Coordinate Descent Fuzzy Twin Support Vector Machine for Classification

Bin-Bin Gao, Jian-Jun Wang, Yao Wang, Chan-Yun Yang

***Abstract*—In this paper, we develop a novel coordinate descent fuzzy twin SVM (CDFTSVM) for classification. The proposed CDFTSVM not only inherits the advantages of twin SVM but also leads to a rapid and robust classification results. Specifically, our CDFTSVM has two distinguished advantages: (1) An effective fuzzy membership function is produced for removing the noise incurred by the contaminant inputs. (2) A coordinate descent strategy with shrinking by active set is used to deal with the computational complexity brought by the high dimensional input. In addition, a series of simulation experiments are conducted to verify the performance of the CDFTSVM, which further supports our previous claims.**

***Keywords*—Coordinate descent, Fuzzy, Active set shrinking, High-dimensional input, TSVM, SVM.**

I. Introduction

SUPPORT vector machine (SVM),invented by Vapnik [1], is a great method for machine learning. At present, its applications have been largely expanded. Prominent examples include machine fault diagnosis [2], image identification [3], text classification [4], and more.

As far as SVM itself is concerned, there has been many variants in literature. twin support vector machine (TSVM), as one of the most successful variants, in recent years has caused much attention and been widely studied. TSVM originates the generalized eigenvalue proximal support vector machine (GEPSVM), the work of Mangasarian and Wild [5] in 2006. The main idea of GEPSVM is to replace two parallel hyperplanes with two nonparallel ones. Following this concept, Jayadeva *et al*. [6] proposed the TSVM in 2007. Different from the GEPSVM involving a set of generalized eigenvalue problems, the induced optimal problems in TSVM consist of two quadratic programming problems which all share the formulation of a typical SVM but is about 4 times faster than SVM with almost identical accurate performance. Since then, a series of improvements on TSVM have been produced. such as Least square TSVM [7], Structural TSVM [8], Robust TSVM [9], Laplacian smooth TSVM [10] *etc*. One can refer to the survey paper [11] for more.

Conventional SVM intrinsically treats every input sample in equivalence. However, when the samples are in low quality or polluted by additional noise, both SVM and its most variants often lead to a poor generalization performance. In order to tackle such problem, a category of fuzzy support vector machines are hence developed, such as Lin and Wang [12], [13], Tang [14] and Yang *et al*. [15] *etc*. The elementary concept of fuzzy SVM is to allocate a small active or passive confident membership to each input sample consistent with the "fuzziness" which the sample has carried to reduce its influence on the optimization. The membership is generally assigned according to the sample's confidence intrinsically related to its native class. The introduction of fuzzy membership reduces effectively the uncertainty caused by the sample noise and leads to a robust classifier.

Here, we have noticed that there is a chance to connect the concept of TSVM and fuzzy membership for pursuing both the computational efficiency and the robust performance. Similar to the conventional SVM, TSVM is also limited by the noise deterioration. Due to the reason, we seek a method to firmly embed the fuzzy concept into TSVM in this study. In addition, we employs further the coordinate descent method to speed-up the computations of the union of TSVM and FSVM. Thus, a novel CDFTSVM is proposed in this paper. Numerical experiments show that the proposal brings more satisfactory effects on both classification accuracy and computational time, compared with the original stumps.

II. Background

With a set of n-dimensional l training samples, a dataset $T = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)|\boldsymbol{x}_i \in R^n, i = 1, 2, \cdots, l\}$ is given first as a composition of input samples $\boldsymbol{x}_i$ and their class labels $\boldsymbol{y}_i$. This paper main focuses on a binary classification problem, so we assume $\boldsymbol{y}_i = \{+1, -1\}$[1]$(i = 1, \cdots, l)$. With the $+1/-1$ labels, the training set T is then divided into the $l_+ \times n$ dimensional matrix X_+ and $l_- \times n$ dimensional matrix X_- for positive and negative classes, respectively, where l_+ and l_- denote the number of samples in the positive and negative classes, respectively. The aggregations $\boldsymbol{X} = [X_+^{\mathrm{T}}\ X_-^{\mathrm{T}}]^{\mathrm{T}}$ denote the whole set of input matrix of T.

A. Fuzzy Support Vector Machine

To tackle the difficulty of producing unambiguously a generalized separating hyperplane with the uncertainty from the noisy samples which are neighboring around the decision boundary, fuzzy numbers, $0 \leq \boldsymbol{s}_i \leq 1, i = 1, 2, \cdots, l$, carrying additional information to reflect the noisy contaminated level

B.B Gao is with Department of Computer Science and Technology, Nanjing University, Nanjing 210023, P.R. China. (e-mail: gaobb@lamda.nju.edu.cn).

J.J Wang is with School of Mathematics and Statistics, Southwest University, Chongqing 400715, P.R. China. (Corresponding author, e-mail: wjj@swu.edu.cn).

Y. Wang is with School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, P.R. China. (e-mail: yao.s.wang@gmail.com).

C.Y Yang is with Department of Electrical Engineering, National Taipei University, New Taipei City 23741, Taiwan. (e-mail: cyyang@mail.ntpu.edu.tw).

