Stochastic Gradient Twin Support Vector Machine for Large Scale Problems

Zhen Wang, Yuan-Hai Shao, Lan Bai, Li-Ming Liu, and Nai-Yang Deng

Abstract—For classification problems, twin support vector machine (TSVM) with nonparallel hyperplanes has been shown to be more powerful than support vector machine (SVM). However, it is time consuming and insufficient memory to deal with large scale problems due to calculating the inverse of matrices. In this paper, we propose an efficient stochastic gradient twin support vector machine (SGTSVM) based on stochastic gradient descent algorithm (SGD). As far as now, it is the first time that SGD is applied to TSVM though there have been some variants where SGD was applied to SVM (SGSVM). Compared with SGSVM, our SGTSVM is more stable, and its convergence is also proved. In addition, its simple nonlinear version is also presented. Experimental results on several benchmark and large scale datasets have shown that the performance of our SGTSVM is comparable to the current classifiers with a very fast learning speed.

Index Terms—Classification, support vector machine, twin support vector machine, stochastic gradient descent.

1 INTRODUCTION

Support vector machine (SVM), being powerful tool for classification [1], [2], [3], has already outperformed most other classifiers in a wide variety of applications [4], [5], [6]. Different from SVM with a pair of parallel hyperplanes, twin support vector machine (TSVM) [7] with a pair of nonparallel hyperplanes has been proposed and developed, e.g., twin bounded support vector machine (TB SVM) [8], twin parametric margin support vector machine (TPMSVM) [9], and weighted Lagrangian twin support vector machine (WLTSVM) [10]. These classifiers have been widely applied in many practical problems [11], [12], [13], [14], [15]. In the training stage, SVM solves a quadratic programming problem (QPP), whereas TSVM solve two smaller QPPs by traditional solver such as interior method [2], [15], [7]. However, neither SVM nor TSVM based on these solvers can deal with the large scale problems, especially millions of samples.

In order to deal with the large scale problems, many improvements were proposed, e.g., for SVM, sequential minimal optimization, coordinate decent method, and trust region Newton in [17], [18], [19], [20], and for TSVM, successive overrelaxation technique, Newton-Armijo algorithm, and dual coordinate decent method in [8], [12], [21]. The stochastic gradient descent algorithm for SVM (SGSVM, PEGASOS) [22], [23], [24], [25] attracts a great attention, because it partitions the large scale problem into a series of subproblems by stochastic sampling with a suitable size. It has been proved that SGSVM is almost sure convergent, and thus is able to find an approximation of the desired solution with high probability [25], [23], [24]. The experiments confirm the effectiveness of these algorithms with an amazing learning speed.

However, for large scale problems, the sampling in SGD may bring some difficulties to SVM due to only a small subset of the dataset is selected for training. In fact, if the subset is not suitable, SGSVM would be weak. It is well known that in SVM the support vectors (SVs), a small subset of the dataset, decides the final classifier. If the stochastic sampling does not include the SVs sufficiently, the classifier would lose some generalizations. Figure 1 is a toy example for SGSVM. There are two classes in this figure, where the positive and the negative class respectively include 6 and 4 samples, where the circle is one of the potential SVs. The solid blue line is the separating line obtained by SGSVM with three different sampling: (i) strengthening the circle sample; (ii) infrequently using the circle sample; (iii) ignoring the circle sample. Figure 1 shows that the circle sample plays an important role, and infrequently using or ignoring this sample would lead to misclassify.

Compared with SVM, it is significant that TSVM is more stable for sampling and does not strongly depend on some special samples such as the SVs [2], [8], which indicates SGS is more suitable for TSVM. Therefore, in this paper, we propose a stochastic gradient twin support vector machine (SGTSVM). Different from SGSVM, our method selects two samples from different classes randomly in each iteration to construct a pair of nonparallel hyperplanes. Due to TSVM fits all of the training samples, our method is stable for the stochastic sampling and thus gains well generalizations. Moreover, the characteristics inherited from TSVM result in that our SGTSVM suits many cases, e.g., “cross plane” dataset [27] and preferential classification [2]. As the toy example, Figure 2 shows the corresponding results by SGTSVM. Comparing Figure 1 and Figure 2, it is clear that...
SGTSVM performs better than SGSVM.

The main contributions of this paper includes:
(i) a SGD-based TSVM (SGTSVM) is first proposed, and it is very easy to be extended to other TSVM-type classifiers;
(ii) we prove that the proposed SGTSVM is convergent, instead of almost sure convergence in SGSVM;
(iii) for the uniformly sampling, it is proved that the original objective of the solution to SGTSVM is bounded instead of almost sure convergence in SGSVM;
(iv) the nonlinear case of SGTSVM is obtained directly based on its original problem, whereas the nonlinear case of SGSVM is derived from SVM’s dual problem;
(v) each iteration of SGTSVM includes no more than $8n + 4$ multiplications without additional storage, so it is the fastest one than other proposed TSVM-type classifiers.

The rest of this paper is organized as follow. Section 2 briefly reviews SVM, SGSVM, and TBSVM. Our linear and nonlinear SGTSVMs together with the theoretical analysis are elaborated in Section 3. Experiments are arranged in Section 4. Finally, we give the conclusions.

2 RELATED WORKS

Consider a binary classification problem in the $n$-dimensional real space $R^n$. The set of training samples is represented by $X \in R^{n \times m}$, where $x \in R^n$ is the sample with the label $y \in \{+1, -1\}$. We further organize the $m_1$ samples of Class $+1$ into a matrix $X_1 \in R^{n \times m_1}$ and the $m_2$ samples of Class $-1$ into a matrix $X_2 \in R^{n \times m_2}$. Below, we give a brief outlines some related works.

