International Journal of Multimedia and Ubiquitous Engineering
Vol.12, No.9 (2017), pp.53-66
http://dx.doi.org/10.14257/ijmue.2017.12.9.05

Random Projection for Linear Twin Support Vector Machine

Huiru Wang[1], Li Sun[1] and Zhijian Zhou[1,*]

[1]*College of Science, China Agricultural University, Beijing 100083, China*
[*]*E-mail: zzjmath@163.com*

Abstract

Twin support vector machine (TSVM) is widely applied in a multitude of aspects. It works faster than SVM, since it solves a pair of smaller-sized quadratic programming problems rather than a larger one. Random projection (RP) is an oblivious feature extraction and dimension reduction method. This paper proposes a novel algorithm, named random projection for twin support vector machine (RP-TSVM), which inherits the high precision and fast solving speed of TSVM bounded with high efficiency and data-independent property of RP. We give two proofs on the geometry of TSVM under random projection. The first is that the sum of squared distances from the hyper-plane to points of one class in TSVM is almost unchanged with high probability, which insure the accuracy of RP-TSVM. The second is that the minimum enclosing ball in the feature space is preserved to within ε - relative error, ensuring comparable generalization as in the original space. Numerical experiments demonstrate the theoretical discoveries. And the computational experimental results also show that the accuracy of the proposed RP-TSVM is higher than RP-SVM. What's more, when solving large scale problems, the proposed algorithm performs almost at least twenty times faster than RP-SVM.

Keywords*: random projection, twin support vector machine, geometry, preserve*

1. Introduction

The support vector machine (SVM) [1] is a popular effective and promising classifier in machine learning. It is good at solving difficulties such as the 'curse of dimensionality', 'over-fitting', and so forth. SVM has been successfully applied in various fields like text categorization, speech recognition, remote sensing image analysis, time series forecasting, and so on.

However, the main challenge of SVM is the high computational complexity. Recently, Jayadeva *et al.* [2] proposed twin support vector machines (TSVM) to improve the computational speed. TSVM generates two nonparallel proximal hyper-planes by solving two smaller-sized quadratic programming problems (QPPs) while SVM solves a larger one, hence the learning speed of TSVM is faster than SVM. And TSVM has become one of the popular methods for its low computational complexity. Many variants of TSVM have been proposed, such as least squares TSVM [3], twin support vector regression (TSVR) [4] and rough *v*-TSVM [5].

For the very high dimensionality problem, high dimensionality also poses high computational overhead to machine learning and data mining algorithms. Dimensionality reduction is an effective technique to tackle the problem. The commonly used dimensionality reduction methods are principal component analysis (PCA) [6], linear discriminant analysis (LDA) [7] and independent component analysis (ICA) [8]. Dimensionality reduction maps data from a high-dimensional space to a low-dimensional sub-space of under the assumption that the intrinsic structure of the high-dimensional data can be retained in the low-dimensional space.

Received (November 9, 2016), Review Result (July 5, 2017), Accepted (July 16, 2017)

ISSN: 1975-0080 IJMUE

However, these methods can only achieve limited performance during the classification. Firstly, the direct projection from the original space to a subspace cannot extract the most discriminative information for classification. Secondly, the most representative structural information in the original space may not be preserved. Thirdly, when adding or deleting a new sample, these methods inevitably need to re-compute the projection matrix which is data-dependent, time-wasting and inefficiently.

Random projection [9, 10] (RP) is a good dimensionality reduction technique due to its efficiency and data-independent property. RP is based on Johnson and Lindenstrauss's (JL) pioneering work. It is a technique of projecting a set of points to a randomly chosen low-dimensional space. The pair-wise distance between samples can be preserved. And the projected dimension *r* is irrelevant to original dimension *d*, but relevant to the number of points *n*. In other words, no training samples are needed to calculate the projection matrix which can be generated beforehand. And even if the data changes, RP does not need to update the projection matrix.

Recently, Paul *et al.*, [11] studied the performance of SVM under RP in the feature space. The theoretical and empirical results indicate that the geometry of SVM can be preserved under RP. The direct use of RP in the classifiers is an interesting research direction. Motivated by the aforementioned works, we proposed a new method named RP-TSVM to solve high dimensional small sample size problem. Firstly, we analyze the geometry properties in terms of theoretical aspects for the proposed method. And empirical experiments demonstrated the discoveries. That is, the accuracy of RP-TSVM keeps almost the same with TSVM. Secondly, the computational experiments show that the classification accuracy of our method is higher than that of RP-SVM under the same dimension. And the solving speed of our method is almost at least twenty times faster than RP-SVM for large scale problems.

The remainder of this paper is organized as follows. Section 2 outlines the prior work based on RP and TSVM. Section 3 mainly introduces our RP-TSVM and analyzes its geometry performance. Numerical experiments in Section 4 testified the efficiency and feasibility of our proposed algorithm. Finally, we make conclusions in Section 5.

2. Prior Work

Suppose the training dataset is $T=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),...,(\mathbf{x}_n,y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ are inputs and $y_i \in \{-1,1\}$ are the corresponding outputs. A hyper-plane $\mathbf{w}^* = \mathbf{X}^T\mathbf{Y}\boldsymbol{\alpha}^*$ is used to separate the data and maximizes the geometric margin in the primal form of SVM, where $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the original input matrix; $\mathbf{Y} \in \mathbb{R}^{n\times n}$ is the diagonal matrix with entries $\mathbf{Y}_{ii} = y_i$; $\boldsymbol{\alpha} = [\alpha_1,\alpha_2,\cdots,\alpha_n] \in \mathbb{R}^n$ is the Lagrange multiplier vector. And the dual formulation of SVM is denoted as follows,

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \ \mathbf{e}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Y}\mathbf{X}\mathbf{X}^T\mathbf{Y}\boldsymbol{\alpha} \\ &\text{s.t.} \quad \mathbf{e}^T\mathbf{Y}\boldsymbol{\alpha} = 0, \quad \mathbf{0} \le \boldsymbol{\alpha} \le \mathbf{C}. \end{aligned} \tag{1}$$

2.1. Random Projection Matrices

Random projections are popular techniques in dealing with the curse-of dimensionality. There are many random projections that satisfy JL theorem. And we list some of the common used random projection matrices.

1) Gaussian random matrix [12]: $r_{ij} \sim N(0,1)$.

2) Random sign matrix [13, 14]:

$$r_{ij}=\begin{cases}\sqrt{3}, & p=1/6,\\ 0, & p=2/3,\\ -\sqrt{3}, & p=1/6,\end{cases} \quad \text{or} \quad r_{ij}=\begin{cases}\sqrt{3}, & p=1/2,\\ -\sqrt{3}, & p=1/2.\end{cases}$$

Later, Li used sparse random projection matrix to improve random sign matrix [15].