[1]Classification label +1 and -1 sometimes will be abbreviated as + and - when misunderstandings are not caused.

of the samples are introduced. The input dataset T is thus modified as $T' = (\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{s}_i)$. These fuzzy memberships $\boldsymbol{s}_i$'s are used to reduce the influence of the contaminated samples for generating the decision function, and to achieve the classifier more robust to the contamination, which induces the fuzzy SVM as follow:

$$\begin{aligned} &\min_{\boldsymbol{w},b,\xi} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\boldsymbol{s}^{\mathrm{T}}\boldsymbol{\xi} \\ &\text{s.t.} \ \ \boldsymbol{y}_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) + \boldsymbol{\xi}_i \geq 1, \boldsymbol{\xi}_i \geq \boldsymbol{0}, i = 1, 2, \cdots, l. \end{aligned} \tag{1}$$

Here, C denotes a scalar whose value determines the trade-off and $\boldsymbol{\xi}_i$ is called as the slack variable which denotes the error variable associated with the ith input sample. The classification of any new input $\boldsymbol{x}$ can be obtained by determining the sign of ${\boldsymbol{w}^*}^{\mathrm{T}}\boldsymbol{x} + b^*$ where $\boldsymbol{w}^*$ and b^* are the solution of (1).

B. Twin Support Vector Machine

Different from the conventional SVM, TSVM is in fact constructed by two nonparallel decision planes, i.e.

$$\boldsymbol{w}_+^{\mathrm{T}}\boldsymbol{x} + b_+ = 0 \text{ and } \boldsymbol{w}_-^{\mathrm{T}}\boldsymbol{x} + b_- = 0 \tag{2}$$

To construct such two nonparallel decision planes, a pair of primal optimization problems are set up:

$$\begin{aligned} \min_{\boldsymbol{w}_+,b_+,\boldsymbol{\xi}_-} \quad & \frac{1}{2} \parallel X_+\boldsymbol{w}_+ + \boldsymbol{e}_+b_+ \parallel^2 + C_1\boldsymbol{e}_-^{\mathrm{T}}\boldsymbol{\xi}_- \\ \text{s.t.} \quad & -(X_-\boldsymbol{w}_+ + \boldsymbol{e}_-b_+) + \boldsymbol{\xi}_- \geq \boldsymbol{e}_-, \boldsymbol{\xi}_- \geq \boldsymbol{0} \end{aligned} \tag{3}$$

and

$$\begin{aligned} \min_{\boldsymbol{w}_-,b_-,\boldsymbol{\xi}_+} \quad & \frac{1}{2} \parallel X_-\boldsymbol{w}_- + \boldsymbol{e}_-b_- \parallel^2 + C_2\boldsymbol{e}_+^{\mathrm{T}}\boldsymbol{\xi}_+ \\ \text{s.t.} \quad & (X_+\boldsymbol{w}_- + \boldsymbol{e}_+b_-) + \boldsymbol{\xi}_+ \geq \boldsymbol{e}_+, \boldsymbol{\xi}_+ \geq \boldsymbol{0} \end{aligned} \tag{4}$$

where $C_1 > 0$ and $C_2 > 0$ are parameters, $\boldsymbol{\xi}_+$ and $\boldsymbol{\xi}_-$ denote the vectors of slack variables for positive and negative classes, respectively, and $\boldsymbol{e}_+, \boldsymbol{e}_-$ correspond to unit row vectors with their dimensions exact to sample sizes in positive and negative classes.

If the solutions to (3) and (4) are obtained respectively as $(\boldsymbol{w}_+^*, b_+^*)$ and $(\boldsymbol{w}_-^*, b_-^*)$, TSVM then can easily label a new input sample $\boldsymbol{x}$ by

$$f(\boldsymbol{x}) = \arg\min_{\pm} \frac{\mid {\boldsymbol{w}_\pm^*}^{\mathrm{T}}\boldsymbol{x} + b_\pm^* \mid}{\parallel \boldsymbol{w}_\pm^* \parallel}. \tag{5}$$

III. Fuzzy Twin Support Vector Machine

In this part, we will present the model of FTSVM. To this end, we introduce the fuzzy membership assignment first.

A. Fuzzy Membership Assignment

Fuzzy membership plays a key role in robust classification learning. However there is no unified standard to construct such fuzzy membership so far. Inspired by [14], we in our paper present a fuzzy membership assignment for training samples. Since the definition of fuzzy membership for both positive and negative class can be obtained similarly, here we just take the fuzzy membership for positive class for example.

The positive class centers $\boldsymbol{\varphi}_{pcen}$ in the feature space $\mathscr{H}$ is first defined as:

$$\boldsymbol{\varphi}_{pcen} = \frac{1}{l_+}\sum_{j=1}^{l_+}\varphi(\boldsymbol{x}_j), \ for \ \boldsymbol{x}_j \in X_+,$$

where $\varphi(\boldsymbol{x}_j) \in \mathscr{H}$ denotes the transformation of an arbitrary input data point $\boldsymbol{x}_j$.

Then the scattering hypersphere radii can be obtained by

$$r_{\varphi+} = \max \parallel \varphi(\boldsymbol{x}_j) - \boldsymbol{\varphi}_{pcen} \parallel, \ for \ \boldsymbol{x}_j \in X_+.$$

With above preparations, the fuzzy membership function can be established as:

$$\boldsymbol{s}_{i+} = \begin{cases} \mu(1 - \sqrt{\parallel \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{pcen} \parallel^2/(r_{\varphi+}^2 + \delta)}), \\ \quad if \ \ \parallel \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{pcen} \parallel \geq \parallel \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{ncen} \parallel \\ (1-\mu)(1 - \sqrt{\parallel \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{pcen} \parallel^2/(r_{\varphi+}^2 + \delta)}), \\ \quad if \ \ \parallel \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{pcen} \parallel < \parallel \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{ncen} \parallel \end{cases}$$

where $\delta > 0$ is defined as a small constant which avoids the vanishing of $\boldsymbol{s}_{i+}$, and μ is a propose constant within [0, 1].