2.1 SVM

C-support vector machine (CSVM) [28], one formulation of the standard SVM, searches for a separating hyperplane

$$w^T x + b = 0,$$  

where $w \in R^n$ and $b \in R$. By introducing the regularization term, the primal problem of CSVM can be expressed as a quadratic programming problem (QPP) as follow

$$\min_{w, b} \frac{1}{2} ||w||^2 + \frac{c}{m} e^T \xi$$

$$\text{s.t.} \quad D(X^T w + b) \geq e - \xi, \quad \xi \geq 0,$$  

where $|| \cdot ||$ denotes the $L_2$ norm, $c > 0$ is a parameter with some quantitative meanings [28], $e$ is a vector of ones with an appropriate dimension, $\xi \in R^m$ is the slack vector, and $D = \text{diag}(y_1, \ldots, y_m)$. Note that the minimization of the regularization term $\frac{1}{2} ||w||^2$ is equivalent to maximize the margin between two parallel supporting hyperplanes $w^T x + b = \pm 1$. And the structural risk minimization principle is implemented in this problem [1].
2.2 SGSVM

SGSVM (or PEGASOS as an alias) [23, 24] considers a strongly convex problem by modifying (2) as follow
\[
\min_w \frac{1}{2}||w||^2 + \frac{c}{m} e^T \xi \\
\text{s.t. } DX^Tw \geq e - \xi, \xi \geq 0,
\]
and recasts the above problem to
\[
\min_w \frac{1}{2}||w||^2 + \frac{c}{m} e^T (e - DX^Tw)_+,
\]
where \((\cdot)_+\) replaces negative components of a vector by zeros.

In the \(t\)th iteration \((t \geq 1)\), SGSVM constructs a temporary function, which is defined by a random sample \(x_t \in X\) as
\[
g_t(w) = \frac{1}{2}||w||^2 + c(1 - y_t^w x_t)_+.
\]
Then, starting with an initial \(w_1\), SGSVM iteratively updates \(w_{t+1} = w_t - \eta_t \nabla_t\) for \(t \geq 1\), where \(\eta_t = 1/t\) is the step size and \(\nabla_t\) is the sub-gradient of \(g_t(w)\) at \(w_t\),
\[
\nabla_t = w_t - c y_t x_t \text{sign}(1 - y_t w_t^T x_t)_+.
\]
After a predetermined \(T\) iterations, the last \(w_T+1\) is outputted as \(w\). And a new sample \(x\) can be predicted by
\[
y = \text{sign}(w^T x).
\]

It has been proved that the average solution \(\bar{w} = \frac{1}{T} \sum_{t=1}^T w_t\) is bounded by the optimal solution \(w^*\) to (4) with \(o(1)\), and thus SGSVM has a probability of at least 1/2 to find a good approximation of \(w^*\) [24]. The authors of [24] also pointed out that \(w_T\) is often used instead of \(\bar{w}\) in practice. The sample \(x_t\), which is selected randomly can be replaced with a small subset belonging to the whole dataset, and the subset only including a sample is often used in practice [23, 24, 25]. In order to extend the generalization ability of SGSVM, the bias term \(b\) in CSVVM can be appended to SGSVM by replacing \(g_t(w)\) of (3) with
\[
g(w_t, b) = \frac{1}{2}||w_t||^2 + C(1 - y_t(w_t^T x_t + b))_+ .
\]
However, this modification would lead to the function not to be strongly convex and thus yield a slow convergence rate [24].

2.3 TBSVM

TBSVM [3], a representative of TSVM, seeks a pair of non-parallel hyperplanes in \(R^n\) which can be expressed as
\[
w_1^T x + b_1 = 0 \quad \text{and} \quad w_2^T x + b_2 = 0,
\]
such that each hyperplane is close to samples of one class and has a certain distance from the other class. To find the pair of nonparallel hyperplanes, it is required to get the solutions to the primal problems
\[
\min_{w_1, b_1} \frac{1}{2}(||w_1||^2 + b_1^2) + \frac{c_1}{m_1} ||X_1^Tw_1 + b_1||^2 + \frac{c_2}{m_2} e^T \xi_1, \\
\text{s.t. } X_1^Tw_1 + b_1 - \xi_1 \leq -e, \xi_1 \geq 0,
\]
and
\[
\min_{w_2, b_2} \frac{1}{2}(||w_2||^2 + b_2^2) + \frac{c_1}{m_1} ||X_2^Tw_2 + b_2||^2 + \frac{c_2}{m_2} e^T \xi_2, \\
\text{s.t. } X_1^Tw_2 + b_2 + \xi_2 \geq e, \xi_2 \geq 0,
\]
where \(c_1, c_2, c_3,\) and \(c_4\) are positive parameters, \(\xi_1 \in R^{m_1}\) and \(\xi_2 \in R^{m_2}\) are slack vectors. Their geometric meaning is clear. For example, for (10), its objective function makes the samples of the first class proximal to the hyperplane \(w_1^T x + b_1 = 0\) together with the regularization term, while the constraints make each sample of the second class has a distance more than \(1/||w_1||\) away from the hyperplane \(w_1^T x + b_1 = -1\).

Once the solutions \((w_1, b_1)\) and \((w_2, b_2)\) to the problems (10) and (11) are respectively obtained, a new point \(x \in R^n\) is assigned to which class depends on the distance to the two hyperplanes in (9), i.e.,
\[
y = \arg \min_x \frac{|w_1^Tx + b_1|}{||w_1||} ,
\]
where \(| \cdot |\) is the absolute value.

3 SGTSVM

In this section, we elaborate our SGTSVM with its nonlinear formation, and give its convergence analysis together with the boundedness.

3.1 Linear Formation

Following the notations in Section 2, we recast the QPPs (10) and (11) to unconstrained problems
\[
\min_{w_1, b_1} \frac{1}{2}(||w_1||^2 + b_1^2) + \frac{c_1}{m_1} ||X_1^Tw_1 + b_1||^2 + \frac{c_2}{m_2} e^T (e + X_1^Tw_1 + b_1)_+, \quad (13)
\]
and
\[
\min_{w_2, b_2} \frac{1}{2}(||w_2||^2 + b_2^2) + \frac{c_1}{m_1} ||X_2^Tw_2 + b_2||^2 + \frac{c_2}{m_2} e^T (e - X_1^Tw_2 - b_2)_+, \quad (14)
\]
respectively.