3) Sampling random matrix of Hadamard transform [16], it is also called fast Hadmard transform: $\mathbf{R}_{SRHT}=\sqrt{d/r}\,\mathbf{DHS}$, where $\mathbf{D}\in\mathbb{R}^{d\times d}$ is a random diagonal matrix satisfying $\mathbf{D}_{ii}=\begin{cases}+1, & p=1/2\\ -1, & p=1/2\end{cases}$; $\mathbf{H}\in\mathbb{R}^{d\times d}$ is the normalized Hadmard transform matrix; $\mathbf{S}\in\mathbb{R}^{d\times r}$ is a random sampling matrix which randomly samples columns of $\mathbf{DH}$.

4) Generalized sparse embedding matrices [17].

Random projection is an effective and efficient method of feature extraction and comprehensively applied in the area of compressed sensing [18] , camera fingerprint matching [19], texture classification [20] and face recognition [21].

2.2. Twin Support Vector Machine

TSVM seeks two nonparallel proximal hyper-planes instead of a single one in traditional SVMs. These two nonparallel hyper-planes are obtained by solving two smaller-sized QPPs instead of a larger one, which makes TSVM work faster than SVM. For a binary classification problem, suppose there are *p* samples belongs to class +1 and *q* samples belongs to class -1, and $p+q=n$. Let matrix $\mathbf{A}\in\mathbb{R}^{p\times d}$ and $\mathbf{B}\in\mathbb{R}^{q\times d}$ represents the positive and negative samples, respectively. Then the linear TSVM seeks the following pair of nonparallel hyper-planes:

$$(\mathbf{w}_+\cdot\mathbf{x})+b_+=0 \text{ and } (\mathbf{w}_-\cdot\mathbf{x})+b_-=0. \tag{2}$$

such that each hyper-plane is closer to one of the two classes and is as far as possible from the other. For the linear case, TSVM is obtained by solving the following pair of QPPs:

$$\begin{aligned}\min_{\mathbf{w}_+,b_+,\zeta_-}\quad & \frac{1}{2}(\mathbf{A}\mathbf{w}_++\mathbf{e}_+b_+)^T(\mathbf{A}\mathbf{w}_++\mathbf{e}_+b_+)+c_1\mathbf{e}_-^T\zeta_-\\ \text{s.t.}\quad & -(\mathbf{B}\mathbf{w}_++\mathbf{e}_-b_+)+\zeta_-\ge\mathbf{e}_-,\ \ \zeta_-\ge\mathbf{0}\end{aligned} \tag{3}$$

and

$$\begin{aligned}\min_{\mathbf{w}_-,b_-,\zeta_+}\quad & \frac{1}{2}(\mathbf{B}\mathbf{w}_-+\mathbf{e}_-b_-)^T(\mathbf{B}\mathbf{w}_-+\mathbf{e}_-b_-)+c_2\mathbf{e}_+^T\zeta_+\\ \text{s.t.}\quad & (\mathbf{A}\mathbf{w}_-+\mathbf{e}_+b_-)+\zeta_+\ge\mathbf{e}_-,\ \ \zeta_+\ge\mathbf{0}\end{aligned} \tag{4}$$

where $c_i>0,\ i=1,2$ are the penalty parameters, $\mathbf{e}_+$ and $\mathbf{e}_-$ are vectors of ones of appropriate dimensions, ζ_+ and ζ_- are slack vectors of appropriate dimension. By introducing the Lagrangian coefficient $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, we can derive their dual problems as follows:

$$\begin{aligned}\max_{\alpha}\quad & \mathbf{e}_-^T\boldsymbol{\alpha}-\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{G}^T\boldsymbol{\alpha}\\ \text{s.t.}\quad & \mathbf{0}\le\boldsymbol{\alpha}\le\mathbf{c}_1\\ \text{where,}\ & \mathbf{H}=[\mathbf{A}\ \mathbf{e}_+],\mathbf{G}=[\mathbf{B}\ \mathbf{e}_-].\end{aligned} \tag{5}$$

and

$$\max_{\alpha} \quad \mathbf{e}_+^T\boldsymbol{\gamma} - \frac{1}{2}\boldsymbol{\gamma}^T\mathbf{P}(\mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{P}^T\boldsymbol{\gamma}$$
$$\text{s.t.} \quad \mathbf{0} \le \boldsymbol{\gamma} \le \mathbf{c}_2$$
$$\text{where, } \mathbf{P} = [\mathbf{A}\ \mathbf{e}_+], \mathbf{Q} = [\mathbf{B}\ \mathbf{e}_-]. \tag{6}$$

Therefore, $(\mathbf{w}_+^T, b_+)^T = -(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{G}^T\boldsymbol{\alpha}$ and $(\mathbf{w}_-^T, \mathbf{b}_-)^T = (\mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{P}^T\boldsymbol{\gamma}$.

A new testing point $\mathbf{x} \in ¡^d$ is predicted by $Class = \arg\min_{k=-,+} |(\mathbf{w}_k \cdot \mathbf{x}) + b_k|$, where $|\cdot|$ is the perpendicular distance of point $\mathbf{x}$ from the two planes $(\mathbf{w}_k \cdot \mathbf{x}) + b_k = 0,\ k = -, +$.

Besides, we give two definitions.

Definition1: The sum of squared distances from one hyper-plane to points of one class of TSVM:

$$\kappa^{*2} = \boldsymbol{\alpha}^T\mathbf{G}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{G}^T\boldsymbol{\alpha} = \boldsymbol{\gamma}^T\mathbf{P}(\mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{P}^T\boldsymbol{\gamma}.$$

Definition2: Data radius: $\mathrm{B} = \min_{\mathbf{x}^*} \max_{\mathbf{x}_i} \|\mathbf{x}_i - \mathbf{x}^*\|_2$, where $\mathbf{x}^*$ is the center of the minimum inclosing ball.

In fact, we usually use $\mathbf{H}^T\mathbf{H} + \varepsilon\mathbf{I}$ and $\mathbf{G}^T\mathbf{G} + \varepsilon\mathbf{I}$ to replace $\mathbf{H}^T\mathbf{H}$ and $\mathbf{G}^T\mathbf{G}$ in order to ensure the matrices are singular, where $\mathbf{I}$ is an identity matrix of appropriate dimensions, ε is a positive scalar, small enough to keep the structure of data.

2.3. RP-SVM

Paul *et al*. [11] showed that random projection can preserve the subspace geometry, and it can preserve the performance of SVM. That is they proved that with high probability, the margin and minimum enclosing ball in the feature space are preserved to within ε-relative error.

Let $\mathbf{R} \in ¡^{d \times r}$ be the dimension reduction matrix that reduces the dimensionality of input from *d* to *r* (*r*<*d*), and $\mathbf{R}$ is a random projection matrix. So the input dataset becomes $\tilde{\mathbf{X}} = \mathbf{XR}$, and the dual form of RP-SVM optimization model is:

$$\max_{\tilde{\alpha}} \quad \mathbf{e}^T\tilde{\boldsymbol{\alpha}} - \frac{1}{2}\tilde{\boldsymbol{\alpha}}^T\mathbf{YXRR}^T\mathbf{X}^T\mathbf{Y}\tilde{\boldsymbol{\alpha}}$$
$$\text{s.t.} \quad \mathbf{e}^T\mathbf{Y}\tilde{\boldsymbol{\alpha}} = 0, \text{and } \mathbf{0} \le \tilde{\boldsymbol{\alpha}} \le \mathbf{C} \tag{7}$$

where $\tilde{\boldsymbol{\alpha}} = [\tilde{\alpha}_1, \tilde{\alpha}_2, \cdots, \tilde{\alpha}_n] \in ¡^n$ is the Lagrange multiplier vector.