B. Linear FTSVM

Considering the crucial trade-off balance between the margin maximization and error minimization, a margin term, similar to that in the standard SVM, should be added first. Since TSVM has two proximal decision functions, two margin terms $1/\|\boldsymbol{w}_+\|$ and $1/\|\boldsymbol{w}_-\|$ are accordingly defined for the proximal decision functions, respectively. Together with the introduced fuzzy numbers and two discrepant margin terms, a weight regularized model of FTSVM for the linear kernel is hence proposed:

$$\begin{aligned} \min_{\boldsymbol{w}_+,b_+,\boldsymbol{\xi}_-} \quad & \frac{1}{2}C_1 \parallel \boldsymbol{w}_+ \parallel^2 + \frac{1}{2} \parallel X_+\boldsymbol{w}_+ + \boldsymbol{e}_+b_+ \parallel^2 + C_3\boldsymbol{s}_-^{\mathrm{T}}\boldsymbol{\xi}_- \\ \text{s.t.} \quad & -(X_-\boldsymbol{w}_+ + \boldsymbol{e}_-b_+) + \boldsymbol{\xi}_- \geq \boldsymbol{e}_-, \ \boldsymbol{\xi}_- \geq \boldsymbol{0}, \end{aligned} \tag{6}$$

and

$$\begin{aligned} \min_{\boldsymbol{w}_-,b_-,\boldsymbol{\xi}_+} \quad & \frac{1}{2}C_2 \parallel \boldsymbol{w}_- \parallel^2 + \frac{1}{2} \parallel X_-\boldsymbol{w}_- + \boldsymbol{e}_-b_- \parallel^2 + C_4\boldsymbol{s}_+^{\mathrm{T}}\boldsymbol{\xi}_+ \\ \text{s.t.} \quad & (X_+\boldsymbol{w}_- + \boldsymbol{e}_+b_-) + \boldsymbol{\xi}_+ \geq \boldsymbol{e}_+, \ \boldsymbol{\xi}_+ \geq \boldsymbol{0}, \end{aligned} \tag{7}$$

where both $\boldsymbol{s}_+ \in R^{l_+}$ and $\boldsymbol{s}_- \in R^{l_-}$ are the fuzzy-number vectors.

The Wolfe dual of the primal problems (6)-(7) can be easily obtained by using KKT conditions. Here, we present the results as follow:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{e}_-^{\mathrm{T}}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}H_-(H_+^{\mathrm{T}}H_+ + C_1E_1)^{-1}H_-^{\mathrm{T}}\boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{0} \leq \boldsymbol{\alpha} \leq C_3\boldsymbol{s}_-, \end{aligned} \tag{8}$$

$$\begin{aligned} \max_{\boldsymbol{\beta}} \quad & \boldsymbol{e}_+^{\mathrm{T}}\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}H_+(H_-^{\mathrm{T}}H_- + C_2E_2)^{-1}H_+^{\mathrm{T}}\boldsymbol{\beta} \\ \text{s.t.} \quad & \boldsymbol{0} \leq \boldsymbol{\beta} \leq C_4\boldsymbol{s}_+, \end{aligned} \tag{9}$$

where $H_+ = [X_+, \boldsymbol{e}_+]$, $H_- = [X_-, \boldsymbol{e}_-]$, and $E_i = \begin{pmatrix} I & \\ & 0 \end{pmatrix}$ $(i = 1, 2)$. Relationships of the optimal solutions between the primal problems (6)-(7) and their dual problems (8)-(9) are

$$\boldsymbol{u}_+^* = -(H_+^{\mathrm{T}} H_+ + C_1 E_1) H_-^{\mathrm{T}} \boldsymbol{\alpha}^*,\ \boldsymbol{u}_-^* = (H_-^{\mathrm{T}} H_- + C_2 E_2) H_+^{\mathrm{T}} \boldsymbol{\beta}^*$$

where $\boldsymbol{u}_\pm^* = [\boldsymbol{w}_\pm^{*\mathrm{T}}, b_\pm^*]^{\mathrm{T}}$, $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ denote the optimal values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively.

Once (8) and (9) are solved, the non-parallel proximal hyperplanes similar to (2) can thus be subsequently obtained. Then for a new input data point $\boldsymbol{x} \in R^n$, the classification decision function can also be similarly obtained as (5).

Actually, matrices $H_+^T H_+$ and $H_-^T H_-$ are not always non-singular in the dual problems of (8) and (9). To impose on non-singular matrices $H_+^T H_+$ and $H_-^T H_-$, substitutions $H_+^T H_+ + \lambda I$ and $H_-^T H_- + \lambda I$ are made for $H_+^T H_+$ and $H_-^T H_-$ to sustain the non-singularity, where I is a unit matrix with dimensions identical to $H_+^T H_+$ or $H_-^T H_-$, and λ is a small positive real number.

