http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (6): 850-858, 2008
ISSN 1812-5638

A Unified Model for Privacy-Preserving Support Vector Machines on Horizontally and Vertically Partitioned Data

Fubo Shao, Hua Duan, Guoping He and Xin Zhang
College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong, People's Republic of China

Abstract: We propose a novel unified model for Privacy-Preserving Support Vector Machines (PPSVM for short) classifier on horizontally and vertically partitioned data. We prove the feasibility of the model. Besides we give out the algorithms for horizontally partitioned data and vertically partitioned data, respectively. The columns of data matrix A represent input features and the rows represent the individual data which is called a training/testing point in SVM. For horizontally partitioned data, the data matrix A whose rows including all input features are divided into groups belonging to different entities. While for vertically partitioned data, the data matrix A's columns are divided into groups belonging to different entities. Each entity is unwilling to share its group of data or leak the data for various reasons. The proposed SVM classifiers are public but do not reveal any private data. And when we calculate the classifier at last, we do not need to recover the original data. Besides, it has comparable accuracy with that of an ordinary SVM classifier that uses the centralized data set directly. Experiments show that our approach is effective.

Key words: Privacy-preserving classification, support vector machines, vertically partitioned data, horizontally partitioned data

INTRODUCTION

With the development of technology, ultra large databases appear. At the same time, people show much concerns about the informational privacy. The problem of privacy-preserving is first proposed by Agrawal and Srikant (2000). Data mining is an efficient tool to discover valuable knowledge from a great deal data. But the general data mining is based on the assumption that complete access to data is available, either in centralized or federated form. In fact, privacy and security concerns often prevent sharing of data which may not be possible due to either legal or commercial reasons. In legal terms, medical data cannot be released for any purpose without appropriate anonymization. In commercial terms, data is often a valuable business asset. So, the research direction in data mining incorporating privacy concerns becomes fruitful. People also show much interest in the problem of privacy-preserving data mining. The environment of data mining can be classified into two cases: centralized and distributed data. In the centralized environment, the focus is on the query restriction. In the distributed environment, the data is distributed in different sites. Now, in the distributed environment, the data is classified into horizontally partitioned and vertically partitioned case. For horizontally partitioned data (Yu *et al*., 2006a; Kantarcioglu and Clifton, 2004), the data matrix A's rows including all input features are divided into groups belonging to different entities. While for vertically partitioned data, the data matrix A's columns are divided into groups belonging to different entities (Yu *et al*., 2006b; Vaidya and Clifton, 2002). The reason is that feature values for each individual are stored as rows of a data matrix, while a specific feature values for all individuals are represented by columns of a data matrix.

Data mining has many applications in the real world. One of the most important and widely found problems is that of classification. The goal of classification is to build a model that can predict the value of one variable, based on the other variables. Support Vector Machine is one of the most important classification methods in machine learning. While SVM is one of the most actively developed classification methodology, so there has been wide interest in privacy-preserving support vector machine (PPSVM) classifier. Recently, people show much interest in the area of privacy-preserving support vector machines, PPSVM for short. We briefly review some of the relevant work. PPSVM classifiers were gained on vertically partitioned data by adding random perturbations to the data by Yu *et al*. (2006b). PPSVM

Corresponding Author: Fubo Shao, College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong, People's Republic of China

classifiers using nonlinear kernels on horizontally partitioned data were obtained by Yu *et al.* (2006a). But it can only deal with binary feature data. Other privacy preserving classifying techniques include wavelet-based distortion and rotation perturbation (Chen and Liu, 2005).

The kernel matrix computed in the above study is equal or close to that computed with the original data. They refer to compute inner product securely (Ioannidis *et al.*, 2002). In this study, we propose another approach to the classification and testing. The kernel matrix computed in our method is not equal to that computed with original data. But the accuracy is comparable with the original data. We propose a unified high efficient novel model of PPSVM classifiers on horizontally and vertically partitioned data that is different from the existing PPSVM classifiers for such partitioned data. During the classification and testing, we use the perturbed data. The study is based on the two ideas. The first idea is that for a given data matrix A, we add the same random real number to the same colunin of the matrix A. Though the rows of the matrix A are modified, the relative location of each row of matrix A is not changed. We prove that this model has comparable accuracy with the original data. Besides, this experiment gives the evidence. The second idea is that each entity generates a random row vector that has the same dimension with that of its input feature space. The entity holds it privately. In the training and testing process, it does not change. The data for training and testing is perturbed with the same data. By employing the two ideas, we shall describe algorithms that protect the privacy of each partitioned data either horizontally partitioned or vertically partitioned. The generated PPSVM classifier has comparable accuracy comparing to that of an ordinary SVM classifier.