Let γ^*, $\tilde{\gamma}^*$ be the resulting margin after solving the optimization problem in the original space and in the transformed space, respectively; B, $\tilde{B}$ be the data radius of the minimum ball enclosing all points in the original space and projected space, respectively. The following inequalities are satisfied: $\tilde{\gamma}^{*2} \ge (1-\varepsilon)\gamma^{*2}$; $\tilde{B}^2 \le (1+\varepsilon)\mathrm{B}^2$.

3. Our Work

In this paper, we proposed a new algorithm, named random projection for linear twin support vector machine (RP-TSVM). We also prove that with high probability, the sum of squared distances from one hyper-plane to points of one class of TSVM and the minimum enclosing ball enclosing all the points in the feature space are preserved, that is

$$\kappa^{*2} \ge (1-2\varepsilon)\cdot\tilde{\kappa}^{*2} \text{ and } \tilde{B}^2 \le (1+\varepsilon)B^2.$$

3.1. Random Projection for Linear Twin Support Vector Machine

In real life, especially in web or image classification, the size of dimension (*d*) is

usually extremely huge and its orders of magnitude up to millions. It is a big challenge to solve these kind of problems. RP is an overwhelmingly popular technique of dimensionality reduction and it can maintain the basic data structure unchanged with high probability. Therefore, we deal with the original data using RP and propose RP-TSVM to improve the running speed and testing accuracy.

We use RP to reduce the dimension, but we cannot reduce the dimension arbitrary and we need to find the most reasonable and effective reduced dimension. Achilioptas gives the theoretical basis in 2001 [13], in which he gave a lower bound of the reduced dimension.

Theorem (Achlioptas, 2001):

Suppose that $\mathbf{A}$ is a $n \times d$ matrix with n points in $\mathbb{R}^d$. Fix constants $\varepsilon, \beta > 0$, and choose an integer r such that

$$r \geq r_0 := \frac{4+2\beta}{\varepsilon^2/2 - \varepsilon^3/2} \log n .$$

We introduce the transformed dataset $\tilde{\mathbf{A}} = \mathbf{AR}, \tilde{\mathbf{B}} = \mathbf{BR}$ and RP-TSVM is denoted as follows:

$$\begin{aligned} \max_{\alpha} \quad & \mathbf{e}_-^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{G} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{G}^T \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{c}_1 \\ \text{where,} \quad & \mathbf{H} = [\tilde{\mathbf{A}} \ \mathbf{e}_+], \mathbf{G} = [\tilde{\mathbf{B}} \ \mathbf{e}_-] . \end{aligned} \tag{8}$$

and

$$\begin{aligned} \max_{\alpha} \quad & \mathbf{e}_+^T \boldsymbol{\gamma} - \frac{1}{2} \boldsymbol{\gamma}^T \mathbf{P} (\mathbf{Q}^T \mathbf{Q})^{-1} \mathbf{P}^T \boldsymbol{\gamma} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\gamma} \leq \mathbf{c}_2 \\ \text{where,} \quad & \mathbf{P} = [\tilde{\mathbf{A}} \ \mathbf{e}_+], \mathbf{Q} = [\tilde{\mathbf{B}} \ \mathbf{e}_-] . \end{aligned} \tag{9}$$

By solving the two above dual QPPs, the nonparallel hyper-planes can be obtained by: $(\tilde{\mathbf{w}}_+^T, \tilde{b}_+)^T = -(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{G}^T \boldsymbol{\alpha}$ and $(\tilde{\mathbf{w}}_-^T, \tilde{b}_-)^T = (\mathbf{Q}^T \mathbf{Q})^{-1} \mathbf{P}^T \boldsymbol{\gamma}$.

Similarly, the sum of squared distances from one hyper-plane to points of one class of RP-TSVM: $\tilde{\kappa}^{*2} = \tilde{\boldsymbol{\alpha}}^{*T} \tilde{\mathbf{G}} (\tilde{\mathbf{H}}^T \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{G}}^T \tilde{\boldsymbol{\alpha}}^* = \tilde{\boldsymbol{\gamma}}^{*T} \tilde{\mathbf{P}} (\tilde{\mathbf{Q}}^T \tilde{\mathbf{Q}})^{-1} \tilde{\mathbf{P}}^T \tilde{\boldsymbol{\gamma}}^*$.

Solving the dimensionally-reduced problem above is computationally more efficient than solving the original d-dimensional problem. The running time of reducing the original data is nearly linear on the size of the original data [11].

3.2. Geometry of TSVM Is Preserved Under Random Projection

In this subsection, we state our main findings. We still use Lemma 3.1 in [11], which is crucial in the following proofs.

Lemma1: *Fix* $\varepsilon \in (0, 1/2], \delta \in (0,1]$. *Let* $\mathbf{V} \in \mathbb{R}^{d \times \rho}$ *be any matrix with orthonormal columns and let* $\mathbf{R} \in \mathbb{R}^{d \times r}$ *be the Gaussian random projection matrix with* $r = O(\rho \varepsilon^{-2} \log(\rho/\delta))$. *Then with probability at least* $1-\delta$,

$$\| \mathbf{V}^T \mathbf{V} - \mathbf{V}^T \mathbf{R} \mathbf{R}^T \mathbf{V} \|_2 \leq \varepsilon .$$

Here, we give the definition of the Singular Value Decomposition (SVD) [9] of matrix. Suppose $\mathbf{A} \in \mathbb{R}^{n \times d}$, $rank\ \mathbf{A} = \rho \leq \min\{n, d\}$, then matrix $\mathbf{A}$ can be decompose to $\mathbf{A} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T$, and the column vector of $\mathbf{U}$ is called left singular vector of matrix $\mathbf{A}$, $\mathbf{U} \in \mathbb{R}^{n \times \rho}$ satisfy $\mathbf{U}^T \mathbf{U} = \mathbf{I}$; the column vector of $\mathbf{V}$ is called right singular vector of matrix $\mathbf{A}$, $\mathbf{V} \in \mathbb{R}^{d \times \rho}$ satisfy $\mathbf{V}^T \mathbf{V} = \mathbf{I}$; $\boldsymbol{\Sigma} \in \mathbb{R}^{\rho \times \rho}$ is a diagonal matrix,

$\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, \cdots, \sigma_\rho)$, its singular value satisfy $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_\rho > 0$. The spectral norm of matrix $\mathbf{A}$ is $\|\mathbf{A}\|_2 = \sigma_1$. Then we give our main theorem and proofs as follows:

Theorem1: *Let ε be an accuracy parameter and let $\mathbf{R} \in \mathbb{R}^{d \times r}$ be a matrix satisfying $\| \mathbf{V}^T\mathbf{V} - \mathbf{V}^T\mathbf{R}\mathbf{R}^T\mathbf{V} \|_2 \leq \varepsilon$. Let κ^{*2} and $\tilde{\kappa}^{*2}$ be the sum of squared distances from the hyper-plane to points of one class obtained by solving the dual TSVM problems (5) and dual RP-TSVM problem (8). Then,*

$$\kappa^{*2} \geq (1-2\varepsilon) \cdot \tilde{\kappa}^{*2}.$$

Proof:

Firstly, we prove the TSVM1's the sum of squared distances from the hyper-plane to points of one class satisfies the theorem. And the other QPP will be proved by the same method.