C. Nonlinear FTSVM

In nonlinear case, the dual proximal hyperplanes of FTSVM can be stated as:

$$\kappa(\boldsymbol{x}, X^{\mathrm{T}}) \boldsymbol{w}_+ + b_+ = 0 \text{ and } \kappa(\boldsymbol{x}, X^{\mathrm{T}}) \boldsymbol{w}_- + b_- = 0,$$

where $\kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) = \langle \varphi(\boldsymbol{x}_1), \varphi(\boldsymbol{x}_2) \rangle$ is a kernel function. By employing the fuzzy number s_i, the nonlinear primal problems of a FTSVM can be expressed as:

$$\begin{aligned} \min_{\boldsymbol{w}_+, b_+, \boldsymbol{\xi}_-} \quad & \frac{1}{2} C_1 \parallel \boldsymbol{w}_+ \parallel^2 + \frac{1}{2} \parallel \kappa(X_+, X^{\mathrm{T}}) \boldsymbol{w}_+ + \boldsymbol{e}_+ b_+ \parallel^2 \\ & + C_3 \boldsymbol{s}_-^{\mathrm{T}} \boldsymbol{\xi}_- \\ \text{s.t.} \quad & -(\kappa(X_-, X^{\mathrm{T}}) \boldsymbol{w}_+ + \boldsymbol{e}_- b_+) + \boldsymbol{\xi}_- \geq \boldsymbol{e}_-, \boldsymbol{\xi}_- \geq 0, \end{aligned} \tag{10}$$

$$\begin{aligned} \min_{\boldsymbol{w}_-, b_-, \boldsymbol{\xi}_+} \quad & \frac{1}{2} C_2 \parallel \boldsymbol{w}_- \parallel^2 + \frac{1}{2} \parallel \kappa(X_-, X^{\mathrm{T}}) \boldsymbol{w}_- + \boldsymbol{e}_- b_- \parallel^2 \\ & + C_4 \boldsymbol{s}_+^{\mathrm{T}} \boldsymbol{\xi}_+ \\ \text{s.t.} \quad & (\kappa(X_+, X^{\mathrm{T}}) \boldsymbol{w}_- + \boldsymbol{e}_+ b_-) + \boldsymbol{\xi}_+ \geq \boldsymbol{e}_+, \boldsymbol{\xi}_+ \geq 0. \end{aligned} \tag{11}$$

The Wolfe dual of the primal problems (10) and (11) are

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{e}_-^{\mathrm{T}} \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} S_+ (S_-^{\mathrm{T}} S_- + C_1 E_1)^{-1} S_+^{\mathrm{T}} \boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{0} \leq \boldsymbol{\alpha} \leq C_3 \boldsymbol{s}_-, \end{aligned} \tag{12}$$

$$\begin{aligned} \max_{\boldsymbol{\beta}} \quad & \boldsymbol{e}_+^{\mathrm{T}} \boldsymbol{\beta} - \frac{1}{2} \boldsymbol{\beta}^{\mathrm{T}} S_- (S_+^{\mathrm{T}} S_+ + C_2 E_2)^{-1} S_-^{\mathrm{T}} \boldsymbol{\beta} \\ \text{s.t.} \quad & \boldsymbol{0} \leq \boldsymbol{\beta} \leq C_4 \boldsymbol{s}_+, \end{aligned} \tag{13}$$

where $S_+ = [\kappa(X_+, X^{\mathrm{T}}), \boldsymbol{e}_+]$ and $S_- = [\kappa(X_-, X^{\mathrm{T}}), \boldsymbol{e}_-]$. By designating $\boldsymbol{v}_\pm^* = [\boldsymbol{w}_\pm^{*\mathrm{T}}, b_\pm^*]^{\mathrm{T}}$ for solutions of the primal problems of (10)-(11), there are parametric relationships between the optimal $\boldsymbol{v}_\pm^*$ and the optimal solutions $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ of their corresponding dual forms (12)-(13):

$$\boldsymbol{v}_+^* = -(S_+^{\mathrm{T}} S_+ + C_1 E_1) S_-^{\mathrm{T}} \boldsymbol{\alpha}^*,\ \boldsymbol{v}_-^* = (S_-^{\mathrm{T}} S_- + C_2 E_2) S_+^{\mathrm{T}} \boldsymbol{\beta}^*.$$

Once solutions of the dual problems (12) and (13) are obtained, the decision function for classifying a new data point $\boldsymbol{x} \in R^n$ is eventually given by:

$$f(\boldsymbol{x}) = \arg\min_{\pm} \frac{\mid \kappa(\boldsymbol{x}, X^{\mathrm{T}}) \boldsymbol{w}_\pm^{*\,\mathrm{T}} + b_\pm^* \mid}{\sqrt{\boldsymbol{w}_\pm^{*\mathrm{T}} \kappa(X, X^{\mathrm{T}}) \boldsymbol{w}_\pm^*}}.$$

IV. Speeding-up FTSVM By Coordinate Descent Strategy with active set Shrinking

Our dual FTSVM involves a pair strictly convex QPPs ((8) and (9) or (12) and (13)), but they can be solved similarly. Take for example (8). By denoting $Q = (H_+^{\mathrm{T}} H_+ + C_1 E_1)^{-1} H_-^{\mathrm{T}}$ and $\overline{Q} = H_- Q$, it can be abbreviated as a quadratic expression:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & f(\boldsymbol{\alpha}) = \frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} \overline{Q} \boldsymbol{\alpha} - \boldsymbol{e}_-^{\mathrm{T}} \boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{0} \leq \boldsymbol{\alpha} \leq C_3 \boldsymbol{s}_-. \end{aligned} \tag{14}$$

In order to solve (14), a coordinate descent strategy with active set shrinking is adopted. Because of the limited space, we here omit the specifical theoretical details and interested readers can find more in [16]–[19]. Instead, we present their pseudo-code, which is exhibited in our algorithm 1.