We first describe the notation. An m×n matrix A represents m data points in an n-dimensional input space. An m×n diagonal matrix D contains the corresponding labels (i.e., +1 or -1) of the data points in A. (A class label D_{ii}, or d_i for short, corresponds to the i-th data points x_i in A). All vectors are colunin vectors unless transposed to a row vector by a prime superscript^T. The scalar (inner) product of two vectors x and y in the n-dimensional space R^n is denoted by x^Ty.

We assume that we can public the class labels D corresponding to data matrix A. Each entity does not collude and does follow the proposed protocol correctly.

SVM OVERVIEW

The Support Vector Machine (SVM) is a classifier, originally proposed by Vapnik (2000), that finds a maximal margin separating hyperplane between two classes of data. The aim of Support Vector Classification (SVC) is to devise a computationally efficient way of learning good separating hyperplane in a high dimensional feature space. SVM finds the separating hyperplane ((ω.x)+b = 0) that maximizes the margin, denoting the distance between the hyperplane and the closest data points (i.e., support vectors). When we can not find a hyperplane to separate the data perfectly, we introduce the soft margin. To maximize the margin while minimizing the error, the standard SVM solution is formulated into the following primal program:

$$\begin{aligned} &\min\ \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}\xi_i^2 \\ &\text{s.t. } d_i((\omega\cdot x_i)+b)+\xi_i \geq 1, i=1,2,..,l, \end{aligned} \tag{1}$$

ξ_i is the slack variable in the constraint. This shows that SVM allows error or the soft margin. The slack or error is minimized in the objective function. C is the margin parameter which is used to tune the margin size and the error. The weight vector ω and the bias b will be computed by this optimization problem. Then we can determine the class of a new data object x by f(x) = (ω.x)+b, where the class is positive if f(x)>0, or else negative.

To solve the primal problem, we can solve its dual problem by applying the Lagrange multipliers:

$$\begin{aligned} &\min\ \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} d_i d_j \alpha_i \alpha_j (K(x_i,x_j)+\frac{1}{C}\delta_{ij}) - \sum_{i=1}^{l}\alpha_i \\ &\text{s.t. } \sum_{i=1}^{l} d_i\alpha_i = 0, \\ &\quad \alpha_i \geq 0, i=1,2,..,l, \end{aligned} \tag{2}$$

The coefficients α are to be computed from the dual problem. Where

$$\delta_{ij}=\begin{cases}1 & i=j,\\ 0 & \text{otherwise,}\end{cases} \quad K(x_i,x_j)$$

is the kernel function where $K(x_i,x_j)=x_i^Tx_j$ for linear kernel. We can also apply a nonlinear kernel for $K(x_i, x_j)$ (e.g.,

$$K(x_i,x_j)=\exp(-\frac{\|x_i-x_j\|^2}{g})$$

for RBF kernel, $K(x_i \cdot x_j) = ((x_i \cdot x_j))+1^p$ for polynomial kernel.) The weight vector $\omega=\sum\alpha_i d_i x_i$ and

$$b=-\frac{1}{2}(\sum_{d_i=1}\alpha_i x_i - \sum_{d_i=-1}\alpha_i x_i)\cdot(\sum_{d_i=1}\alpha_i x_i + \sum_{d_i=-1}\alpha_i x_i)$$

Then the classification function $f(x) = \text{sgn}(\sum \alpha_i d_i K(x_i \cdot x) + b)$ can be computed.

For more information, see the tutorial written by Burges (1998).

A UIFIED MODEL FOR PPSVM

If we can access the data points xi, i = 1,2,.., *l* freely, we can get the original SVM model:

$$\begin{aligned} &\min \ \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{1}\xi_i \\ &\text{s.t. } d_i((\omega \cdot x_i) + b) + \xi_i \geq 1, i = 1,2,..,l, \\ &\quad \xi_i \geq 0 \end{aligned} \tag{3}$$

Solving the problem, we can use the method earlier. While privacy and security prevent access the data points ξ_i, i = 1,2,.., 1 freely. One technique for privacy and security is perturbation (Agrawal and Ramakrishnan, 2000), that is perturbs the data points with some data. In this research, we give the same bias to the same column, that is each data point has the same bias. In the privacy-preserving support vector machines, privacy and security prevent to access the original data.

In present model, the original data points $x_i \in R^n$, i = 1,2,.., *l* become x_i+a, where $a \in R^n$. We can access the data points x_i+a, i = 1,2,.., *l*. The purpose of introducing vector a is to realize the requirement of privacy and security. The PPSVM model is as following.

$$\begin{aligned} &\min \ \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{1}\xi_i \\ &\text{s.t. } d_i((\omega \cdot x_i + a) + b) + \xi_i \geq 1, i = 1,2,..,l, \\ &\quad \xi_i \geq 0 \end{aligned} \tag{4}$$

We give the proof that using the PPSVM model has comparable accuracy comparing with that of the original SVM model.