Then we give a transformation of QPP(5), and its objective function can be written as follows:

$$\begin{aligned} Z_{opt} &= \mathbf{e}_-^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{G}^T\boldsymbol{\alpha} \\ &= \mathbf{e}_-^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}\mathbf{H}^{-1}\mathbf{H}^{-1^T}\mathbf{G}^T\boldsymbol{\alpha} \\ &= \mathbf{e}_-^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}\mathbf{H}^{-1}(\mathbf{G}\mathbf{H}^{-1})^T\boldsymbol{\alpha} \\ &= \mathbf{e}_-^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{M}\mathbf{M}^T\boldsymbol{\alpha} \end{aligned}$$

Then we can rewrite QPP (5) as

$$\begin{aligned} \max_{\alpha} \quad & \mathbf{e}_-^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{M}\mathbf{M}^T\boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{c}_1 \end{aligned} \tag{10}$$

Similarly, we rewrite eqn.(8) as follows. Let $\tilde{\mathbf{M}} = \mathbf{M}\mathbf{R}$, then we can get the following equation.

$$\begin{aligned} \max_{\alpha} \quad & \mathbf{e}_-^T\tilde{\boldsymbol{\alpha}} - \frac{1}{2}\tilde{\boldsymbol{\alpha}}^T\tilde{\mathbf{M}}\tilde{\mathbf{M}}^T\tilde{\boldsymbol{\alpha}} \\ \text{s.t.} \quad & \mathbf{0} \leq \tilde{\boldsymbol{\alpha}} \leq \mathbf{c}_1 \end{aligned} \tag{11}$$

Let $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \cdots, \alpha_p^*]^T \in \mathbb{R}^p$ be the vector achieving the optimal solution for the problem of eqn.(10). Then,

$$\begin{aligned} Z_{opt} &= \boldsymbol{\alpha}^* - \frac{1}{2}\boldsymbol{\alpha}^{*T}\mathbf{M}\mathbf{M}^{\mathbf{T}}\boldsymbol{\alpha}^* \\ &= \boldsymbol{\alpha}^* - \frac{1}{2}\boldsymbol{\alpha}^{*T}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\boldsymbol{\alpha}^* \\ &= \boldsymbol{\alpha}^* - \frac{1}{2}\boldsymbol{\alpha}^{*T}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{R}\mathbf{R}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\boldsymbol{\alpha}^* \\ &\quad - \frac{1}{2}\boldsymbol{\alpha}^{*T}\mathbf{U}\mathbf{\Sigma}\mathbf{E}\mathbf{\Sigma}\mathbf{U}^T\boldsymbol{\alpha}^* \end{aligned} \tag{12}$$

where, $\mathbf{E} = \mathbf{V}^T\mathbf{V} - \mathbf{V}^T\mathbf{R}\mathbf{R}^T\mathbf{V}$. We still use SVD of matrices, and $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$.

Let $\tilde{\boldsymbol{\alpha}}^* = [\tilde{\alpha}_1^*, \tilde{\alpha}_2^*, \cdots, \tilde{\alpha}_p^*]^T \in \mathbb{R}^p$ be the vector achieving the optimal solution for the dimensionally-reduced TSVM of eqn.(11) using $\tilde{\mathbf{M}} = \mathbf{M}\mathbf{R}$. Using the SVD of $\mathbf{M}$, we get

$$\begin{aligned}\tilde{Z}_{opt} &= \tilde{\boldsymbol{\alpha}}^* - \frac{1}{2}\tilde{\boldsymbol{\alpha}}^{*T}\tilde{\mathbf{M}}\tilde{\mathbf{M}}^T\boldsymbol{\alpha}^* \\ &= \tilde{\boldsymbol{\alpha}}^* - \frac{1}{2}\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\mathbf{R}\mathbf{R}^T\mathbf{M}^T\boldsymbol{\alpha}^* \\ &= \tilde{\boldsymbol{\alpha}}^* - \frac{1}{2}\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{R}\mathbf{R}^T\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\boldsymbol{\alpha}^*\end{aligned} \tag{13}$$

Since the constrains on $\boldsymbol{\alpha}^*, \tilde{\boldsymbol{\alpha}}^*$ do not depend on the data, it is clear that $\tilde{\boldsymbol{\alpha}}^*$ is a feasible solution for the problem of eqn.(10). Thus, from the optimality of $\boldsymbol{\alpha}^*$, and using upper eqn.(13), it follows that

$$\begin{aligned}Z_{opt} &= \boldsymbol{\alpha}^* - \frac{1}{2}\boldsymbol{\alpha}^{*T}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{R}\mathbf{R}^T\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\alpha^* - \frac{1}{2}\boldsymbol{\alpha}^{*T}\mathbf{U}\boldsymbol{\Sigma}\mathbf{E}\boldsymbol{\Sigma}\mathbf{U}^T\boldsymbol{\alpha}^* \\ &\geq \tilde{\boldsymbol{\alpha}}^* - \frac{1}{2}\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{R}\mathbf{R}^T\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\tilde{\boldsymbol{\alpha}}^* - \frac{1}{2}\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{U}\boldsymbol{\Sigma}\mathbf{E}\boldsymbol{\Sigma}\mathbf{U}^T\tilde{\boldsymbol{\alpha}}^* \\ &= \tilde{Z}_{opt} - \frac{1}{2}\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{U}\boldsymbol{\Sigma}\mathbf{E}\boldsymbol{\Sigma}\mathbf{U}^T\tilde{\boldsymbol{\alpha}}^*\end{aligned}$$

$$\text{That is} \qquad Z_{opt} \geq \tilde{Z}_{opt} - \frac{1}{2}\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{U}\boldsymbol{\Sigma}\mathbf{E}\boldsymbol{\Sigma}\mathbf{U}^T\tilde{\boldsymbol{\alpha}}^* \tag{14}$$

Let us taking the second term of eqn.(14) using standard sub-multiplicativity properties and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$. Taking $\mathbf{Q} = \tilde{\boldsymbol{\alpha}}^{*T}\mathbf{U}\boldsymbol{\Sigma}$, we get

$$\begin{aligned}\frac{1}{2}\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{U}\boldsymbol{\Sigma}\mathbf{E}\boldsymbol{\Sigma}\mathbf{U}^T\tilde{\boldsymbol{\alpha}}^* &\leq \frac{1}{2}\|\mathbf{Q}\|_2\|\mathbf{E}\|_2\|\mathbf{Q}^T\|_2 \\ &= \frac{1}{2}\|\mathbf{E}\|_2\|\mathbf{Q}\|_2^2 \\ &= \frac{1}{2}\|\mathbf{E}\|_2\|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\|_2^2 \\ &= \frac{1}{2}\|\mathbf{E}\|_2\|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\|_2^2\end{aligned} \tag{15}$$

