WS-TWSVM: Weighted Structural Twin Support Vector Machine by local and global information

Ramin Rezvani-KhorashadiZadeh
Computer Department, Engineering Faculty, Ferdowsi University of Mashhad (FUM), Mashhad, Iran,
raminrezvani@stu-mail.um.ac.ir

Reza, Monsefi
Computer Department, Engineering Faculty, Ferdowsi University of Mashhad (FUM), Mashhad, Iran, Center of Excellence on Soft Computing and Intelligent Information Processing (SCIIP),
monsefi@um.ac.ir

***Abstract*— Recently many researches have published their papers on training a classifier based on the structural information of data. A Structural Twin Support Vector Machine (S-TWSVM) was proposed to introduce and balance all structural information of both intra-class and inter-class into its optimization problems. In fact, this method neither consider the structural information conflicting between clusters of one class nor the noise data points that can influence on the structure of the data distribution. In this paper, we propose a new Weighted Structural Twin Support Vector Machine (WS-TWSVM) by its local and global information. In our proposed method, we use a weighted (rather than a simple) summing of structural information to sufficiently exploit class's distribution information, so that, applying the density information of data points, the effects of the noise points on the data structure can be handled. Thus, our proposed method, WS-TWSVM can fully exploit the prior knowledge more efficiently than S-TWSVM leads to improve the model's generalization capacity. As been shown in the experiments, WS-TWSVM is superior to S-TWSVM in term of classification accuracy.**

Keywords: Structural Twin Support Vector Machine, Structural information, Density information, SVM, k-NN

I. INTRODUCTION

In the last decade, as the most popular classifier among margin classifiers, SVM [1, 12] have become a powerful tool for pattern classification. The main goal of SVM is to separate two classes of examples by finding support vectors and determine an optimal separating hyperplane between two classes. It aims to maximize margin by paying more attention to margin data points (support vectors) between classes and less attention to prior intrinsic structural information within the classes. In real-world problems, different classes have different structural information, so as an optimal classifier it should consider the structural information within the classes.

In recent years, many improvements on SVM have been published. To speed up the execution of SVM, twin support vector machine [5, 15, 21] and its extensions such as weighted twin support vector machines with local information and its application [20], K-nearest neighbor-based weighted twin support vector regression [19] and KNN-based weighted rough ν-twin support vector machine [22] have been developed. There are also some variations of SVM based on the structural information [2-4, 13]. One of which is Structural Twin Support Vector Machine (S-TWSVM) algorithm [4]. With applying the ideas of TWSVM and SRSVM [3, 14] methods, S-TWSVM proposed a new classifier based on the structural information within classes. By using clustering techniques, S-TWSVM extracts the structural information and introduces this information into its optimization problems, to make a better classifier.

In this paper, we use the advantages of TWSVM, SRSVM and kNN [6, 16] and propose a new Weighted Structural Twin Support Vector Machine by local and global information (WS-TWSVM). Similar to S-TWSVM, our proposed method exploits structural information by using clustering methods. WS-TWSVM emphasizes on the structural information obtained by support clusters that involve the support vectors, and then effectively introduces this information into the models of WS-TWSVM. On the other side, by using TWSVM and kNN methods, WS-TWSVM (our proposed algorithm) pays more attention to examples placed in higher density areas and less attention to data in other areas within each class.

WS-TWSVM also has been enhanced by the following features:-

Is an extension of S-TWSVM that cooperate kNN method to better exploit the distribution information of data.

Pays more attention to support clusters and dense data points. This makes our proposed method to have the following properties:

1) Improving the robustness against noise and outliers as low density data.

2) Dealing effectively with large data with many clusters within positive and negative classes and better introducing the structural information within classes to the model.

3) To gain the ability of dealing with confliction between the structural information of clusters within one class and effectively introducing the structural information of data to the model.

4) Paying less attention to outliers which leads to increase the model's generalization and flexibility.

The remaining parts are organized as follows. Section 2 introduces S-TWSVM, and Section 3 explains the details of WS-TWSVM. All experiments are shown in section 4, and finally conclusions in the last section.

978-1-4673-9280-8/15/$31.00 ©2015 IEEE

II. BACHGROUND

Suppose the training data $(\mathbf{X}_i, y_i) \in (R^d \times Y)^l$ such that $\mathbf{X}_i \in R^d, y_i \in Y, i = 1, ..., l, Y = \{1, -1\}$.