Lemma 1: In the original SVM model (3) and the PPSVM model (4), the distance of the data points x_i, x_j is the same with that of the corresponding perturbed data points x_i+a,x_j+a.

Lemma 2: The original SVM model Eq. 3 is equivalent to:

$$\begin{aligned} &\min \ \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{1}\xi_i^2 \\ &\text{s.t. } d_i((\omega \cdot x_i) + b) + \xi_i \geq 1, i = 1,2,..,l, \end{aligned} \tag{5}$$

Similarly, the PPSVM model Eq. 4 is equivalent to:

$$\begin{aligned} &\min \ \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{1}\xi_i^2 \\ &\text{s.t. } d_i((\omega \cdot x_i + a) + b) + \xi_i \geq 1, i = 1,2,..,l, \end{aligned} \tag{6}$$

Lemma 3: Given the same parameter C, solving the original SVM model (5), we get the decision function f(x) = (ω·x)+b and solving the PPSVM model we get the decision function $\tilde{f}(x) = (\tilde{\omega} \cdot x) + \tilde{b}$, then we can get $\omega = \tilde{\omega}$ for linear kernel. For the original SVM model (5), the Lagrange multipliers are α_i, i = 1,2,..., *l*. For the PPSVM model (6), the Lagrange multipliers are $\tilde{\alpha}_i$, i = 1,2,...,*l*. If the slack variable ξ_i is same in the original model (5) and PPSVM model (6) (In Lemma 4, we proof that the assumption is reasonable), then $\alpha_i = \tilde{\alpha}_i$ for both linear and nonlinear kernels.

Proof: For the original SVM model (5), the Lagrange function is:

$$L(\omega, b, \xi) = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{1}\xi_i^2 - \sum_{i=1}^{1}\alpha_i(d_i((\omega \cdot x_i) + b) - 1 + \xi_i) \tag{7}$$

where α_i are Lagrange multipliers. Its KKT conditions are:

$$\nabla_b L(\omega, b, \xi) = \sum_{i=1}^{1}\alpha_i d_i = 0, i = 1,2,...,l \tag{8}$$

$$\nabla_\omega L(\omega, b, \xi) = \omega - \sum_{i=1}^{1}\alpha_i d_i x_i = 0, i = 1,2,...,l \tag{9}$$

$$\nabla_\xi L(\omega, b, \xi) = C\xi - \alpha = 0 \tag{10}$$

$$\alpha_i(1 - \xi_i - d_i(\omega \cdot x_i) + b) = 0, i = 1,2,...,l \tag{11}$$

$$\alpha_i \geq 0, i = 1,2,...,l \tag{12}$$

$$d_i(\omega \cdot x_i) + b) \geq 1 - \xi_i, i = 1,2,...,l \tag{13}$$

We replace original data points x_i, i = 1,2,...,*l* with $x_i + \alpha_i$, i = 1,2,...,*l* and replace ω, x_i,α_i,b with $\tilde{\omega}, \tilde{x}_i, \tilde{\alpha}_i, \tilde{b}$ in the Eq. 8-13. We get the KKT conditions of the PPSVM model (6).

From Eq. 8-9, we get

$$\tilde{\omega} = \sum_{i=1}^{1}\tilde{\alpha}_i d_i(x_i + a).$$

For linear kernel:

$$\tilde{\omega} = \sum_{i=1}^{1}\tilde{\alpha}_i d_i x_i + a\sum_{i=1}^{1}\tilde{\alpha}_i d_i = \sum_{i=1}^{1}\alpha_i d_i x_i = \omega$$

So, ω does not change for linear kernel.

If the slack variable ξ_i is same in the original model (5) and PPSVM model (6), the parameter C is the same in (5) and (6), from the Eq. 10, we can know that the Lagrange multipliers do not change from the Eq. 10. That is:

$$\alpha_i = \tilde{\alpha}_i, i = 1,2,..,l \tag{14}$$

From Eq. 14, we know that if x_i is support vector for original SVM model, then x_i+a is support vector for PPSVM model.

Lemma 4: If

$$\underline{b} = \max\{b + (\omega \cdot x_i) - (\omega \cdot (x_i + a)) \mid d_i = 1\}$$
$$\leq \bar{b} = \min\{b + (\omega \cdot x_i) - (\omega \cdot (x_i + a)) \mid d_i = -1\}.$$

(ω,b,ξ) is the optimal solution of the original SVM model Eq. 5, then there is $\tilde{b}$ such that $(\omega,\tilde{b},\xi)$ is the feasible solution of the PPSVM model Eq. 6.