Combining eqn.(14) and (15), we get

$$Z_{opt} \geq \tilde{Z}_{opt} - \frac{1}{2}\|\mathbf{E}\|_2\|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\|_2^2 \quad . \tag{16}$$

Now we see the second term of eqn.(16), we could find the difference:

$$\begin{aligned}&|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\mathbf{R}\mathbf{R}^T\mathbf{M}^T\tilde{\boldsymbol{\alpha}}^* - \tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\mathbf{M}^T\tilde{\boldsymbol{\alpha}}^*| \\ =\ &|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{U}\boldsymbol{\Sigma}(\mathbf{V}^T\mathbf{R}\mathbf{R}^T\mathbf{V} - \mathbf{V}^T\mathbf{V})\boldsymbol{\Sigma}\mathbf{U}^T\tilde{\boldsymbol{\alpha}}^*| \\ =\ &|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{U}\boldsymbol{\Sigma}(-\mathbf{E})\boldsymbol{\Sigma}\mathbf{U}^T\tilde{\boldsymbol{\alpha}}^*| \\ \leq\ &\|\mathbf{E}\|_2\|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{U}\boldsymbol{\Sigma}\|_2^2 \\ =\ &\|\mathbf{E}\|_2\|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\|_2^2 \\ =\ &\|\mathbf{E}\|_2\|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\|_2^2\end{aligned}$$

Because

$$\begin{aligned}\|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\|_2^2 - \|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\mathbf{R}\|_2^2 &\leq \left|\|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\mathbf{R}\|_2^2 - \|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\|_2^2\right| \\ &= |\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\mathbf{R}\mathbf{R}^T\mathbf{M}^T\tilde{\boldsymbol{\alpha}}^* - \tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\mathbf{M}^T\tilde{\boldsymbol{\alpha}}^*| \\ &\leq \|\mathbf{E}\|_2\|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\|_2^2\end{aligned}$$

Then, we can get

$$\|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\|_2^2 \leq \frac{1}{1-\|\mathbf{E}\|_2}\|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\mathbf{R}\|_2^2. \tag{17}$$

We combine eqn.(14) and (15), we get

$$Z_{opt} \geq \tilde{Z}_{opt} - \frac{1}{2}\left(\frac{\|\mathbf{E}\|_2}{1-\|\mathbf{E}\|_2}\right)\|\tilde{\boldsymbol{\alpha}}^{*T}\mathbf{M}\mathbf{R}\|_2^2 \tag{18}$$

As the discussion in [2], that we get

$$Z_{opt} = \frac{1}{2}\| \kappa^* \|_2^2, \text{ and } \tilde{Z}_{opt} = \frac{1}{2}\| \tilde{\kappa}^* \|_2^2. \quad (19)$$

Solving eqn.(6) we get

$$(\mathbf{w}_+^T, b_+)^T = -(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{G}^T\boldsymbol{\alpha} \text{ and } \| \tilde{\boldsymbol{\alpha}}^{*T}\mathbf{MR} \|_2^2 = \tilde{\boldsymbol{\alpha}}^{*T}\mathbf{MRR}^T\mathbf{M}^T\tilde{\boldsymbol{\alpha}}^* = \| \tilde{\kappa}^* \|_2^2. \quad (20)$$

So from eqn.(16), we get

$$\frac{1}{2}\| \kappa^* \|_2^2 \geq \frac{1}{2}\| \tilde{\kappa}^* \|_2^2 - \frac{1}{2}\left(\frac{\| \mathbf{E} \|_2}{1-\| \mathbf{E} \|_2} \right)\| \tilde{\kappa}^* \|_2^2 \quad (21)$$

That is

$$\| \kappa^* \|_2^2 \geq 1-\left(1-\left(\frac{\| \mathbf{E} \|_2}{1-\| \mathbf{E} \|_2} \right) \right)\| \tilde{\kappa}^* \|_2^2 \quad (22)$$

From $\| \mathrm{E} \|_2 = \| \mathbf{V}^T\mathbf{V} - \mathbf{V}^T\mathbf{R}\mathbf{R}^T\mathbf{V} \|_2 \leq \varepsilon$, thus $\frac{\| \mathbf{E} \|_2}{1-\| \mathbf{E} \|_2} \leq 2\varepsilon$, then $\kappa^{*2} \geq (1-2\varepsilon)\cdot\tilde{\kappa}^{*2}$ is proved.

Our second theorem discusses the radius of the minimum ball enclosing all points in original feature space and projected feature space is really close to each other. Here is the theorem.

Theorem2: *Fix* $\varepsilon \in (0, 1/2], \delta \in (0,1]$. *Let* B *be the radius of the minimum ball enclosing all points in the full-dimensional space (the rows of the matrix* $\mathbf{X}$ *), and let* $\tilde{B}$ *be the radius of the ball enclosing all points in the dimensionally reduced space (the rows of the matrix* $\mathbf{XR}$ *). Then, if* $r = O(\rho\varepsilon^{-2}\log(\rho/\delta))$ *, that is* $\mathbf{R} \in \mathbb{R}^{d\times r}$ *be the Gaussian random projection matrix, with probability at least* $1-\delta$ *,*

$$\tilde{B}^2 \leq (1+\varepsilon)B^2.$$

Proof:

We set $\tilde{\mathbf{A}} = \mathbf{AR}, \tilde{\mathbf{B}} = \mathbf{BR}$ and let $\mathbf{X} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$, so $\mathbf{XR} = \begin{bmatrix} \mathbf{AR} \\ \mathbf{BR} \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}\mathbf{R}$. We consider the matrix $\mathbf{X}_B \in \mathbb{R}^{(n+1)\times d}$, whose first *n* rows are the rows of $\mathbf{X}$ and whose last row is the vector $\mathbf{x}_B^T$, where $\mathbf{x}_B$ denotes the center of the minimum radius ball enclosing all *n* points. So the progress of proving is the almost same as [11]. Here we will not give the detailed proof.

4. Experiments

To evaluate the performance of our algorithm, in this section, we compare our RP-TSVM with RP-SVM, SVM and TSVM on eight benchmark datasets. These datasets come from Mldata [22] and UCI database [23], and they all satisfy $n \gg d$.

These datasets are Orlface, Dbworld emails, Eyes, Yale face, Arcene.nips, Yahoo web directory topics, Dmoz web directory topics. For Orlface, we choose (32,32) pixel and get 1024 dimension, naming it Orlface32; we choose (64,64) pixel and get 4096 dimension, naming Orlface64; Arcene.nips includes Arcene_train and Arcene_valid. Dmoz web directory topics includes five different classes, we choose data with labels 6 and 7 as dataset Dmoz67, data with labels 7 and 8 as dataset Dmoz 78 and data with labels 5 and 9 as dataset Dmoz 59. Yahoo web directory topics includes four different classes, we divided it into two part: data with label 1 and 2 as a dataset named Yahoo12, and the others as Yahoo34. We use these datasets for binary classification.