The positive class (P) has C_p clusters and the negative (N) has C_N clusters. So we have $P = \bigcup_{i=1}^{C_p} P_i$, $N = \bigcup_{j=1}^{C_N} N_j$

A. S-TWSVM

S-TWSVM has two steps. As the first step it extracts the structural information within classes by one of many clustering methods. Ward's linkage clustering (WIL) [7], as hierarchical clustering analysis, has been used in this step. Learning the model is implemented next. S-TWSVM model for positive class can be formulated as

$$\begin{aligned} \underset{w_1, b_1, \xi}{Min} \ & \frac{1}{2}\|AW_1 + e_1 b_1\|_2^2 + c_1 e_2^T \xi + \frac{1}{2} c_2 \left(\|W_1\|_2^2 + b_1^2\right) + \frac{1}{2} c_3 W_1^T \Sigma_1 W_1 \\ s.t. \ & -\left(BW_1 + e_2 b_1\right) + \xi \geq e_2 \ , \ \xi \geq 0 \end{aligned} \tag{1}$$

and for negative class;

$$\begin{aligned} \underset{w_2, b_2, \eta}{Min} \ & \frac{1}{2}\|BW_2 + e_2 b_2\|_2^2 + \frac{1}{2} c_4 e_1^T \eta + \frac{1}{2} c_5 \left(\|W_2\|_2^2 + b_2^2\right) + \frac{1}{2} c_6 W_2^T \Sigma_2 W_2 \\ s.t. \ & \left(AW_2 + e_1 b_2\right) + \eta \geq e_1 \ , \ \eta \geq 0 \end{aligned} \tag{2}$$

where $\{c_t \geq 0 \,|\, t = 1, ..., 6\}$ are the pre specified penalty coefficients, e_1, e_2 are vectors of appropriate dimensions of ones, ξ_i as slack variables, $\Sigma_1 = \Sigma_{P_1} + ... + \Sigma_{P_{C_P}}, \Sigma_2 = \Sigma_{N_1} + ... + \Sigma_{N_{C_N}}$ where Σ_{P_i} denotes the covariance matrix of i[th] cluster in positive class.

III. THE PROPOSED METHOD (WEIGHTED STRUCTURAL TWIN SUPPORT VECTOR MACHINE BY LOCAL AND GLOBAL INFORMATION (WS-TWSVM))

In this section 4 steps of our proposed method are given:-

A. Data Clustering and Structural Information Extraction

In this step, the training examples are clustered. In order to compare our proposed method to S-TWSVM, the same clustering method, i.e., Ward's linkage clustering (WIL), is adopted.

Consider two clusters A_1 and A_2 , Ward's linkage $W(A_1, A_2)$ between these two clusters is expressed as:

$$W(A_1, A_2) = \frac{|A_1| . |A_2| . \|\mu_{A_1} - \mu_{A_2}\|^2}{|A_1| + |A_2|} \tag{3}$$

with the means μ_{A_1} and μ_{A_2} for two clusters respectively.

At first iteration in the clustering algorithm, each sample is considered as one cluster. Suppose a_1 and a_2 are two examples then the Ward's linkage is:

$$W(a_1, a_2) = \frac{\|a_1 - a_2\|^2}{2} \tag{4}$$

when a_1 and a_2 are being merged to construct A', the Ward's linkage for A' and B is expressed as

$$W(A', B) = \frac{(|a_1| + |B|)W(a_1, B) + (|a_2| + |B|)W(a_2, B) - |B|W(a_1, a_2)}{|a_1| + |a_2| + |B|}. \tag{5}$$

While the clusters are being merged, the ward's linkage between them increases and the number of clusters decreases [2]. So when the knee point which lies in the curve between the merge distance and the number of clusters, is found, the clustering process should be stopped, and the optimal number of clusters is determined [8].

B. Determining importance coefficient for each cluster (Global information)

Here, for each cluster we give an importance coefficient. This is one of the strength of our proposed method. Importance coefficient for each cluster in positive class can be determined as

1) Initially importance coefficients are set to 1 for all clusters.
2) For each of the negative class example, kNN algorithm (with the neighborhood size k_1) is executed over all the points of positive class, looking for neighbors of the inspected negative point in positive class. Hence we have a list of points of positive class (support vectors).
3) For each point in the list, the cluster (support cluster) that involves the point is determined and the importance coefficient for this cluster is incremented by 1.