Proof: Because $\underline{b} \leq \bar{b}$ so there is $\tilde{b}$ subjected to $\underline{b} \leq \tilde{b} \leq \bar{b}$. So:

$$d_i \tilde{b} \geq d_i(b + (\omega \cdot x_i) - (\omega \cdot (x_i + a))).$$
$$d_i((\omega \cdot x_i) + b) \leq d_i((\omega \cdot (x_i + a)) + b).$$

(ω,b,ξ) is the optimal solution of the original SVM model Eq. 5, so $d_i((\omega \cdot x_i) + b) + \xi_i \geq 1$. Then $d_i((\omega \cdot (x_i + a)) + b) + \xi_i \geq 1$. So $(\omega,\tilde{b},\xi)$ is the feasible solution of the PPSVM model Eq. 6.

Theorem 1: For linear kernel, given the same parameter C, the decision function f (x) = (ω·x)+b got from the original SVM model has the same function value with the decision function $\tilde{f}(x + a) = (\tilde{\omega} \cdot (x + a)) + \tilde{b}$.

Proof: From Lemma 3, we know

$$\omega = \tilde{\omega},\ b = \frac{1 - \xi_i}{d_i} - (\omega \cdot x_i).$$

Then for any i, α_i>0:

$$\begin{aligned} f(x) &= (\omega \cdot x) + b \\ &= (\omega \cdot x) + \frac{1 - \xi_i}{d_i} - (\omega \cdot x_i) \\ &= (\omega \cdot (x - x_i)) + \frac{1 - \xi_i}{d_i} \end{aligned} \tag{15}$$

$$\begin{aligned} \tilde{f}(x + a) &= (\tilde{\omega} \cdot (x + a)) + \tilde{b} \\ &= (\omega \cdot (x + a)) + \frac{1 - \xi_i}{d_i} - (\omega \cdot (x_i + a)) \\ &= (\omega \cdot (x - x_i)) + \frac{1 - \xi_i}{d_i} = f(x) \end{aligned} \tag{16}$$

$\forall x, f(x) = f(x + a)$. So we get Theorem 1.

Suppose from the original SVM model (5), we get the support vectors are x_{si}, i = 1,2,...,k. We give the notations. x_{imax}, i = max $\{x_{sij}, i = 1,2,...,k\}$, x_{jmin}, i = max $\{x_{sij}, i = 1,2,...,k\}$, j =1,2,..,n, where x_{sij} is the j-th element of vector x_{si}. We define $x_{max} = (x_{1max}, x_{2max},..., x_{nmax})$, $x_{min} = (x_{1min}, x_{2min},..., x_{nmin})$.

Theorem 2: Given the same parameter C in the PPSVM model and the original SVM model. We get the classification function f(x) = sgn((ω·x)+b) solving the original SVM model. We get the classification function $\tilde{f}(x + a) = \text{sgn}((\tilde{\omega} \cdot x) + \tilde{b})$. The original data distribution function is P(x), its mean value is μ and standard dubitation is σ.

Under the assumption that the slack variable ξ_i is same in the original model (5) and PPSVM model (6), for 0-1 loss function c(x,d,f(x)), the difference of the expect risk of classification function f(x) and $\tilde{f}(x)$ is not more than $P(x_{max})$-$P(x_{min})$. Then we say that the PPSVM model has comparable accuracy comparing with the original SVM model.

Proof: From Lemma 3, we know that under the assumption the corresponding support vectors do not change, that is if x_i is a support vector then x_i+a is a support vector, too. From lemma 3, we know that the difference of classification of f(x) and $\tilde{f}(x + a)$ only may be the data points between the support vectors. Then difference of expect risk of decision functions $f(x),\tilde{f}(x)$ is:

$$\begin{aligned} &|R(f) - R(\tilde{f})| \\ &= \left|\int_{X \times Y} (c(x,d,f(x)) - c(x,d,\tilde{f}(x)))dP(x)\right| \\ &\leq \left|\int_{x_{min}}^{x_{max}} dP(x)\right| = |P(x_{max}) - P(x_{min})| \end{aligned} \tag{17}$$

We get that the PPSVM model has comparable accuracy comparing with the original SVM model.