In order to solve the above two problems, we construct a series of strictly convex functions \(f_{1,t}(w_1, b_1)\) and \(f_{2,t}(w_2, b_2)\) with \(t \geq 1\) as
\[
f_{1,t} = \frac{1}{2}(||w_1||^2 + b_1^2) + \frac{c_1}{m_1} ||X_1^Tw_1 + b_1||^2 + \frac{c_2}{m_2} e^T (1 + w_1^Tx_t + b_1)_+, \quad (15)
\]
and
\[
f_{2,t} = \frac{1}{2}(||w_2||^2 + b_2^2) + \frac{c_1}{m_1} ||X_2^Tw_2 + b_2||^2 + \frac{c_2}{m_2} e^T (1 - w_2^Tx_t - b_2)_+, \quad (16)
\]
where \(x_t\) and \(\hat{x}_t\) are selected randomly from \(X_1\) and \(X_2\), respectively.

The sub-gradients of the above functions at \((w_{1,t}, b_{1,t})\) and \((w_{2,t}, b_{2,t})\) can be obtained as
\[
\nabla_{w_{1,t}, f_{1,t}} = w_{1,t} + c_1 (w_{1,t}^T x_t + b_{1,t}) x_t \\
+ c_2 \hat{x}_t \text{sign}(1 + w_{1,t}^T \hat{x}_t + b_{1,t})_+, \\
\nabla_{b_{1,t}, f_{1,t}} = b_{1,t} + c_1 (w_{1,t}^T x_t + b_{1,t}) \\
+ c_2 \text{sign}(1 + w_{1,t}^T \hat{x}_t + b_{1,t})_+, \\
\nabla_{w_{2,t}, f_{2,t}} = w_{2,t} + c_3 (w_{2,t}^T \hat{x}_t + b_{2,t}) \hat{x}_t \\
- c_4 x_t \text{sign}(1 - w_{2,t}^T x_t - b_{2,t})_+, \\
\nabla_{b_{2,t}, f_{2,t}} = b_{2,t} + c_3 (w_{2,t}^T \hat{x}_t + b_{2,t}) \\
- c_4 \text{sign}(1 - w_{2,t}^T x_t - b_{1,t})_+, \quad (18)
\]
respectively.

Our SGTSVM starts from the initial \((w_{1,1}, b_{1,1})\) and \((w_{2,1}, b_{2,1})\). Then, for \(t \geq 1\), the updates are given by

\[
\begin{align*}
w_{1,t+1} &= w_{1,t} - \eta_t \nabla_{w_{1,t}} f_{1,t}, \\
b_{1,t+1} &= b_{1,t} - \eta_t \nabla_{b_{1,t}} f_{1,t}, \\
w_{2,t+1} &= w_{2,t} - \eta_t \nabla_{w_{2,t}} f_{2,t}, \\
b_{2,t+1} &= b_{2,t} - \eta_t \nabla_{b_{2,t}} f_{2,t},
\end{align*}
\tag{19}
\]

where \(\eta_t\) is the step size and typically is set to \(1/t\). If the terminated condition is satisfied, \((w_{1,1}, b_{1,1})\) and \((w_{2,1}, b_{2,1})\) are obtained. Then, a new sample \(x \in \mathbb{R}^n\) can be predicted using the same as TBSVM.

The above procedures can be summarized as follows

**Algorithm 1** SGTSVM Framework.

**Input:**

Given the training dataset \(X_1 \in \mathbb{R}^{n \times m_1}\) as positive class, \(X_2 \in \mathbb{R}^{n \times m_2}\) as negative class, select parameters \(c_1, c_2, c_3, c_4\), and a small tolerance \(\text{tol}\), typically \(\text{tol} = 1e-4\).

**Output:**

\[w_1, b_1, w_2, b_2,\]

1. Set \(w_{1,1}, b_{1,1}, w_{2,1}, b_{2,1}\) as zeros; For \(t = 1, 2, \ldots\)
2. Choose a pair of samples \(x_t\) and \(\hat{x}_t\) from \(X_1\) and \(X_2\) at random, respectively;
3. Compute the \(t\)th gradients by \((17)\) and \((18)\);
4. Update \(w_{1,t+1}, b_{1,t+1}, w_{2,t+1}, b_{2,t+1}\) by \((19)\);
5. If \(|w_{1,t+1} - w_{1,t}| + |b_{1,t+1} - b_{1,t}| < \text{tol}\), stop updating \(w_{1,t+1}\) and \(b_{1,t+1}\).
6. Let \(w_1 = w_{1,t+1}, b_1 = b_{1,t+1}\);
7. If \(|w_{2,t+1} - w_{2,t}| + |b_{2,t+1} - b_{2,t}| < \text{tol}\), stop updating \(w_{2,t+1}\) and \(b_{2,t+1}\).