The importance coefficient for each positive cluster is been determined such that the coefficients for support clusters (the clusters placed near the opposite class) are higher than others in the same class. At the end, each importance coefficient is been normalized to the range [0, 1]. For the negative class we apply the same operations.

The covariance of positive class can be formulated as

$$\Sigma_{+} = \lambda_1 \Sigma_{P_1} + \lambda_2\, \Sigma_{P_2} + \lambda_3 \Sigma_{P_3} + \ldots + \lambda_{CP} \Sigma_{P_{CP}} \tag{6}$$

and the covariance of negative class can be obtained by

$$\Sigma_{-} = \gamma_1 \Sigma_{N_1} + \gamma_2\, \Sigma_{N_2} + \gamma_3 \Sigma_{N_3} + \ldots + \gamma_{CN} \Sigma_{N_{CN}} \tag{7}$$

where λ_i is the importance coefficient of the i[th] positive cluster and γ_j is for j[th] negative cluster.

Giving weight to the support clusters leads to differently consider the clusters in one class. As the support vectors which lie loser to opposite class in SVM, are more important than other data points, the clusters near the other class should have more effect on the structure information of corresponding class. So we give higher importance coefficient to the covariance matrices of these clusters than others. Therefore the hyperplane of each class can better follow the distribution trend of samples in sensitive areas and improves the classification performance as seen in Fig. 2.

C. *Density extraction of points (Local information)*

S-TWSVM has considered neither the difference between the samples in one class nor the difference between the clusters extracted in each class. But in WS-TWSVM, we show how these two types of difference could be introduced into the optimization problems. For the cluster difference, we distinguish between clusters according to their proximity to the other class which has mentioned above, and for the samples difference, we recognize which sample in one class is more important than others according to the density of its location. Here we explain how the sample difference is calculated and introduced to the model.

As we know, density of point x in one class, has been defined as the number of points that the point x is appeared at the neighborhood of them. To determine density coefficient d_j that corresponds to point x_j in positive class, kNN algorithm (with the neighborhood size k_2) is executed for every positive point x_i such as:

$$d_j = \Sigma_{i=1}^{N_1} W_{ij},\ j = 1,2,\ldots,N_1 \tag{8}$$

where d_j is the number of points in positive class (with i=1..N_1 points) so that x_j appears in the neighborhood of x_i. If $W_{ij} = 1$, the point x_j is at the neighborhood of point x_i. N_1 is the number of examples in positive class. Initially, W_{ij} for each pair of points is zero. When samples are placed in more dense area, d_j for them is higher than other density coefficients. For data points in more sparse area, this coefficient become smaller than others, so our proposed method pays more attention to area with higher density and less attention to sparse area (where noise data usually appears in). This idea has also been employed in KNN-STSVM [18] to adjust the hyperplane of the corresponding class such that each hyperplane should lie near the dense points. But in WS-TWSVM by adding the global information to this idea the hyperplanes can better be adjusted and a better decision hyperplane is obtained.

D. *Model learning*

In this section, the linearly non-separable case of WS-TWSVM is been shown.

Suppose that P and N clusters exist in two classes of positive and negative respectively,

$$P = \bigcup_{i=1}^{C_p} P_i\ ,\ N = \bigcup_{j=1}^{C_N} N_j$$

Positive class's samples are represented as $A \in R^{m_1 \times d}$ and $B \in R^{m_2 \times d}$, shows all samples that exists in negative class, where $m_1 + m_2 = l$. WS-TWSVM determines two nonparallel hyperplanes in linear case:

$$f_1(x) = w_1^T x + b_1 = 0\ ,\ f_2(x) = w_2^T x + b_2 = 0 \tag{9}$$

where $w_1, w_2 \in R^d, b_1, b_2 \in R$.