PRIVACY-PRESERVING METHODS AND QUANTIFYING PRIVACY

Present basic approach to preserving privacy is to let users provide a modified value of its original data. We consider the technique for modifying values:

Table 1: Quantifying privacy

	Confidence (%)		
Distrbution n	50	95	99.9
Uniform	0.5×2α	0.95×2α	0.99×2α
Gaussian	1.34×σ	3.92×σ	6.8×σ

Value perturbation: Return a value x_i+r instead of $x_i \epsilon R$ where r is a random value drawn from certain distribution. We consider mainly two random distributions. In fact, we can draw r from more random distributions which can improve the privacy:

- **Uniform:** The random variable has a uniform distribution, between [-α, α]. The mean of the random variable is 0.
- **Gaussian:** The random variable has a normal distribution, with mean $\mu = 0$ and standard deviation σ.

Table 1 shows the privacy preserved by the methods of Uniform and Gaussian. We fix the perturbation of an entity. So, it has the same bias for the same attribute of all data points.

We use the same quantifying methods with the study by Agrawal and Srikant (2000). For quantifying privacy provided by a method, we use a measure based on how closely the original values of a modified attribute can be estimated. If it can be estimated with c% confidence that a value x lies in the interval $[x_1, x_2]$, then the interval width x_1, x_2 defines the amount of privacy at c% confidence level.

Table 1 shows the privacy offered by the different methods using this metric.

ALGORITHMS OF PRIVACY-PRESERVING SUPPORT VECTOR MACHINES

Before give the algorithms, we define a notation firstly.

Definition 1 ⊕: If data matrix $A = (x_1^T, x_2^T, ..., x_n^T) \in R^{m \times n}$ and vector $a = (a_1, a_2, ..., a_n)^T$, $a_i \in R, i = 1, 2, ..., n$, then $A \oplus a = ((x_1+a))^T, (x_2+a)^T, ..., (x_n+a)^T$.

Lemma 5: For two vectors $a,b \epsilon R^n$ and $A \epsilon R^{m \times n}$, then $A \oplus a \oplus b = A \oplus b \oplus a$.

Algorithm of PPSVM on horizontally partitioned data: The dataset using to obtain a classifier consists of m points in R^n represented by the m rows of the matrix $A \epsilon R^{m \times n}$. Each row contains values for n features associated with a specific individual, while each column contains m values of a specific feature associated with m different individuals. The data matrix A is divided into q blocks of $m_1, m_2, ..., m_q$ rows with $m_1+m_2+...+m_q = m$, $A = [A_1^T A_2^T ... A_q^T]^T$. The blocks of matrix A are held by q entities $P_1, P_2, ..., P_q$, respectively. That is the entity P_i owns the data A_i. Each entity is unwilling to make public or share its data with others. Furthermore, they do not reveal its data for various reasons such as commercial reason. Data is often a valuable business asset for a factory. Each row of $A \epsilon R^{m \times n}$ is labeled belonging to the classes by a corresponding diagonal matrix $D \epsilon R^{m \times n}$. Class label D_{ii}, or d_i for short, corresponds to the i-th data points x_i^T in A. We assume that entities make public the matrix D.

In the horizontally partitioned case, we need an entity P_0 that is independent from the participating entities $P_1, P_2, ..., P_q$. It actually owns no data and its task is only computing the global classification function. It is the protocol initiator. Each entity P_i, except entity P_0, generates a random vector r_i, i = 1,2,..., q having the same dimension with the row vector dimension of data matrix A_i. The random vector r_i is privately held by the entity and is never made public and keeps the same during the procedure of training and testing. The elements of these random vectors are drawn from the Uniform distribution or Gaussian distribution or other distributions. Each row of the data owned by all entities is added to the sum of random vectors r_i, i = 1,2,..., q. Then the data matrix A becomes $A \oplus r_1 \oplus r_2 \oplus ... \oplus r_q$. After that, we can use the data set and the PPSVM model to compute the classification function $f(x) = sgn((\omega \cdot x)+b)$. When one entity has a new data point x whose class label the entity wants to know. The initiator P_0 can predict its label using the data $x+r_1+r_2+...+r_q$.

Now we completely describe algorithms for horizontally partitioned data using PPSVM model. The whole algorithm is described as Algorithm 1 and Algorithm 2.

ALGORITHM 1

- **Step 0 (initialize):** P_0 is the protocol initiator. The entity P_i (i = 1,2,..,q) generates its random vector r_i and holds it privately. The vector has the same dimension with the row of its dataset owned and keeps the same during the procedure of training and test.
- **Step 1:** The first step begins by the initiator P_0 sending an empty dataset to its neighbor P_1. Note that, to prevent P_0 from contributing to training set, P_1 must reject P_0's start request if P_0 sends a non-empty set in the first round.