In conclusion, we use three different kinds of datasets to do our experiment. The first part with light blue stands for the dataset with small sample size and light low dimension;

the second part with light pink stands for small sample size and slight high dimension; the third part with light green stands for larger sample size and high dimension.

We set ε=0.3 in theorem to certain the value of $r_0 = 4(\varepsilon^2/2 - \varepsilon^3/3)^{-1}\log n + 1$ with Gauss random matrix and the entries of matrix are chosen independently from $N(0,1)$. The dataset's composition is displayed as Table 1. We set the parameter c_1=c_2 in TSVM and RP-TSVM. All these experiments run in the computer Intel (R) Core (TM) 2 Duo CPU E7500 2.94GHz, with memory of 4.00GB（3.00GB available）32 bits Windows7 in the platform of MATLAB 7.10.0 (R2010a).

We partitioned the data randomly for five-fold cross-validation in order to estimate the accuracy. The parameters c, c_1, c_2 are chosen from $\{2^i \mid i = -5,-4,\cdots,5\}$. We report its mean value μ and its standard deviation σ. The title '*Accuracy*' is composed by the form of $\mu \pm \sigma$. '*Time*' concludes the time of computing random matrix and the time of five-fold cross-validation.

Table 1. Description of Datasets

Dataset	*Samples*	*Positive*	*Negative*	*Dimension*	r_0
Orlface32	20	10	10	1024	334
Orlface64	20	10	10	4096	334
Dbworld	64	35	29	4702	464
Eyes	80	40	40	4704	488
Yale face	22	11	11	10000	345
Arcene_train	100	44	56	10000	513
Arcene_valid	100	44	56	10000	513
Dmoz67	524	261	263	10629	697
Dmoz78	528	263	265	10629	698
Dmoz59	540	268	272	10629	701
Yahoo12	548	254	294	10629	702
Yahoo34	558	284	274	10629	704

4.1. Experiments on Benchmark Datasets

The results are displayed in TABLE 2. We find that it is difficult for SVM and TSVM to solve large scale problem, especially when the dimension is high. While RP-SVM and RP-TSVM can deal with high dimensional problems effectively and efficiently. What's more, the accuracy of TSVM and RP-TSVM are almost the same, which validate our proposed theorem1.

For dataset Orl32, Orl64 and Eyes, the accuracy of the four compared algorithms all yields 100%, and RP-SVM takes the least time, followed by SVM, RP-TSVM and TSVM. When the scale of dataset increase, like Yale dataset and Arcene dataset, *i.e.* the size of sample or the dimension increase, the accuracy of RP-TSVM yields the highest, but the time advantage is not obvious. The scale of Dmoz datasets and Yahoo datasets are larger scale datasets. The proposed RP-TSVM have an obvious advantage, in terms of not only the accuracy but also the running time.

The reason is that RP-TSVM embedded the data from high space into appropriate low subspace and the distance of different data are controlled in a small error. Besides, it solves two smaller-sized QPPs rather than a single large one to shorten the time, thus it offers more space to make more specific decision function.

Table 2. Accuracy on Different Datasets

Datasets	Metrics	SVM	RP-SVM	TSVM	RP-TSVM
Orlface32	***Accuracy***	**100.00±0.00**	**100.00±13.69**	**100.00±0.00**	**100.00±0.00**
	Time(s)	0.69	0.75	240.50	15.69
Orlface64	***Accuracy***	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
	Time(s)	0.75	0.68	2862.00	15.79
Dbworld	***Accuracy***	**85.00±12.36**	83.33±13.18	82.55±12.45	84.36±5.20
	Time(s)	4.33	3.18	26558.00	4.79
Eyes	***Accuracy***	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
	Time(s)	4.77	3.69	29341.00	35.48
Yale face	***Accuracy***	**100.00±0.00**	**100.00±0.00**	N/A	**100.00±0.00**
	Time(s)	1.19	0.82	N/A	16.21
Arcene_train	***Accuracy***	88.42±5.77	85.26±7.81	N/A	**87.42±6.15**
	Time(s)	5.89	2.530	N/A	40.48
Arcene_valid	***Accuracy***	84.21±7.44	**85.26±7.81**	N/A	84.92±9.38
	Time(s)	6.65	3.38	N/A	40.21
Dmoz67	***Accuracy***	N/A	65.38±5.81	N/A	**77.23±5.46**
	Time(s)	N/A	2090.30	N/A	105.00
Dmoz78	***Accuracy***	N/A	64.19±7.02	N/A	**81.04±4.22**
	Time(s)	N/A	1450.40	N/A	109.02
Dmoz59	***Accuracy***	N/A	49.35±0.78	N/A	**60.28±9.33**
	Time(s)	N/A	2687.40	N/A	111.07
Yahoo12	***Accuracy***	N/A	58.70±3.85	N/A	**69.64±2.27**
	Time(s)	N/A	2482.70	N/A	113.04
Yahoo34	***Accuracy***	N/A	58.00±5.73	N/A	**67.70±3.32**
	Time(s)	N/A	2799.80	N/A	115.56

4.2. Influence of Parameter ε

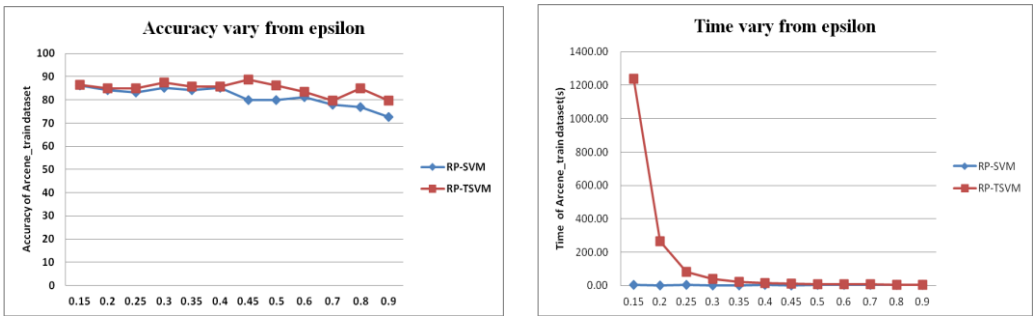
In order to see the performance of RP-TSVM with different $\varepsilon \in (0,1)$, we choose a medium scale dataset Arcene_train and a large scale dataset Yahoo34, Dmoz67 to do the experiment. At the same time, we record the lowest dimension r_0 under different ε.

The results on Arcene_train dataset are shown in Table 3 and Figure1. From Table 3, we can see that as the increase of ε, the lowest value of dimension r_0 decrease slightly. Under the same ε, the accuracy of RP-TSVM all higher than RP-SVM， but the running time of RP-TSVM is not obvious compared with RP-SVM, especially when ε=0.1 or 0.2, the time of RP-TSVM is very slow. In order to see the change of accuracy and time, we draw Figure1.