WS-TWSVM model can be formulated as:-

$$\begin{aligned} \underset{w_1,b_1,\xi}{Min}\ & \frac{1}{2} D \left\| AW_1 + e_1 b_1 \right\|_2^2 + c_1 e_2^T \xi + \frac{1}{2} c_2 \left(\left\| W_1 \right\|_2^2 + b_1^2 \right) + \frac{1}{2} c_3 W_1^T \Sigma_1 W_1 \\ s.t.\ & -\left(BW_1 + e_2 b_1 \right) + \xi \geq e_2\ ,\ \xi \geq 0 \end{aligned} \tag{10}$$

where $d_j = \Sigma_{i=1}^{N_1} W_{ij}, j = 1,2,\ldots,N_1$ and N_1 is the number of points in positive class, D denotes diagonal matrix form of d_j.

$$\begin{aligned} \underset{w_2,b_2,\eta}{Min}\ & \frac{1}{2} K \left\| BW_2 + e_2 b_2 \right\|_2^2 + \frac{1}{2} c_4 e_1^T \eta + \frac{1}{2} c_5 \left(\left\| W_2 \right\|_2^2 + b_2^2 \right) + \frac{1}{2} c_6 W_2^T \Sigma_2 W_2 \\ s.t.\ & \left(AW_2 + e_1 b_2 \right) + \eta \geq e_1\ ,\ \eta \geq 0 \end{aligned} \tag{11}$$

where $k_j = \Sigma_{i=1}^{N_2} W_{ij}, j = 1,2,\ldots,N_2$ and N_2 is the size of negative samples, K denotes diagonal matrix form of k_j and $\{c_t \geq 0 \,|\, t = 1,\ldots,6\}$ as penalty coefficients. e_1, e_2 and ξ_i are the same with that of S-TWSVM.

$\Sigma_1 = \lambda_1 \Sigma_{P_1} + \lambda_2\, \Sigma_{P_2} + \lambda_3 \Sigma_{P_3} + \ldots + \lambda_{C_P} \Sigma_{P_{C_p}}$, $\Sigma_2 = \gamma_1 \Sigma_{N_1} + \gamma_2\, \Sigma_{N_2} + \gamma_3 \Sigma_{N_3} + \ldots + \gamma_{C_N} \Sigma_{N_{C_N}}$ where Σ_{P_i} denotes the covariance matrix of i[th] cluster in positive class.

For the problem (10) its wolfe-dual is expressed as:

$$\underset{\alpha}{Max}\ e_2^T\alpha-\frac{1}{2}\alpha^T G(H^T DH+c_2I+c_3J)^{-1}G^T\alpha \tag{12}$$
$$s.t.\ \ 0\le\alpha\le c_1e_2$$

where

$$H=[A\,e_1],G=[B\,e_2],J=\begin{bmatrix}\text{å}_1 & 0\\ 0 & 0\end{bmatrix}, \tag{13}$$

and the vector $v_1=[w_1^T\ b_1^T]^T$ is given by

$$v_1=-(H^T DH+c_2I+c_3J)^{-1}(G^T\alpha) \tag{14}$$

I is an identity matrix of appropriate dimensions. It can be proved in [9] that $H^T DH+c_2I+c_3J$ is a positive definite matrix. Similarly, for the problem (11) we have:

$$\underset{\beta}{Max}\ e_1^T\beta-\frac{1}{2}\beta^T P(Q^T KQ+c_5I+c_6F)^{-1}P^T\beta \tag{15}$$
$$s.t.\ \ 0\le\beta\le c_4e_1$$

where

$$P=[A\,e_2],F=\begin{bmatrix}\text{å}_2 & 0\\ 0 & 0\end{bmatrix},Q=[B\,e_1] \tag{16}$$

and the vector $v_2=[w_2\ b_2]^T$ is:

$$v_2=-(Q^T KQ+c_5I+c_6F)^{-1}P^T\beta, \tag{17}$$

such that $Q^T KQ+c_5I+c_6F$ is a positive definite matrix. By obtaining two vectors v_1 and v_2, the separating hyperplanes are constructed as follow:

$$W_1^T x+b_1=0,\ W_2^T x+b_2=0. \tag{18}$$

The new sample $x_t\in R^d$ is classified according to its proximity to the two hyperplanes, i.e.,

$$f(x_t)=\arg\min_{1,2}\{d_1(x_t),d_2(x_t)\}, \tag{19}$$

where

$$d_1(x)=\left|W_1^T x+b_1\right|,\ d_2(x)=\left|W_2^T x+b_2\right|, \tag{20}$$

consider $|.|$ as the perpendicular distance from one hyperplane of (18) to sample x.