3.2 Nonlinear Formation

Now, we extend our SGTSVM to nonlinear case by the kernel trick \((27), (17), (8), (29), (30)\). Suppose \(K(\cdot, \cdot)\) is the predefined kernel function, then the nonparallel hyperplanes can be expressed as

\[K(x, X)w_1 + b_1 = 0 \quad \text{and} \quad K(x, X)w_2 + b_2 = 0. \tag{20}\]

The counterparts of \((13)\) and \((14)\) can be formulated as

\[
\begin{align*}
\min_{w_1, b_1} & \frac{1}{2}||w_1||^2 + b_1^2 + \frac{c_1}{m_1}||K(X_1, X)w_1 + b_1||^2 + \frac{c_2}{m_2}e^T(e + K(X_2, X)w_1 + b_1) + ,
\end{align*}
\tag{21}
\]

and

\[
\begin{align*}
\min_{w_2, b_2} & \frac{1}{2}||w_2||^2 + b_2^2 + \frac{c_3}{m_1}||K(X_1, X)w_2 + b_2||^2 + \frac{c_4}{m_2}e^T(e - K(X_1, X)w_2 - b_2) + .
\end{align*}
\tag{22}
\]

Then, we construct a series of functions with \(t \geq 1\) as

\[
\begin{align*}
h_{1,t} &= \frac{1}{2}||w_1||^2 + b_1^2 + \frac{c_1}{2m_1}||K(x_t, X)w_1 + b_1||^2 + c_2(1 + K(\hat{x}_t, X)w_1 + b_1) + ,
\end{align*}
\tag{23}
\]

and

\[
\begin{align*}
h_{2,t} &= \frac{1}{2}||w_2||^2 + b_2^2 + \frac{c_3}{2m_1}||K(x_t, X)w_2 + b_2||^2 + c_4(1 - K(x_t, X)w_2 - b_2) + .
\end{align*}
\tag{24}
\]

For large scaled problems, it is time consuming to calculate the kernel \(K(\cdot, X)\). However, the reduced kernel strategy, which has been successfully applied for SVM and TSVMS \((31), (32), (12)\), can also be applied for our SGTSVM. The reduced kernel strategy replaces \(K(\cdot, X)\) with \(K(\cdot, \hat{X})\), where \(\hat{X}\) is a random sampled subset of \(X\). In practice, \(\hat{X}\) just needs \(1\% - 10\%\) samples from \(X\) to get a well performance, reducing the learning time without loss of generalization \((32)\).

3.3 Analysis

In this subsection, we discuss two issues: (i) the convergence of the solution in SGTSVM; (ii) the relation between the solution in SGTSVM and the optimal one in TBSVM. For convenience, we just consider the first QPP of linear TBSVM together with its SGD formation. The conclusions on another QPP and the nonlinear formations can be obtained easily the same as the first one.

Let \(u = (w^T, b)^T, Z_1 = (X_1^T, e)^T, Z_2 = (X_2^T, e)^T, z = (x^T, 1)^T\), and the notations with the subscripts in SGTSVM also comply with this definition. Then, the first QPP \((13)\) is reformulated as

\[
\min_u f(u) = \frac{1}{2}||u||^2 + \frac{c_3}{m_1}||Z_1 u||^2 + \frac{c_2}{m_2}e^T(e + Z_2 u) + .
\tag{25}
\]

Next, we reformulate the \(t\)th \((t \geq 1)\) function in SGTSVM as

\[
f_t(u) = \frac{1}{2}||u||^2 + \frac{c_3}{m_1}||z_t u||^2 + c_2(1 + u^T z_t) + ,
\tag{26}
\]

where \(z_t\) and \(\hat{z}_t\) are the samples selected randomly from \(Z_1\) and \(Z_2\) for the \(t\)th iteration, respectively. The sub-gradient of \(f_t(u)\) at \(u_t\) is denoted as

\[
\nabla_t = u_t + c_1(u^T z_t)z_t + c_2z_t \text{sign}(1 + u^T \hat{z}_t) + .
\tag{27}
\]

Given \(u_1\) and the step size \(\eta_t = 1/t, u_{t+1}\) with \(t \geq 1\) is updated by

\[
u_{t+1} = u_t - \eta_t \nabla_t,
\tag{28}
\]

i.e.,

\[
u_{t+1} = (1 - \frac{1}{t})u_t - c_2z_t \text{sign}(1 + u_t^T \hat{z}_t) + .
\tag{29}
\]

**Lemma 3.1.** For all \(t \geq 1\), \(||\nabla_t||\) and \(||u_t||\) have the upper bounds.

**Proof.** The formation \((29)\) can be rewritten as

\[
u_{t+1} = A_t u_t + \frac{1}{t}v_t,
\tag{30}
\]

where \(A_t = \frac{1}{t}((t-1)I - c_1z_t^T)\), \(I\) is the identity matrix, and \(v_t = -c_2z_t \text{sign}(1 + u_t^T \hat{z}_t)\). Note that for sufficient \(t\), there is a positive integer \(N\) such that for \(t > N, A_t\) is positive definite, and the largest eigenvalue \(\lambda_t\) of \(A_t\) is smaller than or equal to \(\frac{1}{t}\). Based on \((30)\), we have

\[
u_{t+1} = \frac{t}{i=N_1} A_{t+N+1-i} u_{N+1} + \frac{t}{i=N_1} \frac{1}{i-N_1} (\frac{1}{t} A_{t+i+1-j} v_i).
\tag{31}
\]

For \(i \geq N + 1\), \(||A_{t+N+1-i} u_{N+1}|| \leq \lambda_t ||u_{N+1}|| \leq \frac{N}{t} ||u_{N+1}||\) \[33\]. Therefore,

\[
||\frac{t}{i=N_1} A_{t+i+1-j} u_{N+1}|| \leq \frac{N}{t} ||u_{N+1}||,
\tag{32}
\]
and 
\[ ||u_{t+1}|| \leq \frac{\widehat{G}}{\sum_{t=1}^{T} \xi_t} \max_{t \leq T} ||v_t||. \]  
(33)

Thus, we have 
\[ ||u_{t+1}|| \leq \frac{\widehat{G}}{\sum_{t=1}^{T} \xi_t} \max_{t \leq T} ||v_t|| \leq \frac{\xi_t}{\sum_{t=1}^{T} \xi_t} \max_{t \leq T} ||v_t||. \]  
(34)

Let \( M \) be the largest norm of the samples in the dataset and \( G_1 = \max\{\max(||u_1||, \ldots, ||u_T||), ||u_{N+1}|| + c_2 M\} \). This leads to that \( G_1 \) is an upper bound of \( ||u_t|| \), and \( G_2 = G_1 + c_1 G_1 M^2 + c_2 M \) is an upper bound of \( ||\nabla_t|| \), for \( t \geq 1 \). \( \square \)

