-TWSVM and TWSVM in most cases. Besides, when the data is large, and there are many clusters with too different distribution trends, the WS-SVM algorithm effectively simulates the distribution trend of clusters and thus improves the accuracy in data classification. When $k = l = 1$, WS-SVM is exactly S-TWSVM, and if $\lambda_+ = \lambda_- = 0$ it becomes TWSVM again. The WS-SVM algorithm generally only achieves high accuracy when the clustering within each class is clear. In the case of clustering being ambiguous the algorithm needs to be improved, for example by combining it with a cluster-vs-class strategy flexibly. The WS-SVM algorithm may not be really suitable for a multi-class problem. However, it seems to be useful in solving a classification problem with unbalanced data. And this is an interesting research direction in the future.

ACKNOWLEDGMENT

The authors would like to thank the referees for their helpful comments and valuable suggestions.

REFERENCES

[1] G. Fung and O. L. Mangasarian, "Proximal support vector machine," in *KDD '01: Proceedings of the Seventh ACM SIGKDD International Conference On Knowledge Discovery And Data Mining.* San Francisco California: Association for Computing Machinery, New York, NY, United States, 2001, pp. 77–86. [Online]. Available: https://dl.acm.org/doi/10.1145/502512.502527

[2] V. Vapnik, *The Natural Of Statistical Learning Theory.* Springer-Verlag New York, 1995.

[3] M. Adancon and M. Cheriet, "Model selection for the ls-svm. application to handwriting recognition," *Pattern Recognition*, vol. 42, pp. 3264–3270, 2009.

[4] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[5] W. Noble, *Support Vector Machine Applications in Computational Biology.* MIT Press, 2004.

[6] Y. Tian, Y. Shi, and X. Liu, “Recent advances on support vector machines research,” *Technological and Economic Development of Economy*, vol. 18, pp. 5–33, 2012.

[7] D. Tomar and S. Agarwal, “Twin support vector machine: A review from 2007 to 2014,” *Egyptian Informatics Journal*, vol. 16, pp. 55–69, 2015.

[8] B. Mei and Y. Xu, “Multi-task least squares twin support vector machine for classification,” *Neurocomputing*, vol. 338, pp. 26–33, 2019.

[9] X. Pan, Y. Luo, and Y. Xu, “K-nearest neighbor based structural twin support vector machine,” *Knowledge-Based Systems*, vol. 88, pp. 34–44, 2015.

[10] X. Xie and S. Sun, “Multitask centroid twin support vector machines,” *Neurocomputing*, vol. 149, pp. 1085–1091, 2015.

[11] Jayadeva, R. Khemchandani, and S. Chandra, “Twin support vector machines for pattern classification,” *IEEE Transactions on Pattern Analysis and Machine intelligence*, vol. 29, pp. 905–910, 2007.

[12] Z. Qi, Y. Tian, and Y. Shi, “Structural twin support vector machine for classification,” *Knowledge-Based Systems*, vol. 43, pp. 74–81, 2013.

[13] H. Xue, S. Chen, and Q. Yang, “Structural regularized support vector machine: A framework for structural large margin classifier,” *IEEE Transactions on Neural Networks*, vol. 22, pp. 573–587, 2011.

[14] O. Mangasarian and E. Wild, “Multisurface proximal support vector classification via generalized eigenvalues,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, pp. 69–74, 2006.

[15] D. Jeung, D. Wang, W. Wing, E. Tsang, and X. Wang, “Structured large margin machines: sensitive to data distributions,” *Machine Learning*, vol. 68, pp. 171–200, 2007.

[16] B. Schoelkopf and A. Smola, *Learning with Kernel.* MIT Press, 2002.

[17] G. Golub and C. Van Loan, *Matrix Computations.* The John Hopkins University Press, 1996.

[18] N. Cuong, *Python code.* [Online]. Available: https://github.com/makeho8/python/

[19] *UCI Machine Learning Repository*, Center for Machine Learning and Intelligent Systems at the University of California, Irvine. [Online]. Available: http://archive.ics.uci.edu/ml/machine-learning-databases/

Received on August 20, 2020
Accepted on February 19, 2021