IV. EXPERIMENTS

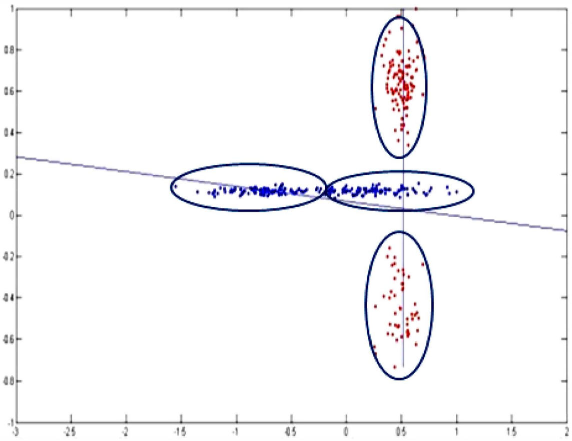
In this section, two algorithms WS-TWSVM and S-TWSVM are tested on different data sets. In all experiments, for simplicity we set $c_1=c_4$, $c_2=c_5$, $c_3=c_6$, $k_1=k_2$. Accuracy of testing for all experiments is calculated using ten-fold cross validation procedure and this procedure is employed to select the parameters c_1, c_2, c_3, from $\{2^{-7},2^{-6},\ldots,2^{0},\ldots,2^{7}\}$ on 10 % of training samples. Also neighborhood size k_1 is selected from the set $[2,3,\ldots,8]$. For measuring both accuracy and CPU time of S-TWSVM and WS-TWSVM, we employed a Matlab toolbox as Gunn SVM toolbox which is developed in [5,10].

A. Toy data

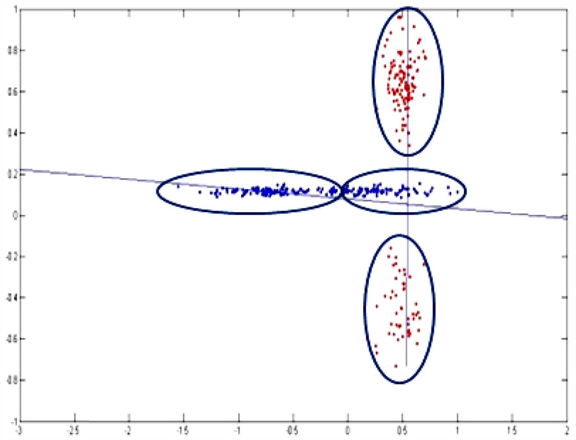
For a visual representation of WS-TWSVM performance and compare it with the S-TWSVM, a two dimension synthetic data which is a typical XOR dataset [3] is used. In this paper, two experiments on these data sets are implemented with 50 % of data for training, 50 % for testing and 10% of training data is used for parameter selection procedure. In both experiments we have two horizontal and two vertical distributions in positive and negative class respectively.

In first experiment seen in Fig. 1, the distribution structure of two positive clusters is quietly horizontally and they support each other. So as seen in Fig. 1, both algorithms S-TWSVM and WS-TWSVM act in the same way.

In the second experiment, seen in Fig. 2, the distribution structure of right side positive cluster is quietly horizontally and the left side is more scattered caused the structural information obtained by these two positive clusters being conflicted. In this situation that the structural information of clusters in one class is conflicted, S-TWSVM accuracy decreased and only summing the structural information of clusters could not be very useful to exploit the structural information of classes. In WS-TWSVM, support clusters are identified and the structural information of them will be more important than others, and thus by weighted summing rather than simple summing, the problem of conflictive between the structural information of clusters in one class, can be solved. On the other hand, by using density coefficients, the noise data will not so effect in the learning process, thus the distribution information of data can be fully exploited. As seen in Fig. 2, WS-TWSVM algorithm performs better than S-TWSVM and has higher generalization capacity, constructs more reasonable classifier.