**Theorem 3.1.** The iterative formation \( \text{sgt}_{\text{svm}} \) of our SGTSMV is convergent.

**Proof.** On the one hand, from \( \text{sgt}_{\text{svm}} \) in the proof of Lemma 3.1, we have 
\[ \lim_{t \to \infty} \left( \prod_{i=N+1}^{T} A_{i, N+1} \right) u_{N+1} = 0, \]  
(35)

which indicates 
\[ \lim_{t \to \infty} \left( \prod_{i=N+1}^{T} A_{i, N+1} \right) u_{N+1} = 0. \]  
(36)

On the other hand, from \( \text{sgt}_{\text{svm}} \), we have 
\[ \sum_{t=N+1}^{T} \left( \frac{1}{t} \prod_{j=N+1}^{T} A_{j, N+1} \right) v_i \leq M, \]  
(37)

which indicates that the following limit exists 
\[ \lim_{t \to \infty} \sum_{i=N+1}^{T} \left( \frac{1}{t} \prod_{j=N+1}^{T} A_{j, N+1} \right) v_i < \infty. \]  
(38)

Note that an infinite series of vectors is convergent if its norm series is convergent \( \text{sgt}_{\text{svm}} \). Therefore, the following limit exists 
\[ \lim_{t \to \infty} \sum_{i=N+1}^{T} \left( \frac{1}{t} \prod_{j=N+1}^{T} A_{j, N+1} \right) v_i < \infty. \]  
(39)

Combine \( \text{sgt}_{\text{svm}} \) with \( \text{sgt}_{\text{svm}} \), we conclude that the series \( u_{t+1} \) is convergent for \( t \to \infty \). \( \square \)

Based on the above theorem, it is reasonable to take the terminate condition to be \( ||u_{t+1} - u_t|| < \text{tol} \). Moreover, if we reform \( \text{sgt}_{\text{svm}} \) by \( u_t \), then 
\[ u_{t+1} = \prod_{i=1}^{T} A_{i, t+1} u_i + \sum_{i=1}^{t} \frac{1}{i} \prod_{j=i+1}^{T} A_{j, t+1} v_i. \]  
(40)

In order to keep \( u_{t+1} \) to be convergent fast, it is suggested to set \( u_0 = 0 \).

In the following, we analyse the relation between the solution \( u_t \) in SGTSMV and the optimal solution \( u^* = (w^*, b^*) \) in TBSVM.

**Lemma 3.2.** Let \( f_1, \ldots, f_T \) be a sequence of convex functions, and \( u_1, \ldots, u_{T+1} \in R^n \) be a sequence of vectors. For \( t \geq 1 \), \( u_{t+1} = u_t - \eta_t \nabla_t, \) where \( \nabla_t \) belongs to the sub-gradient set of \( f_t \) at \( u_t \) and \( \eta_t = 1/t \). Suppose \( ||u_1|| \) and \( ||\nabla_t|| \) have the upper bounds \( G_1 \) and \( G_2 \), respectively. Then, for all \( t \in R^n \), we have

(i) \( \frac{1}{T} \sum_{t=1}^{T} f_t(u_t) \leq \frac{1}{T} \sum_{t=1}^{T} f_t(\theta) + G_2 G_1 + \frac{1}{T} G_2^2 (1 + \ln T); \)

(ii) for sufficiently large \( T \), given any \( \varepsilon > 0 \), then 
\[ \frac{1}{T} \sum_{t=1}^{T} f_t(u_t) \leq \frac{1}{T} \sum_{t=1}^{T} f_t(\theta) + \varepsilon. \]

**Proof.** Obviously, \( f_t(1, \ldots, T) \) is convex. Let \( G_1 \) and \( G_2 \) respectively be the upper bounds of \( ||u_1|| \) and \( ||\nabla_t|| \), the conclusions come from Lemmas 3.1 and 3.2. \( \square \)

In the following, let us discuss the relation between the solutions to SGTSMV and TBSVM with the uniform sampling.
Corollary 3.1. Assume the conditions stated in Theorem 3.1 and $m_1 = m_2$, where $m_1$ and $m_2$ are the sample number of $X_1$ and $X_2$, respectively. Suppose $T = km_1$, where $k > 0$ is an integer, and each sample is selected $k$ times at random. Then

(i) $f(u_T) \leq f(u^*) + G_2(G_1 + ||u^*|| + G_2) + \frac{1}{k}G_1^2(1 + \ln T)$;
(ii) for sufficiently large $T$, given any $\varepsilon > 0$, then $f(u_T) \leq f(u^*) + G_2^2 + \varepsilon$.

Proof. First, we prove that for all $i, j = 1, 2, \ldots, T$,

$$|f_t(u_i) - f_t(u_j)| \leq G_2||u_i - u_j||, \quad t = 1, 2, \ldots, T.$$  \hspace{1cm} (45)

From the formation of $f_t(u)$, we have

$$|f_t(u_i) - f_t(u_j)| \leq \frac{1}{t}||u_i||^2 - ||u_j||^2| + \frac{1}{t^2}((u_i^T z_i)^2 - (u_j^T z_i)^2)| + c_2(1 + u_i^T \hat{z}_i + (1 + u_j^T \hat{z}_i))_+.$$  \hspace{1cm} (46)

Since $G_1$ is the upper bound of $||u_i||$ ($t \geq 1$) and $M$ is the largest norm of the samples in the dataset, the first part, the second part, and the third part on the right hand of (46) are respectively

$$\frac{1}{t}||u_i||^2 - ||u_j||^2| \leq G_1||u_i - u_j||,$$  \hspace{1cm} (47)

$$\frac{1}{t^2}((u_i^T z_i)^2 - (u_j^T z_i)^2) \leq c_1G_1M^2||u_i - u_j||,$$  \hspace{1cm} (48)

and

$$c_2(1 + u_i^T \hat{z}_i + (1 + u_j^T \hat{z}_i))_+ \leq c_2M||u_i - u_j||.$$  \hspace{1cm} (49)

Therefore, there is a constant $G_2 = G_1 + c_1G_1M^2 + c_2M$ satisfying (45).