In algorithm 1, $\nabla_i^{proj} f(\boldsymbol{\alpha})$ is a projected gradient and is denoted as

$$\nabla_i^{proj} f(\boldsymbol{\alpha}) = \begin{cases} \min(0, \nabla_i f(\boldsymbol{\alpha})), & \text{if } \boldsymbol{\alpha}_i = 0 \\ \nabla_i f(\boldsymbol{\alpha}), & \text{if } 0 < \boldsymbol{\alpha}_i < C_3 \boldsymbol{s}_{i-} \\ \max(0, \nabla_i f(\boldsymbol{\alpha})), & \text{if } \boldsymbol{\alpha}_i = C_3 \boldsymbol{s}_{i-} \end{cases}$$

where $\nabla_i f$ denotes the i-th component of gradient $\boldsymbol{\nabla} f$.

V. Numerical Experiments

To show the learning efficiency and generalization ability of CDFTSVM, numerical experiments related to classification accuracy and execution time are conducted on some benchmark datasets. For the multi-classification datasets, we take the majority class as the first class and gathering all the remainders together as the adversary class. To equalize the influence of the features in the input samples, every feature is normalized and scaled-down within [0, 1].

In our experiments, the model parameters $c_i (i = 1, 2, 3, 4)$ are carefully searched in the grids $\{2^i | i = -8, -7, \cdots, 8\}$ by setting $C_1 = C_2$ for TSVM, and $C_1 = C_2$, $C_3 = C_4$ for CDFTSVM. The grid-searching is conducted in a 10-folds cross-validations, randomly selecting 30% of the whole samples for learning with the equivalent conditions mentioned above. In addition, Gaussian kernel is used to deal with the nonlinear cases, *i.e.* $\kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2 / g^2)$. All the experiments are implemented in MATLAB(R2014a) on linux running on a PC with an Intel core i7 processor(3.6GHz) with 32GB RAM, and the Matlab code of all the experiments will be released on webpage[2].

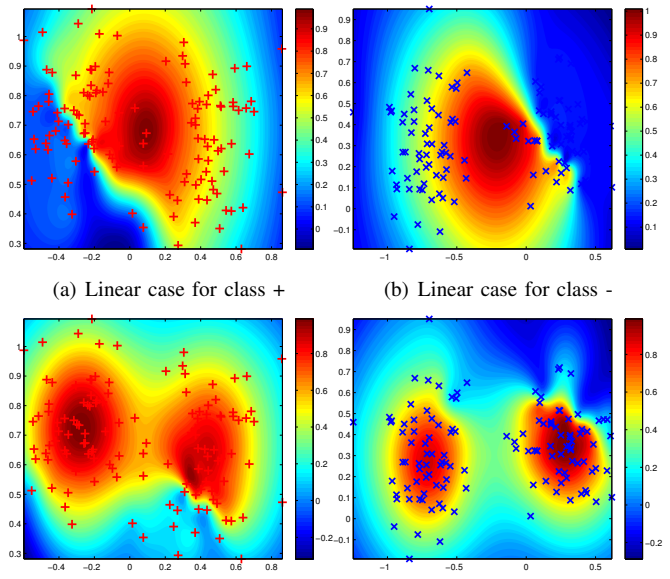
[2]https://github.com/gaobb/CDFTSVM

Algorithm 1 CDFTSVM with active set shrinking

Compute $Q = (H_+^{\mathrm{T}} H_+ + C_1 E)^{-1} H_-^{\mathrm{T}}$ and $\overline{Q}_{ii} = H_{-i} Q_i$
Set $A \leftarrow \{1, \cdots, l_-\}$
Given ϵ and initialized $\boldsymbol{\alpha} \leftarrow \mathbf{0}$,$\boldsymbol{u}_+ \leftarrow \mathbf{0}$
Initialized $\overline{M} \leftarrow \infty$ and $\overline{m} \leftarrow -\infty$
while **do**
  Initialize $M \leftarrow -\infty$,$m \leftarrow \infty$
  for all $i \in A$ (a randomly and exclusively selected) **do**
    Compute $\nabla_i f(\boldsymbol{\alpha}) = -H_{-i}\boldsymbol{u}_+ - 1$
    Assign temporally $\nabla_i^{proj} f(\boldsymbol{\alpha}) \leftarrow 0$
    if $\boldsymbol{\alpha}_i = 0$ **then**
      if $\nabla_i^{proj} f(\boldsymbol{\alpha}) > \overline{M}$, **then** $A = A \backslash \{i\}$ **end if**
      if $\nabla_i^{proj} f(\boldsymbol{\alpha}) < 0$, **then** $\nabla_i^{proj} f(\boldsymbol{\alpha}) \leftarrow \nabla_i f(\boldsymbol{\alpha})$ **end if**
    else if $\boldsymbol{\alpha}_i = C_3 \boldsymbol{s}_{i-}$ **then**
      if $\nabla_i^{proj} f(\boldsymbol{\alpha}) < \overline{m}$, **then** $A = A \backslash \{i\}$ **end if**
      if $\nabla_i^{proj} f(\boldsymbol{\alpha}) > 0$, **then** $\nabla_i^{proj} f(\boldsymbol{\alpha}) \leftarrow \nabla_i f(\boldsymbol{\alpha})$ **end if**
    else
      $\nabla_i^{proj} f(\boldsymbol{\alpha}) \leftarrow \nabla_i f(\boldsymbol{\alpha})$
    end if
    $M \leftarrow \max(M, \nabla_i^{proj} f(\boldsymbol{\alpha}))$
    $m \leftarrow \min(m, \nabla_i^{proj} f(\boldsymbol{\alpha}))$
    if $\nabla_i^{proj} f(\boldsymbol{\alpha}) \neq 0$ **then**
      $\overline{\boldsymbol{\alpha}}_i \leftarrow \boldsymbol{\alpha}_i$
      $\boldsymbol{\alpha}_i \leftarrow \min(\max(\boldsymbol{\alpha}_i - \boldsymbol{\nabla}_i f(\boldsymbol{\alpha})/\overline{Q}_{ii}, 0), C_3 \boldsymbol{s}_{i-})$
      $\boldsymbol{u}_{+i} \leftarrow \boldsymbol{u}_{+i} - Q_i(\boldsymbol{\alpha}_i - \overline{\boldsymbol{\alpha}}_i)$
    end if
  end for
  if $M - m < \epsilon$ **then**
    if $A = \{1, \cdots, l_-\}$, **break**
  else
    $A \leftarrow \{1, \cdots, l_-\}, \overline{M} \leftarrow \infty, \overline{m} \leftarrow -\infty$
    if $M \leq 0$, **then** $\overline{M} \leftarrow \infty$. **else** $\overline{M} \leftarrow M$ **end if**
    if $M \geq 0$, **then** $\overline{m} \leftarrow -\infty$. **else** $\overline{m} \leftarrow m$ **end if**
  end if
end while