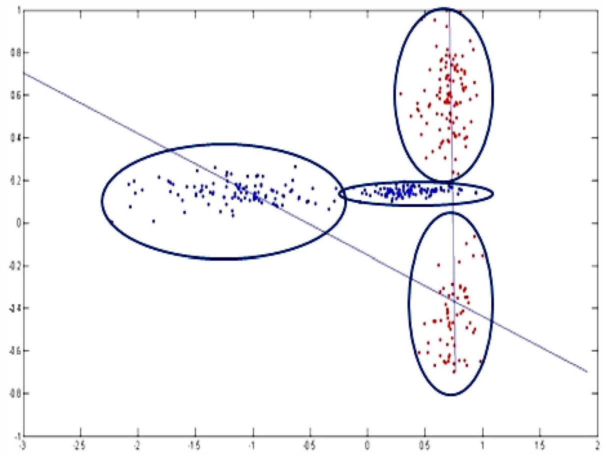


(a) S-TWSVM

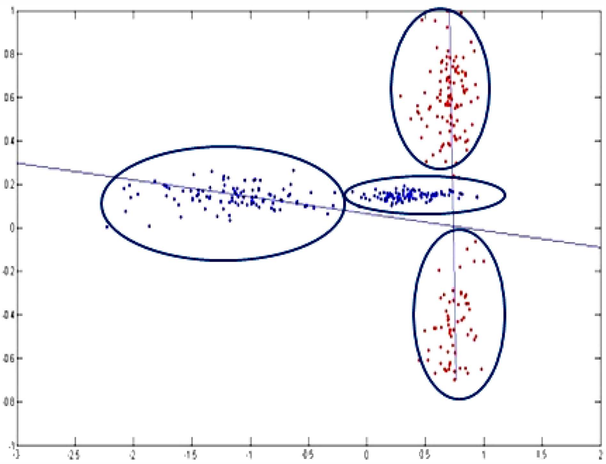


(b) WS-TWSVM

Fig. 1. The results of S-TWSVM and WS-TWSVM on synthetic data. The positive and negative samples are represented by the blue and red points respectively. In positive class, two distribution structures support each other and WS-TWSVM and S-TWSVM act in the same way.



(a) S-TWSVM



(b) WS-TWSVM

Fig. 2. The results of S-TWSVM and WS-TWSVM on synthetic data. The positive and negative samples are represented by the blue and red points respectively. Structure distribution of one positive cluster is quiet horizontal whereas the other is more scattered. This causes the structural information obtained by these two positive clusters to have conflict. In this respect, WS-TWSVM acts better than S-TWSVM.

B. UCI datasets

We apply S-TWSVM, KNN-STSVM and our proposed method on UCI datasets [11]. Fifty percent of data are used for training and 50% for testing. Model's parameters selection is performed by the method of ten-fold cross validation over 10% of the training set. All data are normalized to the range [0,1]. We can obtain the following conclusions from the experimental results in Table 1:-

1) Both algorithms WS-TWSVM and KNN-STWSVM obtain better performance than S-TWSVM. This is because they consider the density information of samples in their models and also in WS-TWSVM, the structural information of clusters is better introduced to its optimization problems.

2) With the comparison of two algorithms WS-TWSVM and KNN-STSVM, we can conclude that by adding the global information to KNN-STSVM, the corresponding hyperplanes can better exploit the structural information and WS-TWSVM can better handle the structural information conflicting between clusters within a class and improve the classification performance.

Table 1. The results of WS-TWSVM and S-TWSVM on UCI datasets.

Datasets	WS-TWSVM Acc. Exe. (s)	KNN-STSVM Acc. Exe. (s)	S-TWSVM Acc. Exe. (s)
Australian (690×14)	**86.06 ± 0.71** 2.97	85.92 ± 0.53 1.87	85.70 ± 0.62 2.97
BUPA liver (345×6)	**61.15 ± 0.64** 1.71	61.11 ± 0.66 1.53	60.58 ± 0.57 1.67
Diabetis (768×8)	74.64 ± 0.29 6.11	**74.70 ± 0.24** 5.03	61.56 ± 1.50 6.07
German (1000×20)	**69.87 ± 0.41** 2.81	69.85 ± 0.42 2.73	69.73 ± 0.33 2.76
Heart-Statlog (270×14)	**81.94 ± 0.42** 7.69	79.35 ± 0.32 4.93	78.61 ± 0.82 5.66
Ionosphere (351×34)	**79.10 ± 0.51** 7.12	77.46 ± 0.47 6.42	77.24 ± 1.28 6.44