From $u_{t+1} = u_t - \frac{1}{t}\nabla f_t$, it is easy to obtain

$$u_{t+1} = u_t - \sum_{i=1}^{t} \frac{1}{i} \nabla f_t, \quad t = 1, 2, \ldots, T.$$  \hspace{1cm} (50)

Thus, for $1 \leq i < j \leq T$,

$$||u_i - u_j|| = \frac{1}{2} \left| \sum_{i=1}^{j-1} \frac{1}{i} \nabla f_i \right| \leq \frac{1}{2} \sum_{i=1}^{j-1} \frac{1}{i} G_2.$$  \hspace{1cm} (51)

Since $T = km_1 = km_2$, for all $u \in R^n$, $\frac{1}{T} \sum_{i=1}^{T} f_t(u) = f(u)$. Note that $f(u)$ is the objective of TBSVM. Based on (45) and (51), we have

$$f(u_T) - \frac{1}{T} \sum_{i=1}^{T} f_t(u_i) = \frac{1}{T} \sum_{i=1}^{T} (f_t(u_T) - f_t(u_i)) \leq \frac{1}{T} \sum_{i=1}^{T} G_2||u_T - u_i|| \leq \frac{G_2^2(T-1)}{T} \leq G_2^2.$$  \hspace{1cm} (52)

Using the Theorem 3.1, we have the conclusion immediately.

If $m_1 \neq m_2$, we can modify the sampling rule to obtain the same result as one in Corollary 3.1.

Corollary 3.2. Assume the conditions stated in Corollary 3.1, but $m_1 \neq m_2$. Suppose $T = kd(m_1, m_2)$, where $k > 0$ is an integer and $d$ is the least common multiple of $m_1$ and $m_2$. The sample in $X_1$ is selected $kd/m_1$ times at random, and the one in $X_2$ is selected $kd/m_2$ times at random. Then

(i) $f(u_T) \leq f(u^*) + G_2(G_1 + ||u^*|| + G_2) + \frac{1}{k}G_1^2(1 + \ln T)$;
(ii) for sufficiently large $T$, given any $\varepsilon > 0$, then $f(u_T) \leq f(u^*) + G_2^2 + \varepsilon$.

Note that for all $u \in R^n$, $\frac{1}{T} \sum_{i=1}^{T} f_t(u) = f(u)$. The proof of the above corollary is the same as Corollary 3.1.

The above corollaries provide the approximations of $u^*$ by $u_T$. If the sampling rule is not as stated in these corollaries, these upper bounds no longer holds. However, Kakade and Tewari [33] have shown a way to obtain a similar bounds with high probability.

4 Experiments

In the experiments, we compared our SGTSVM with CSVM [11], LSSVM [26], SG SVM [24], TBSVM [8], and W LTSVM [10] on several artifical and benchmark datasets. CSVM was implemented by Libsvm [19] based on SMO algorithm on small sample size datasets (i.e., $m \leq 10,000$), while Liblinear [20] was implemented for CSVM based on trust region algorithm on large scale datasets (i.e., $m > 10,000$). TBSVM was solved by SOR algorithm [8]. All of the methods were implemented on a PC with an Intel Core Duo processor (3.4 GHz) with 4 GB RAM, while LSSVM, SG SVM, W LTSVM, and SGTSVM were implemented by Matlab [37]. For practical convenience, the corresponding SGT SVM Matlab codes uploaded in [http://www.optimal-group.org/Resource/SGT SVM.html](http://www.optimal-group.org/Resource/SGT SVM.html).

4.1 Artificial datasets

We first test our SGTSVM compared with TBSVM on two artificial datasets [9, 27] in $R^2$ (see Figures ?? and ??). The samples of two classes consist of uniform points are denoted as “+” and “x”, respectively. Figures ?? and ?? show the learning results, in which the parameters were all fixed by 1, and the iteration in SGTSVM was 1000. It is obvious that both SGTSVM and TBSVM obtain well classification results, and SGTSVM performs a little better than TBSVM on the second dataset. Therefore, in Figures ?? and ??, SGTSVM is effective as TBSVM.
4.2 Benchmark datasets

In order to analyze the convergent rate of our SGTSVM, it was tested on several small sample size datasets (see Table 1) and its parameters \(c_1, c_2, c_3,\) and \(c_4\) were selected from \(\{2^i | i = -8, \ldots, 7\} \). For nonlinear case, Gaussian kernel \(K(x_1, x_2) = \exp(-\mu |x_1 - x_2|^2)\) was used, and its parameter \(\mu\) was selected from \(\{2^i | i = -10, \ldots, 5\}\). For the optimal parameters, we depicted the iteration, accuracy (by ten-fold cross validation \(\[36\]\)), and learning time along with the parameter \(\text{tol}_{\mu}\) in Figures 2 and 4 for linear and nonlinear cases, respectively. From these figures, it is observed that: (i) SGTSVM terminates very fast for \(\text{tol} \geq 1e - 4\) but slow for \(\text{tol} < 1e - 4\); (ii) the converge rates of two problems in SGTSVM is generally different, but almost the same for \(\text{tol} \geq 1e - 4\); (iii) the accuracies are higher for smaller \(\text{tol}\), but almost do not increase when \(\text{tol} < 1e - 5\); (iv) the learning time is very fast for large \(\text{tol}\), but zooms when \(\text{tol} < 1e - 5\). Based on the above observations, \(\text{tol}\) in SGTSVM is suggested between \(1e - 5\) and \(1e - 4\). Anyone could set a smaller \(\text{tol}\) to obtain a higher accuracy together with more learning time.