From Figure1(a), we can see that the accuracy of RP-TSVM is higher than RP-SVM under the same ε. As ε increase, the accuracy of RP-TSVM and RP-SVM both decrease, but the decrease speed of RP-TSVM is slower than RP-SVM, which indicates the stability of RP-TSVM is better than that of RP-SVM. When $\varepsilon \in (0, 0.5]$, the accuracy of RP-TSVM and RP-SVM are stable, while $\varepsilon \in (0.5, 0.9)$, their accuracy fall sharply, that indicates the value of ε cannot beyond the range of values $(0, 0.5]$. That further verified our proposed Theorem1. From Figure1(b), RP-TSVM takes more time compared with RP-SVM, but it falls deeply when ε=0.2. That shows it is available to get better classification ability for RP-TSVM in a large scale problem.

Table 3. Accuracy of Arcene_Train Datasets(100*10000)

		RP-SVM		RP-TSVM	
ε	r_0	*Accuracy*	*Time(s)*	*Accuracy*	*Time(s)*
0.1	3949	87.37±6.00	4.97	**84.83±5.51**	11757.00
0.15	1821	86.32±7.98	3.65	**86.52±4.94**	1238.00
0.2	1064	84.21±6.44	2.75	**84.83±9.43**	264.06
0.25	709	83.16±4.40	3.71	**84.83±5.51**	81.84
0.3	513	85.26±7.81	2.53	**87.42±6.15**	40.49
0.35	394	84.21±9.85	2.73	**85.71±3.69**	22.86
0.4	315	85.26±10.79	2.92	**85.71±4.95**	15.32
0.45	261	80.00±6.86	2.65	**88.75±8.15**	10.88
0.5	223	80.00±10.12	2.90	**86.17±5.08**	8.08
0.6	172	81.05±7.06	3.23	**83.42±10.83**	6.41
0.7	142	77.89±6.86	3.64	**79.75±8.12**	6.36
0.8	125	76.84±9.56	3.90	**84.83±7.06**	5.07
0.9	115	72.63±7.81	3.88	**79.75±5.18**	4.80



(a) (b)

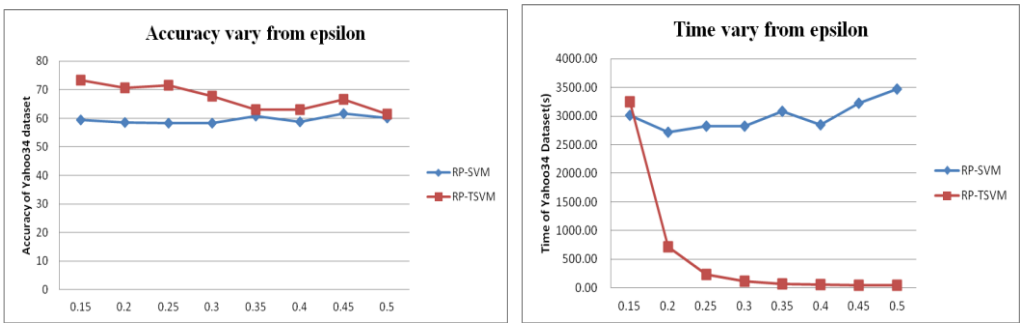
Figure1. Accuracy (a) and Time (b) Vary from Epsilon in Arcene_train Dataset

In order to see the rationality of RP-TSVM dealing with large scale problems. We choose Yahoo34 to do the experiment which is a larger scale dataset. The result is shown in TABLE 4. The results indicate that as the increase of ε, the lowest value of dimension r_0 decrease slightly. When ε equals 0.1, it cannot use RP-TSVM to obtain the result. Under the same ε, both the testing accuracy and the running time of RP-TSVM is better than RP-SVM. In order to see the variation of accuracy and time as ε, we draw Figure2.

From Figure2(a), we see that the accuracy of RP-TSVM is higher than RP-SVM under the same ε. As ε increase, the accuracy of RP-TSVM and RP-SVM vary slightly. From Figure2(b), we get that RP-TSVM takes less time compared with RP-SVM under the same ε except when epsilon equals 0.15. That indicates our RP-TSVM can deal with large scale problem quickly and efficiently. And as ε increase, RP-SVM needs more and more time to solve the problem. RP-TSVM all need almost over 2500 seconds while the time of RP-TSVM less than 1000 seconds whenever ε is. That states our RP-TSVM can shorten the time sharply and keep the accuracy unchanged at the same time.

Table 4. Accuracy of Yahoo34 Datasets (558*10629)

		RP-SVM		RP-TSVM	
ε	r_0	*Accuracy*	*Time(s)*	*Accuracy*	*Time(s)*
0.1	5422	58.36±5.73	2755.20	N/A	N/A
0.15	2500	59.45±4.24	3016.60	**73.32±6.53**	3247.1
0.2	1461	58.55±6.19	**2720.90**	**70.63±3.61**	**712.18**
0.25	973	58.18±4.98	**2827.60**	**71.53±2.56**	**230.85**
0.3	704	58.18±3.91	**2830.00**	**67.70±3.32**	**115.56**
0.35	540	60.73±5.28	**3080.90**	**63.02±4.31**	**72.20**
0.4	433	58.73±4.84	**2842.40**	**62.99±4.94**	**53.84**
0.45	358	61.64±4.83	**3220.30**	**66.60±3.85**	**47.28**
0.5	305	60.00±3.75	**3472.70**	**61.45±4.74**	**47.13**



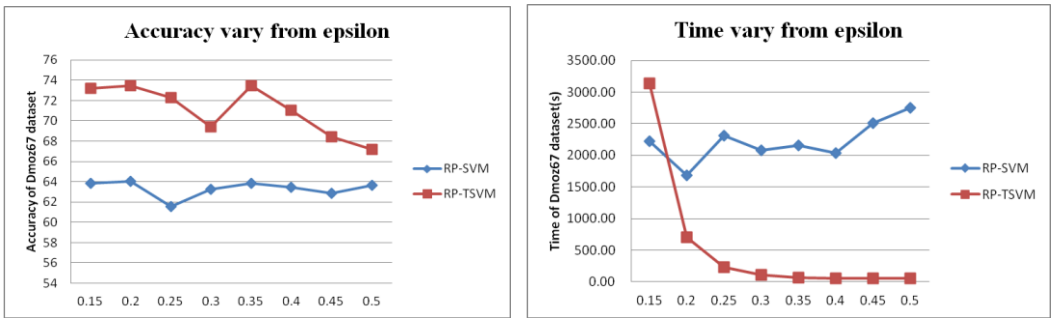
(a) (b)

Figure 2. Accuracy (a) and Time (b) Vary from Epsilon in Yahoo34 Dataset

In order to test our theorem further, we do the experiment on Dmoz67 dataset. The results are displayed in Table 5. It shows that r_0 decrease as ε increase, when ε equals 0.5, r_0 equals 302. When ε equal 0.1, r_0 equal 5,368 which is still a large scale problem, we cannot use RP-TSVM; When ε equal 0.2, r_0 equal 1,446, using RP-TSVM can get higher accuracy and use almost a half time compared with using RP-SVM, the accuracy increased almost 10%; When $\varepsilon \in (0.3, 0.5)$, we use less time and get higher accuracy using RP-TSVM. In order to see the tendency of accuracy and time vary from ε, we draw Figure 3.