- **Step 2:** For i = 1,2,...q-1 entity P_i sends the dataset $[A_1^T \oplus r_1 \oplus ... \oplus r_i \;\; A_2^T \oplus r_2 \oplus ... \oplus r_i \;\; ... \;\; A_i^T \oplus r_i]^T$ to entity P_{i+1}.
- **Step 3:** P_q send the dataset

$$[A_1^T \oplus r_1 \oplus ... \oplus r_q \;\; A_2^T \oplus r_2 \oplus ... \oplus r_q \;\; ... \;\; A_q^T \oplus r_q]^T$$

to P_0.
- **Step 4:** P_0 removes $A_1^T \oplus r_1 \oplus ... \oplus r_q$ and send the remaining dataset $[A_2^T \oplus r_2 \oplus ... \oplus r_q \;\; ... \;\; A_i^T \oplus r_q]^T$ to entity P_1.
- **Step 5:** For i = 1,2,...,q-2, P_i sends the dataset $A_{i+1}^T \oplus r_{i+1} \oplus ... \oplus r_q \oplus r_1 \oplus ... \oplus r_i$ to P_0 and sends

$$[A_{i+2}^T \oplus r_{i+2} \oplus ... \oplus r_q \oplus r_1 \oplus ... \oplus r_i$$
$$A_{i+3}^T \oplus r_{i+3} \oplus ... \oplus r_q \oplus r_1 \oplus ... \oplus r_i \;\; ...$$
$$A_q^T \oplus r_q \oplus r_1 \oplus ... \oplus r_i]^T$$

to P_{i+1}.
- **Step 6:** P_{q-1} sends $A_q^T \oplus r_q \oplus r_1 \oplus ... \oplus r_{q-1}$ to P_0.
- **Step 7:** P_0 gets dataset

$$[A_1^T \oplus r_1 \oplus ... \oplus r_q$$
$$A_2^T \oplus r_1 \oplus ... \oplus r_q \;\; ... \;\; A_q^T \oplus r_1 \oplus ... \oplus r_q]^T$$

(Lemma 5) . Go to Algorithm 2.

ALGORITHM 2

- **Step 0 :** P_0 now owns the dataset

$$[A_1^T \oplus r_1 \oplus ... \oplus r_q$$
$$A_2^T \oplus r_1 \oplus ... \oplus r_q \;\; ... \;\; A_q^T \oplus r_1 \oplus ... \oplus r_q]^T.$$

All entities make public the class matrix $D_{ll} = \pm 1$, l = 1,2,..,q for the data matrices A_i, i = 1,2,..q.
- **Step 1:** Solve the PPSVM's dual problem

$$\min_{\alpha} \;\; \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} d_i d_j \alpha_i \alpha_j K(x_i + r_1 + r_2 + ... + r_q) - \sum_{i=1}^{m} \alpha_i$$
$$\text{s.t.} \;\; \sum_{i=1}^{m} d_i \alpha_i = 0, \qquad (18)$$
$$0 \le \alpha_i \le C, i = 1, 2, ..., m,$$

Get the optimal α^*.
- **Step 2:** $2P_0$ can get

$$\omega = \sum_{i=1}^{m} \alpha_i^* d_i (x_i + r_1 + .. + r_q),$$

$$b = -\frac{1}{2}\left(\sum_{y_i=1} \alpha_i^* (x_i + r_1 + .. + r_q) - \sum_{y_i=-1} \alpha_i^* (x_i + r_1 + .. + r_q)\right).$$
$$\left(\sum_{y_i=1} \alpha_i^* (x_i + r_1 + .. + r_q) + \sum_{y_i=-1} \alpha_i^* (x_i + r_1 + .. + r_q)\right)$$

Get the classification function is f(x) = sgn $((\omega \cdot x)+b)$.
- **Step 3:** For each new $x \in R^n$ obtained by an entity, that entity wants to know its label. Execute the Algorithm 1. At last, P_0 gets the data $x+r_1+..+r_q$. Using the classification function in step 2, we can get the label of $x+r_1+..+r_q$ which is also the label of x.

Assuming that the participating entities do not collude with the initiator and do follow the given protocol correctly, the initiator successfully acquires the global SVM model without disclosing any information on the private data of any entity.

One may ask why not one entity adds its random row vector to each row of its private dataset and sends to the initiator entity P_0 directly. The reason is that random row vectors generated by entities may be different. Then the same column of the data horizontally partitioned has different bias values comparing to the original data. Thus it effects the classification accuracy. Another problem is that the initiator entity P_0 must send the empty dataset to its neighbor P_1 in the step 1 of the Algorithm 1. The key is that if P_0 sends data, in the end, it will know the bias value of its data. Because each row of the dataset has the same bias value ($r_1+..+r_q$), the initiator will know the data privately owned by the entities. It violates the privacy of the datasets.