A. *Simulation on Artificial Dataset*

Since CDFTSVM is a synthesized method, the experiments first compare it with its original stumps, including the standard SVM, TSVM, and FSVM. To validate its classification performance, the comparative validation is first made on an artificial-generated Ripleys synthetic dataset [20]. This "Ripleys synathetic", is often adopted to gauge a classifier performance. It has 250 training samples with 2 dimensions and is equally divided into two classes, and 1000 testing samples. In order to decrease outlier data's effect toward the hyperplane, we assign a small positive real number $\mu = 0.1$ for CDFTSVM. We visualize the distribution of fuzzy membership value for training samples under linear and nonlinear case in Fig 1, respectively. As shown in Fig 1, compared to the samples locating near the class center, the fuzzy membership value of the samples which are far from the center of class always more smaller.



(a) Linear case for class + (b) Linear case for class -

(c) Nonlinear case for class + (d) Nonlinear case for class -

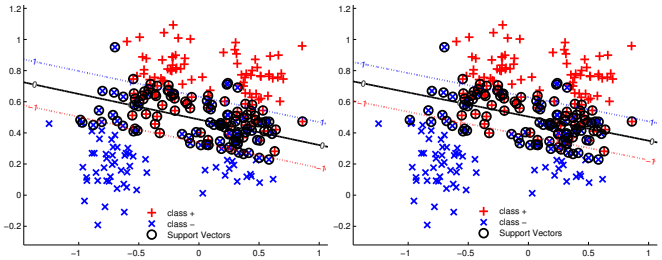
Fig. 1: Fuzzy membership distribution of training samples on Ripleys dataset

Table I summarizes the classification performance of SVM, FSVM, TSVM and our CDFTSVM on Ripleys dataset. The results show that, with respect to the classification accuracy, the linear standard SVM and the nonlinear CDFTSVM performs best. The reason for the outperformance of the linear standard SVM is more likely from the dataset itself than from the classifier due to the fact that there is no noise existing in the artificial dataset. The noiseless fact of the test dataset suppresses the outstanding ability of CDFTSVM in the experiment. The ability of CDFTSVM is confirmed if we examine the accuracy in the nonlinear classification in this table. From the viewpoint of execution time, CDFTSVM shows its excellence in computational efficiency for both linear and nonlinear learning in table I. The excellence manifests the remarkable potential of employing CDFTSVM for a swift classification.

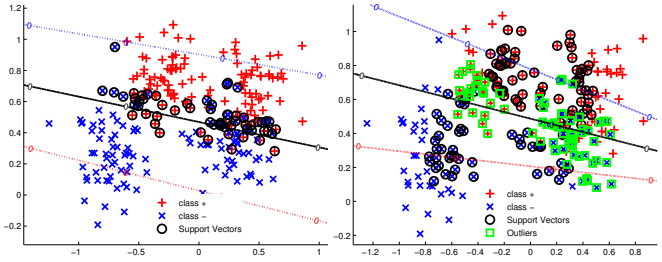
TABLE I: Classification performance comparison on Ripleys dataset

Methods	SVM	FSVM	TSVM	CDFTSVM
	Acc(%) Time(s)	Acc(%) Time(s)	Acc(%) Time(s)	Acc(%) Time(s)
Linear	**89.70** 1.46	88.80 2.00	89.20 0.28	89.10 0.21
Nonlinear	90.40 1.56	91.10 1.79	90.50 0.60	**91.30** 0.24

Panels in Fig. 2 and Fig. 3 show the linear and nonlinear separating hyperplanes produced by the comparative stumps with suitable parameters. In the panels, while the standard SVM and FSVM produce only a single hyperplane (Fig. 2(a), 2(b), 3(a), and 3(b)), TSVM and CDFTSVM produce a paired proximal hyperplanes (Fig. 2(c), 2(d), 3(c), and 3(d)). Instead of the decision boundary identical to the single hyperplane in the standard SVM and FSVM, the pavement-space between the proximal hyperplanes in TSVM and CDFTSVM can be used for a more accurate discrimination. By comparing more