Then, we compared SGTSVM with CSVM, LSSVM, SGSVM, TBSVM, and WLTSVM on these datasets. All of the regularization parameters \(c\) in CSVM, LSSVM and SGSVM, \(c_1, c_2, c_3,\) and \(c_4\) in TBSVM, WLTSVM, and SGTSVM were selected from \(\{2^i | i = -8, \ldots, 7\}\). The Gaussian kernel parameter \(\mu\) was selected from \(\{2^i | i = -10, \ldots, 5\}\). Table 2 shows the ten-fold cross validation accuracy and learning time by these classifiers for linear case, where the bold accuracy is the one that is much worse than others. It can be seen from the table that our SGTSVM has a similar accuracies as CSVM and LSSVM on these datasets, and their differences are no more than two percent. TBSVM obtains similar results as CSVM, LSSVM, and SGTSVM, though it cannot work on two datasets because of out of memory, where these datasets have more than five thousands samples. However, SGSVM and WLTSVM work worse than other classifiers, for SGSVM has a much lower accuracies than the highest classifier on four datasets and WLTSVM on three datasets (these results are bolded). It also can be seen from the table that LSSVM, SGSVM, WLTSVM, and SGTSVM learn a little faster than CSVM and much faster than TBSVM on the datasets that over one thousand samples. Table 3 shows the accuracy and learning time of these classifiers for nonlinear case. The conclusions about accuracy are similar as the linear case. In addition, the learning time of CSVM, LSSVM, TBSVM, and WLTSVM increases much more than the linear case except SGSVM and SGTSVM, indicating the SGD-based methods are stable on saving time.

4.3 Large scale datasets

To test the feasibility of these methods on large scale datasets, we ran them on three large scale datasets: “C0DRNA”, “SKIN”, and “SUSY” \(\[20\]\). Thereinto, CODRNA includes 59,555 training samples and 271,618 testing samples with 8 features, SKIN includes 245,057 samples with 3 features, and SUSY includes 5,000,000 samples with 18 features. We split SKIN and SUSY into two sets, where one set including 20% samples is used for training, and the other including 80% samples is used for testing. Due to the nonlinear cases spend too much time to learn on these datasets, we just considered the linear methods. Moreover, since LSSVM, TBSVM, and WLTSVM are out of memory on all of these datasets, the comparisons only include CSVM, SGSVM, and our SGTSVM, where CSVM is implemented by Liblinear \(\[20\]\). The iteration of SGTSVM is set to 10,000. To speed up the learning rate of SGTSVM, we set \(\text{tol} = 1e - 3\) and add another paraactive terminate condition that the iteration is no more than 10,000. Table 3 shows the ten-fold cross validation and testing accuracies, learning time, and required memory in training on these datasets. It is obvious that SGTSVM owns the highest accuracies on testing, and is as fast as SGSVM. In addition, our SGTSVM is much faster than CSVM with less memory. In detail, CSVM need store the entire training set in RAM, while SGSVM and SGTSVM only store a subset related to the iteration. Due to the required memory of CSVM increases with the size of dataset, it tends to out of memory with the increasing data size, while the same thing does not appear in SGSVM and SGTSVM, which indicates SGTSVM has a better generalization than CSVM.

5 Conclusion

The stochastic gradient twin support vector machine (SGTSVM) based on stochastic gradient decent algorithm (SGD) has been proposed. As our knowledge, it is the first time that SGD is introduced for TSVM-type classifiers. In addition, according to our experiments, our method is the fastest one among the TSVM-type classifiers on large scale datasets. By hiring the nonparallel hyperplanes, SGTSVM is more stable on sampling than SGSVM. In theory, SGTSVM is convergent, and is an approximation of TSVM with uniform sampling. Experimental results on several public available datasets have indicated that our SGTSVM has comparable accuracy compared with other TSVM-type classifiers, but with the fastest learning speed. For practical convenience, the corresponding SGTSVM Matlab codes can be downloaded from [http://www.optimal-group.org/Resource/SGTSVM.html](http://www.optimal-group.org/Resource/SGTSVM.html). For the future work, it is possible to design and study other SGD-based nonparallel hyperplanes SVM model.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 11501310, 11201426, and 11371365), the Natural Science Foundation of Inner Mongolia Autonomous Region of China (No. 2015BS0606), and the Zhejiang Provincial Natural Science Foundation of China (No. LY15F030013).

References

[1] C. Cortes and V. Vapnik, “Support vector networks,” Machine Learning, vol. 20, pp. 273–297, 1995.
[2] O. Mangasarian, Nonlinear Programming. SIAM, 1994.
[3] C. Zhang, Y. Tian, and N. Deng, “The new interpretation of support vector machines on statistical learning theory,” Science China, vol. 53, no. 1, pp. 151–164, 2010.
[4] W. Noble, “Support vector machine applications in computational biology,” in Kernel Methods in Computational Biology, Cambridge, 2004.
Fig. 3. The influences of parameter \( tol \) on iteration, accuracy, and learning time on the benchmark datasets for linear SGTSVM.
Table 3
The results on the benchmark datasets for nonlinear case.

| Data     | CSVM Time (Sec.) | LSSVM Time (Sec.) | SGSVM Time (Sec.) | TBSVM Time (Sec.) | WLTSVM Time (Sec.) | SCTSVM Time (Sec.) |
|----------|------------------|-------------------|-------------------|-------------------|-------------------|--------------------|
| Australian 690×14 | 86.28±0.32        | 87.10±0.26        | 86.10±0.42        | 87.10±0.43        | 87.14±0.32        | 85.21±0.16         |
| CMC 1,473×9  | 2.7363            | 0.2104            | 0.0495            | 0.9043            | 4.3916            | 0.1377             |
| Creadit 690×15 | 0.0279            | 0.0351            | 0.0114            | 0.1780            | 0.1120            | 0.0304             |
| German 1,000×20 | 0.8429            | 0.0854            | 0.0281            | 0.8935            | 0.2464            | 0.0736             |
| Hypothyroid 3,163×25 | 0.1509         | 0.1089            | 0.0112            | 5.8786            | 6.1016            | 0.0253             |
| Pimaindian 768×8  | 0.3243            | 0.0484            | 0.0167            | 2.097             | 0.1171            | 0.0419             |
| Ring 7,400×20 | 0.6827            | 0.2628            | 0.0153            | *                 | *                 | 0.1282             |
| TIC 5,822×85  | 3.5700            | 0.2525            | 0.0139            | *                 | *                 | 0.1301             |
| Titanic 2,201×3  | 0.4455            | 0.1174            | 0.0091            | 1.768             | 1.6229            | 0.0220             |
| Two 7,400×20 | 9.783±0.03        | 9.81±0.02         | 9.56±0.07         | *                 | *                 | 9.74±0.07          |