From the algorithms described above, we know that each row of all datasets has the same bias value. The initiator P_0 constructs the PPSVM model using the perturbed dataset. Then we discuss how to perform testing/classification using the PPSVM model constructed by our algorithms. Suppose the entity P_i has the dataset New_i to know its classification. The entity P_i could simply send the original dataset New_i to initiator P_0 to classify. But this would violate this constraint that the initiator should learn nothing about the private data of any entity. Furthermore, the training datasets have the bias value. Using the original dataset to classify has a low accuracy. To be corresponding with the training dataset, the testing/classifying datasets should carry out the algorithms described above. Then the initiator would have the perturbed datasets for testing/classifying using the constructed PPSVM model.

As discussed earlier, the initiator is prohibited from contributing data to the testing set. That is to prevent it from obtaining any information. Thus P_1 must reject P_0's start request if P_0 sends a nonempty set in the Step 1 of Algorithm 1. No data is disclosed to any entity in this process. The details of security can be found in the paper by Agrawal and Srikant (2000).

Algorithm of PPSVM on vertically partitioned data: Suppose there are $P_1,P_2,...,P_q$ participating entities, having the datasets $A_1,A_2,...,A_q$ correspondingly. All of these entities hold some feature values of the same group. The group is constituted with m individuals. The entity P_i has f_i features, i = 1,2,..,q, where $f_1+f_2+...+f_q = n$. The whole data is $A = A[A_1\ A_2 ...A_q] \in R^{m\times n}$. Suppose there is a volunteer entity P_1 that is willing to compute the global PPSVM model. We call it initiator and other entities are called passive entities. Similar to the horizontally partitioned case, our goal is to obtain the perturbed dataset that has the same bias value for each column. Entity P_i(i = 2,3,...,q) generates a random vector r_i whose dimension is the same with the row dimension of it owned dataset. Each element of the vector r_i is drawn from Uniform distribution, Gaussian distribution or other distributions. Entity P_i holds the vector r_i privately. We now describe the algorithm (Algorithm 3) for PPSVM model on vertically partitioned data.

ALGORITHM 3

- **Step 0 (Initialize):** Choose the initiator P_1. Entity P_i (i = 2,3,...,q) generates its random vector r_i. The random vector does not change during the training and testing. All q entities agree on the same labels (matrix D) for each data point. If an agreement on D is not possible, we can use semi-supervised learning (Fung and Mangasarian, 2001b) to handle such data points. It is the future work.
- **Step 1:** For i = 2,3,...,q, entity P_i (i = 2,3,...,q) executes the operation $\oplus$: $A_i \oplus r_i$. Then the entity P_i sends $A_i \oplus r_i$ to the initiator P_1. Thus the initiator P_1 now owns the data $A_i \oplus r_i$, i = 2,3,..q. The initiator P_1 owns the data $\tilde{A} = [A_1\ A_2\ ...\ A_q] \oplus (0\ r_2\ ..\ r_q)$, where 0 is the vector whose elements are zero. Its data point is represented with $\bar{x}_i, i = 1,2,...,m$.
- **Step 2:** Using all data, initiator P_1 solve the PPSVM dual problem:

$$\begin{aligned} \min\ & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} d_i d_j \alpha_i \alpha_j K(\bar{x}_i, \bar{x}_j) - \sum_{i=1}^{m}\alpha_i \\ \text{s.t.}\ & \sum_{i=1}^{m} d_i \alpha_i = 0 \\ & 0 \le \alpha_i \le C \end{aligned} \tag{19}$$

Get the optimal α^*.

- **Step 3:** P_1 can get

$$\omega = \sum_{i=1}^{m} \alpha_i^* d_i \bar{x}_i.$$

$$b = -\frac{1}{2}\left(\sum_{y_i=1} \alpha_i^* \bar{x}_i - \sum_{y_i=-1} \alpha_i^* \bar{x}_i\right)\cdot\left(\sum_{y_i=1} \alpha_i^* \bar{x}_i + \sum_{y_i=-1} \alpha_i^* \bar{x}_i\right)$$

The classification function is f(x) = sgn ((ω·x)+b).

- **Step 4:** For each new x obtained by an entity P_i, i = 1,2,...,q, that entity want to know its label. Entity P_i sends the data $x+r_i$ to initiator P_1, if i = 1, then $r_i = 0$.

Using the classification generated in step 3, we can get the label of $x+r_i$ that is also the label of x.