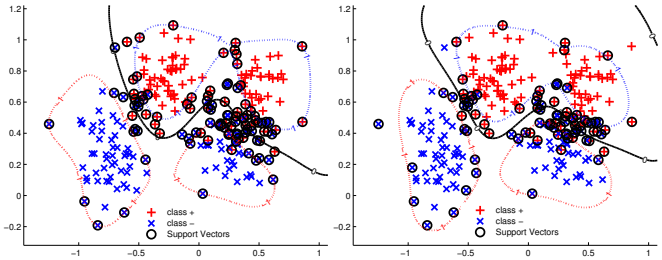
(a) Single hyperplane by the SVM (b) Single hyperplane by the FSVM



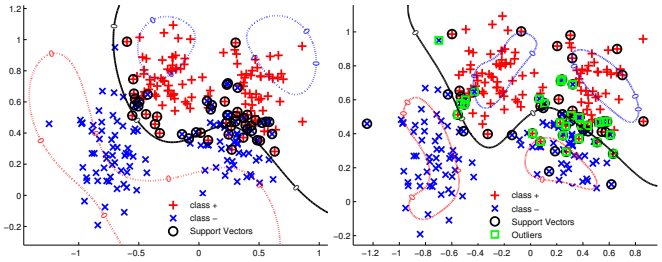
(c) Dual hyperplanes by TSVM (d) Dual hyperplanes by CDFTSVM

Fig. 2: Results of linear SVM, FSVM, TSVM, and CDFTSVM on Ripleys dataset.

of the CDFTSVM and TSVM, the positions of the proximal hyperplanes of CDFTSVM is relatively exact than those of TSVM. The fact reveals that CDFTSVM is more capable to produce an unbiased accuracy than TSVM.



(a) Single hyperplane by the SVM (b) Single hyperplane by the FSVM



(c) Dual hyperplanes by TSVM (d) Dual hyperplanes by CDFTSVM

Fig. 3: Results of nonlinear SVM, FSVM, TSVM, and CDFTSVM on Ripleys dataset.

B. *Simulation on Real Datasets*

To further examine the performance of CDFTSVM, 13 commonly used datasets are gathered from the public UCI machine learning dataset[3].

Similarly, Tables II and III show the significant characteristics which the classifier has concerned in the comparison, including: classification accuracy and total execution time for all algorithm-dataset combination, with both a linear and nonlinear kernel. In order to assess the generalization performance, a 10-folds cross-validation is taken. It means every classifier is repeatedly validated in the datasets with a ratio of 90%/ 10% for respective training / testing phase. Ten characteristics values are collected and averaged for the assessment, and a standard deviation of the 10 collected classification accuracy is provided in addition to the average to reflect the classification robustness.

Comparing to its original stumps, namely, the standard SVM, FSVM and TSVM, Tables II and III show an excellence performance of the CDFTSVM. Except the linear-kernel validation on Pima ,Bupa and WDBC datasets, all the rest datasets consistently confirmed the outperformance of both average accuracy and the accuracy deviation of the CDFTSVM. Additionally, the TSVM exhibits its computational efficiency in table II and III. An obvious reduction in the quadratic programming time actually leads to the efficiency. Inheriting the merit, a drastic computational reduction of CDFTSVM is also generally sustained. In short, the results in Tables II and III indicate that whether linear or nonlinear, CDFTSVM effectively improves the classification accuracy and reduces learning time compared to the traditional stumps. The excellence strongly reflects that the classifier is potential for future applications.

VI. Conclusion

Based on twin support vector machines, the dual form of a fuzzy TSVM is developed for classification in this study. The developed dual form has been brought to a paired convex quadratic programming problems, and confirmed it is capable of solution uniqueness and singularity avoidance. The embedded fuzzy concept enhances the capabilities of noise-resistance and generalization. With the dual form, the FTSVM has been brought to the coordinate descent with an active set shrinking to speed-up the computations of the optimization with less memory. Experiments with simulated and UCI datasets reveal that an exceedingly high classification accuracy rate with less computation time was achieved using both a linear kernel and a nonlinear kernel of the CDFTSVM model.

References

[1] V. N. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.

[2] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2560–2574, 2007.

[3] http://archive.ics.uci.edu/ml/

TABLE II: Comparison on UCI data sets with linear kernel

Dataset	(m× n)	SVM		FSVM		TSVM		CDFTSVM	
		Acc(%)	Time(s)	Acc(%)	Time(s)	Acc(%)	Time(s)	Acc(%)	Time(s)
Breast	(106×9)	96.33±4.77	0.554	**97.17**±4.58	0.571	**97.17**±4.58	0.346	**97.17**±4.58	0.008
Ionosphere	(351×34)	83.53±6.48	2.593	85.75±4.06	2.637	82.33±5.18	0.841	**87.19**±4.22	0.035
Iris	(150×4)	100.00±0.00	0.240	100.00±0.00	0.245	100.00±0.00	0.104	100.00±0.00	0.042
Australian	(690×14)	84.92±4.53	19.536	85.50±4.59	18.613	85.07±4.77	6.035	**85.93**±4.39	0.045
WDBC	(569×30)	95.34±5.17	0.464	95.87±3.21	0.463	93.84±5.86	0.149	**96.39**±3.52	0.049
Wine	(178×13)	**98.89**±2.34	0.365	**98.89**±2.34	0.377	98.33±2.68	0.168	**98.89**±2.34	0.056
Hepatitis	(155×19)	79.26±8.76	0.231	84.94±7.29	0.243	75.58±12.50	0.152	**85.77**±7.21	0.058
WPBC	(198×33)	**79.93**±9.49	0.414	74.24±10.35	0.436	76.88±7.01	0.253	77.96±10.03	0.137
Bupa	(345×6)	66.36±6.04	3.065	**67.51**±7.36	3.114	61.72±5.96	1.662	64.38±6.24	0.177
Sonar	(208×60)	74.08±8.96	0.535	77.46±7.14	0.531	72.15±7.48	0.183	**78.34**±8.29	0.263
Glass	(214×10)	70.41±10.01	1.431	74.18±11.01	1.447	67.64±11.02	0.872	**81.17**±13.56	0.355
Heart	(270×14)	82.22±5.18	2.064	82.59±3.51	2.321	**84.07**±4.95	1.244	**84.07**±6.06	0.323
Pima	(768×8)	**77.21**±3.75	23.199	75.65±4.22	22.796	76.95±3.37	7.969	75.13±3.78	0.482