Table 5. Accuracy of Dmoz67 Datasets(524*10629)

		RP-SVM		RP-TSVM	
ε	r_0	*Accuracy*	*Time(s)*	*Accuracy*	*Time(s)*
0.1	5368	65.96±5.97	2253.30	N/A	N/A
0.15	2475	63.85±5.29	2220.30	**73.20±5.19**	3137.20
0.2	1446	64.04±6.14	1681.00	**73.44±5.99**	**705.76**
0.25	913	61.54±2.71	2314.70	**72.27±6.29**	**231.83**
0.3	697	63.27±10.82	2078.90	**69.39±3.67**	**108.54**
0.35	535	63.84±8.34	2154.70	**73.46±3.81**	**68.21**
0.4	428	63.46±1.80	2033.00	**71.04±3.80**	**53.00**
0.45	355	62.88±2.59	2509.00	**68.41±2.96**	**51.12**
0.5	302	63.65±6.52	2755.70	**67.21±4.17**	**50.20**



(a) (b)

Figure 3. Accuracy (a) and Time (b) Vary from Epsilon in Dmoz67 Dataset

From Figure3(a), we can see it clearly that the accuracy of our RP-TSVM is all higher than RP-SVM. From Figure3(b), we can see it clearly that our RP-TSVM takes less time compared with RP-SVM. When $\varepsilon \in (0.15, 0.5]$, the time of RP-SVM vary slowly as ε increase, the time keeps around 2,000 seconds; while $\varepsilon \in (0.3, 0.5]$, RP-TSVM cost the least time, and the time keeps around 50 seconds, *i.e.* our RP-TSVM is almost forty times faster than RP-SVM.

5. Summary

In this paper, we propose a new algorithm named random projection for twin support vector machine. RP-TSVM inherits not only the high solving speed and high precision of TSVM but also the efficiency and data-independent property of RP. The crucial contribution of this paper is that we find two paramount theorems and give detailed proofs. We prove that within high probability, the sum of squared distances from the hyper-plane to points of one class of TSVM and the minimum enclosing ball in the feature space are preserved, ensuring comparable generalization as in the original space. Large amounts of experiments verified our proposed theorem. What's more, the experiment results also show that the accuracy of our proposed RP-TSVM is higher than RP-SVM, and the solving speed of our proposed RP-TSVM is faster. In further research, we will study the different random production under different SVMs, and it will be a great value in the area of big data.

Acknowledgments

This work was supported in part by the China Agricultural Research System (CARS-30).

References

[1] V. N. Vapnik, "The nature of statistical learning theory", Berlin: Springer, **(1995)**.
[2] R. Jayadeva Khemchandani and S. Chandra, "Twin Support Vector Machines for Pattern Classification", IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 29, no. 5, **(2007)**, pp. 905-910.
[3] M. A. Kumar and M. Gopal, "Least squares twin support vector machines for pattern classification [J]," Expert Systems with Applications, vol. 36, no. 4, **(2009)**, pp. 7535-7543.
[4] X. J. Peng, "TSVR: An efficient twin support vector machine for regression," Neural Networks, vol. 23, no. 3, **(2010)**, pp. 365-372.
[5] H. Wang and Z. Zhou, "An improved rough margin-based *v*-twin bounded support vector machine," Knowledge-Based Systems, vol. 128, **(2017)**, pp. 125-138.
[6] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space", Philosophical Magazine, vol. 2, no. 11, **(1901)**, pp. 559–572.
[7] H. Abdi, "Discriminant correspondence analysis", In: N.J. Salkind (Ed.): Encyclopedia of Measurement and Statistic, Thousand Oaks (CA): Sage, **(2007)**, pp. 270-275.
[8] M. S. Bartlett, J. R. Movellan and T. J. Sejnowski, "Face recognition by independent component

analysis", IEEETrans. Neural Netw, vol. 13, **(2002)**, pp. 1450–1464.
[9] W. B. Johnson, and J. Lindenstrauss, "Extensions of Lipschitz maps into a Hilbert space," In Contemp Math, vol. 26, **(1984)**, pp. 189-206.
[10] L. Zhang, M. Mahdavi and R. Jin, "Recovering Optimal Solution by Dual Random Projection," In Conference on Learning Theory (COLT) JMLR W&CP, no. 30, **(2013)**, pp. 135-157.
[11] S. Paul, "Random Projections for Linear Support Vector Machines", ACM Transactions on knowledge discovery from data, vol. 8, no. 224, **(2014)**, pp. 1-20.
[12] S. Dasgupta and A. Gupta, "An elementary proof of a theorem of Johnson and Lindenstrauss", Random Structures & Algorithms, vol. 22, no. 1, **(2003)**, pp. 60 - 65.
[13] D. Achlioptas, "Database-friendly random projections," in Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Santa Barbara, California, USA: ACM, **(2001)**.
[14] D. Achlioptas, "Database-friendly random projections: Johnson- Lindenstrauss with binary coins", Journal of computer and System Sciences, vol. 66, no. 4, **(2003)**, pp. 671-687.
[15] P. Li, T. J. Hastie and W. K. Church, "Very Spase Random Projections," In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, **(2006)**, pp. 287-296.
[16] N. Alion, and B. Chazelle, "Approximate nearest neighbors and the fast Johnson-Lindenstrass transform", In Proceedings of the 38th Annual ACM Symposium on Theory of Computing, (**2006)**, pp. 557-563.
[17] K. L. Clarkson and D. W. Woodruff, "Low Rank Approximation and Regression in Input Sarsity Time," In Proceedings of the 45th ACM Symposium on the Theory of Computing, **(2013)**.
[18] P. Li, and C. Zhang, "Compressed Sensing with Very Sparse Gaussian Random Projections," AISTATS, **(2015)**.
[19] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Scale-robust compressive camera fingerprint matching with random projections", ICASSP, **(2015)**, pp. 697-1701.
[20] L. Liu, P. W. Fieguth, D. Hu, Y. Wei, and G. Kuang, "Fusing Sorted Random Projections for Robust Texture and Material Classification," IEEE Trans. Circuits Syst. Video Techn., vol. 25, no. 3, **(2015)**, pp. 482-496.
[21] C. Ma, J. Jung, S. Kim, and S. Ko, "Random projection-based partial feature extraction for robust face recognition," Neurocomputing, vol. 149, **(2015)**, pp.1232-1244.
[22] Mldata.org machine learning data set responsitory, Available: http://mldata.org/repository/tags/data/Classification/
[23] UCI database, Available: http://archive.ics.uci.edu/ml/datasets.html