V. CONCLUSIONS

The S-TWSVM algorithm leads to avoid the structural information conflicting between two classes and this doesn't guarantee that in the S-TWSVM, conflicting between the structural information of clusters in one class doesn't occur. On the other hand, the noise data points which can influence on the structure of the data distribution, is not considered in the S-TWSVM. In order to overcome these shortcomings, we proposed a Weighted Structural Twin Support Vector Machine by local and global information (called WS-TWSVM) in this paper. We proposed a solution to handle the structural information conflicting between the clusters in one class as the weighted summing of the structural information of clusters in one class. To avoid the effects of the noise data on the data structure, density information of points were used. Theoretical analysis and results show WS-TWSVM can better exploit this structural information. Moreover, our proposed algorithm is superior to the S-TWSVM and KNN-STSVM in term of the classification accuracy. A limitation of WS-TWSVM is that it must find the k- nearest neighbors for all the points, thus the selection of the parameter k is under our consideration. In addition, how to further accelerate the algorithm is also a topic in the future.

VI. References

1. **C. Cortes, V.N. Vapnik**, Support-vector networks, Machine Learning, 273–297, 1995.
2. D**. Yeung, D. Wang, W. Ng, E. Tsang, X. Wang**, "Structured large margin machines: sensitive to data distributions", Machine Learning 171–200 **2007**.
3. ***H. Xue, S. Chen, Q. Yang**, Structural regularized support vector machine: a framework for structural large margin classifier, IEEE Transactions on Neural Networks, 573–587, 2011.*
4. **Z. Qi, Y. Tian, Y. Shi**, Structural twin support vector machine for classification, Knowledge-Based Systems, 2013.
5. ***Jayadeva, R. Khemchandani, S. Chandra**, Twin support vector machines for pattern classification, IEEE Transactions on Pattern Analysis, 2007.*
6. ***Cover, T. M., & Hart, P. E.**, Nearest neighbor pattern classification, IEEE Transactions on Information Theory, 1967.*
7. ***J. H. Ward**, Hierarchical grouping to optimize an objective function, Journal of the American Statistical Association, 1963*
8. ***S. Salvador, P. Chan**, Determining the number of clusters/segments in hierarchical clustering/ segmentation algorithms, In Proceedings of the 16th IEEE international conference on tools with AI, 2004.*
9. *F.R. Gantmacher, Matrix Theory,1990.*
10. ***Jayadeva, R. Khemchandani, S. Chandra**, Fuzzy multi-category proximal support vector classification via generalized eigenvalues, Soft Computing, 2007.*
11. **A. Asuncion, D. Newman,** UCI Machine Learning Repository, 2007.
12. ***N. Cristianini and J. Shawe-Taylor**, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods Cambridge, U.K Cambridge, 2000.*
13. *K. **Huang, H. Yang, I. King, and M. R. Lyu,** Learning large margin classifiers local and global, in Proc. 21st Int. pp. 1–8., 2004.*
14. ***H. Xue, S. Chen, Q. Yang**, Structural support vector machine, The 15th International Symposium on Neural Networks, 2008.*
15. *Y**.-H. Shao, C.-H. Zhang, X.-B. Wang, N.-Y. Deng,** Improvements on twin support vector machines, IEEE Transactions on Neural Networks, 2011.*
16. ***Y. Wu, K. Ianakiev & V. Govindaraju,** Improved k-nearest neighbor classification. Pattern Recognition, 2002.*
17. **Y. LeCun, L. Bottou, Y. Bengio, P. Haffner,** Gradient-based learning applied to document recognition, Proceedings of the IEEE, 2278–2344, 1998.
18. **X. Pan, Y. Luo, Y. Xu,** K-nearest neighbor based structural twin support vector machine, Knowledge-Based Systems, 2015.
19. **Y. XU, L. Wang**, K-nearest neighbor-based weighted twin support vector regression, Applied intelligence, 299-309, 2014,
20. Q. Ye, C Zhao, S.Gao, H. Zheng, Weighted twin support vector machines with local information and its application, Neural Networks, 31-39, 2012.
21. D. Tomar, S. Agarwal, Twin support vector machine: a review from 2007 to 2014, Egyptian Informatics, 2015.
22. Y. Xu, J. Yu, Y. Zhang, KNN-based weighted rough ν-twin support vector machine, 2014.