TABLE III: Comparison on UCI data sets with Rbf kernel

Dataset	(m× n)	SVM		FSVM		TSVM		CDFTSVM	
		Acc(%)	Time(s)	Acc(%)	Time(s)	Acc(%)	Time(s)	Acc(%)	Time(s)
Breast	(106×9)	97.26±4.43	0.570	97.17±4.58	1.205	96.33±4.77	0.352	**98.09**±4.03	0.395
Ionosphere	(351×34)	94.84±4.01	2.558	94.59±4.31	3.775	92.61±6.12	0.954	**95.41**±4.93	0.104
Iris	(150×4)	100.00±0.00	0.245	100.00±0.00	0.479	100.00±0.00	0.106	100.00±0.00	0.039
Australian	(690×14)	85.50±4.53	17.896	85.50±4.59	24.472	85.50±4.59	5.884	**86.81**±4.84	2.354
WDBC	(569×30)	94.84±4.23	0.449	95.34±3.80	0.811	95.34±3.80	0.165	**96.39**±3.52	0.045
Wine	(178×13)	99.44±1.76	0.380	98.89±2.34	0.706	**100.00**±0.00	0.166	**100.00**±0.00	0.039
Hepatitis	(155×19)	80.51±8.32	0.238	84.28±7.42	0.536	82.00±6.84	0.155	**84.48**±6.86	0.050
WPBC	(198×33)	81.51±7.13	0.411	77.88±9.43	0.885	75.30±7.93	0.241	**82.51**±8.05	0.065
Bupa	(345×6)	70.68±8.28	3.125	**72.71**±7.93	4.406	71.86±5.71	1.816	71.84±5.67	0.375
Sonar	(208×60)	89.42±5.41	0.549	88.92±6.95	0.953	89.42±5.41	0.196	**89.44**±5.31	0.065
Glass	(214×10)	**97.68**±3.95	1.581	96.77±4.38	2.001	94.87±5.08	0.968	97.21±3.93	0.217
Heart	(270×14)	84.07±5.25	1.026	82.59±4.29	1.786	80.74±7.16	0.378	**84.81**±4.08	0.050
Pima	(768×8)	75.65±3.80	22.057	75.26±2.91	27.757	**77.34**±5.16	8.278	76.17±2.68	0.226

[3] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machines: Global versus component-based approach," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 688–694.

[4] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[5] O. L. Mangasarian and E. W. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 1, pp. 69–74, 2006.

[6] R. Khemchandani, S. Chandra *et al.*, "Twin support vector machines for pattern classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 5, pp. 905–910, 2007.

[7] M. A. Kumar and M. Gopal, "Least squares twin support vector machines for pattern classification," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7535–7543, 2009.

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

[9] Z.-Q. Qi, Y.-J. Tian, and Y. Shi, "Robust twin support vector machine for pattern classification," *Pattern Recognition*, vol. 46, no. 1, pp. 305–316, 2013.

[10] W.-J. Chen, Y.-H. Shao, and N. Hong, "Laplacian smooth twin support vector machine for semi-supervised classification," *International Journal of Machine Learning and Cybernetics*, vol. 5, no. 3, pp. 459–468, 2014.

[11] S. Ding, J. Yu, B. Qi, et al. "An overview on twin support vector machines," *Artificial Intelligence Review*, vol. 42, no. 2, pp.245–252, 2014.

[12] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 464–471, 2002.

[13] C.-F. Lin *et al.*, "Training algorithms for fuzzy support vector machines with noisy data," *Pattern recognition letters*, vol. 25, no. 14, pp. 1647–1656, 2004.

[14] W. M. Tang, "Fuzzy svm with a new fuzzy membership function to solve the two-class problems," *Neural processing letters*, vol. 34, no. 3, pp. 209–219, 2011.

[15] C.-Y. Yang, J.-J. Chou, and F.-L. Lian, "Robust classifier learning with fuzzy class labels for large-margin support vector machines," *Neurocomputing*, vol. 99, pp. 1–14, 2013.

[16] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[17] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "Coordinate descent method for large-scale l2-loss linear support vector machines," *The Journal of Machine Learning Research*, vol. 9, pp. 1369–1398, 2008.

[18] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear svm," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 408–415.

[19] Y.-H. Shao and N.-Y. Deng, "A coordinate descent margin based-twin support vector machine for classification," *Neural networks*, vol. 25, pp. 114–121, 2012.

[20] B. D. Ripley, *Pattern recognition and neural networks*. Cambridge university press, 1996.