*Out of Memory
Y. Shao and N. Deng, “A coordinate descent margin based-
Z. Wang, Y. Shao, and T. Wu, “A ga-based model selection for
Y. Shao, W. Chen, J. Zhang, Z. Wang, and N. Deng, “An efficient
D. Li, Y. Tian, and H. Xu, “Deep twin support vector machine,” in
M. Bazarra, H. Sherali, and C. Shetty, Nonlinear Programming—the
R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, “LIBLINEAR: a library for large linear classification,” in Advances in neural information processing systems, 2008.
C. Blake and C. Merz, UCI Repository for Machine Learning Databases, http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.
R. Duda, P. Hart, and D. Stork, Pattern Classification, 2nd Edition. John Wiley and Sons, 2001.

| TABLE 4 | The results on the large scale datasets. |
|---------|-----------------------------------------|
| Data    | SVM Validation | SGSVM Validation | SGT SVM Validation |
|         | Testing        | Time(Sec.)       | Testing          |
|         | Time(Sec.)     | Memory           | Testing          |
| CODRNA  | 90.76          | 0.0518           | 0.1352           |
| 59,535 × 8 | 91.78          | 0.1100           | 0.1591           |
| 271,618 × 8 | 0.8520         | <1MB             | <1MB             |
| SKIN    | 93.40          | 93.90            | 94.07            |
| 49,011 × 3 | 90.40          | 90.23            | 94.07            |
| 196,046 × 3 | 0.6180         | 0.0990           | 0.1417           |
| SUSY    | 76.64          | 70.11            | 76.60            |
| 1,000,000 × 18 | 78.28        | 70.86            | 78.32            |
| 5,000,000 × 18 | 13.8010       | 0.1500           | 0.2002           |
| 330MB   | <1MB           | <1MB             | <1MB             |

[5] T. Lal, M. Schröder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Schölkopf, “Support vector channel selection in BCI,” Data Mining and Knowledge Discovery, vol. 15, no. 2, pp. 1003–1010, 2004.
[6] H. Ince and T. Trafalis, “Support vector machine for regression and applications to financial forecasting,” in International Joint Conference on Neural Networks, Istanbul, 2002, pp. 6348–6354.
[7] Jayadeva, R. Khemchandani, and S. Chandra, “Twin support vector machines for pattern classification,” IEEE Trans. Pattern Anal. Machine Intell., vol. 29, no. 5, pp. 905–910, 2007.
[8] Y. Shao, C. Zhang, X. Wang, and N. Deng, “Improvements on twin support vector machines,” IEEE Transactions on Neural Networks, vol. 22, no. 6, pp. 962–968, 2011.
[9] X. Peng, “TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition,” Pattern Recognition, vol. 44, no. 10–11, pp. 2678–2692, 2011.
[10] Y. Shao, W. Chen, J. Zhang, Z. Wang, and N. Deng, “An efficient weighted lagrangian twin support vector machine for imbalanced data classification,” Pattern Recognition, vol. 47, no. 9, pp. 3158–3167, 2014.
[11] Y. Shao and N. Deng, “A coordinate descent margin based twin support vector machine for classification,” Neural Networks, vol. 25, pp. 114–121, 2012.
[12] Z. Wang, Y. Shao, and T. Wu, “A ga-based model selection for smooth twin parametric-margin support vector machine,” Pattern Recognition, vol. 46, no. 8, pp. 2267–2277, 2013.
[13] D. Li, Y. Tian, and H. Xu, “Deep twin support vector machine,” in Data Mining Workshop (ICDMW), 2014 IEEE International Conference on. IEEE, 2014, pp. 65–73.
[14] Z. Wang, Y. Shao, L. Bai, and N. Deng, “Twin support vector machine for clustering,” IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 10, pp. 2583–2588, 2015.
[15] W. Chen, Y. Shao, C. Li, and N. Deng, “Mlsvm: A novel twin support vector machine to multi-label learning,” Pattern Recognition, vol. 52, pp. 61–74, 2015.
[16] M. Bazarra, H. Sherali, and C. Shetty, Nonlinear Programming: Theory and Algorithms, 2nd ed. Wiley, 2004.
[17] J. Platt, “Fast training of support vector machines using sequential minimal optimization,” in Advances in kernel methods—support vector learning, Cambridge, MA: MIT Press, 1999, pp. 185–208.
[18] T. Joachims, “Making large-scale svm learning practical,” in Advances in Kernel Methods—Support Vector Learning, Cambridge, 1998, pp. 169–184.
[19] C. Chang and C. Lin, LIBSVM: A library for support vector machines, http://www.csie.ntu.edu.tw/~cjlin 2001.
[20] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, “LIBLINEAR: a library for large linear classification,” Journal of Machine Learning Research, vol. 9, pp. 1871–1874, 2008.
[21] Y. Tian and Y. Ping, “Large-scale linear nonparallel support vector machine solver,” Neural Networks, vol. 50, pp. 166–174, 2014.
[22] J. Kivinen, A. Smola, and R. Williamson, “Online learning with kernels,” Signal Processing, IEEE Transactions on, vol. 52, no. 8, pp. 2165–2176, 2004.