In the Algorithm 3, the initiator P_1 does not send its privately owned data to any entity, so it does not disclose any information of its owned data. For the entity P_i (i = 2,3,..,q), it sends dataset $A_i \oplus r_i$ to initiator P_1. Because each element of the random vector r_i is generated from certain distribution, which is known to P_i, it is hard for the initiator P_1 to get any information of A_i. When entity P_i has data New_i to know its class label, P_i sends $New_i \oplus r_i$ to P_i as described in Algorithm 3. There is not any information disclosed in the protocol. Our algorithm is secure. The details of security can be found in the stduy by Agrawal and Srikant (2000).

EXPERIMENTAL RESULTS

The present research demonstrate the efficiency of our protocol in three ways. First, our experiments show that the accuracy obtained using this PPSVM is the same as that of SVM when the data is centralized. Second, our experiments show that using our approach, entities can obtain classifiers with lower misclassification error than that obtained using only one entity's data alone.

We choose three datasets from UCI repository. To simulate a situation in which each entity has only a subset of the feature space for each data point, we randomly distribute the features among the entities such that each entity receives about the same number features. Similarly, to simulate the situation in which each entity has only a subset of the individuals with the same features, we randomly distribute the data points among the entities such that each entity has about the same number individuals. The SVM model is created using the LIBSVM. During the experiments, we choose the same SVM model for one dataset.

Comparison of this approach with classifiers obtained when the data is original centralized: We demonstrate the classification accuracy of our approach is the same with that of the case when the data is centralized. The accuracy is shown in the Table 2 no matter horizontally partitioned or vertically partitioned.

From the Table 2, we can observe that the accuracy of our approach obtained the same with that obtained when the data is original centralized. Besides, the accuracy does not change with the increasing of entities number.

Table 2: Accuracy comparison of our approach no matter horizontally or vertically partitioned data case with the original centralized data case

Dataset examples×input feature	Entities	PPSVM accuracy (%)	Centralized accuracy (%)
SPECT heart	4	100.00	86.1423
267×22	10	100.00	88.3895
Heart disease	4	100.00	100.00
270×13	2	100.00	100.00
Lung cancer	4	92.5926	92.5926
27×56	3	92.5926	92.5926

Table 3: Comparison the classification accuracy of our approach with that of using only one entity's original data in the horizontally partitioned case

Dataset examples×input feature	Entities	PPSVM accuracy (%)	Centralized accuracy (%)
SPECT heart	4	100.00	80.5243
267×22	10	100.00	70.7865
Heart disease	4	100.00	67.03
270×13	2	100.00	79.6296
Lung cancer	4	92.5926	55.556
27×56	3	92.5926	66.667

Table 4: Comparison the classification accuracy of our approach with that of using only one entity's original data in the vertically partitioned case

Dataset examples×input feature	Entities	PPSVM accuracy (%)	Centralized accuracy (%)
SPECT heart	4	100.00	79.4007
267×22	10	100.00	79.4007
Heart disease	4	100.00	86.667
270×13	2	100.00	100.00
Lung cancer	4	92.5926	100.00
27×56	3	92.5926	92.5926

Comparison of our approach with classifiers obtained when the data is part of original centralized: We compare the classification accuracy of this approach with that of using only one entity's original data. The accuracy is shown in the Table 3 in the case of horizontally partitioned. Table 4 shown the accuracy of vertically partitioned case.

From Table 3 and 4, we can see that this approach has lower classification error comparing with that using only one entity's original data in the most cases, especially when the number of the data points is not large.

DISCUSSION

Though our approach is efficient and secure, there are some challenges. The requirement of an entrusted intermediator seems overly restrictive. A key challenge for the future work is to remove the need of the intermediator, while still efficiently constructing the SVM model. In the vertically partitioned case, we can deal with data points using semi-supervised learning methods when an agreement on class label D is not possible. We also can immerge our approach with other SVM model such as RSVM (Lee and Huang, 2007), PSVM (Fung and Mangasarian, 2001a) etc. to improve the accuracy and reduce running time. There is room for future work in PPSVM. Privacy-preserving Support Vector Regression may be a prospective study direction.

CONCLUSION

This study proposes a unified model for privacy-preserving classification on horizontally and vertically partitioned data. We use the distinct character of classification. Giving the same bias value to the same column of the training and testing datasets, it does not affect the accuracy of classification. During the data passing, no data information is disclosed. Our protocol is secure. Besides, our experiment shows that our approach is scalable as the increasing of entities number. It costs lower time. PPSVM proposed can deal with the real number data points comparing with the paper by Yu *et al.* (2006a). The initiator in our approach can deal with the received dataset as its original dataset. It is convenient for initiator to construct the global SVM model and test the new data.

ACKNOWLEDGMENT

This research is supported by the National Science Foundation of China under Grant (No. 10571109, 60603090 and 90718011).

REFERENCES

Agrawal, R. and R. Srikant, 2000. Privacy-preserving data mining. Proceedings of the 2000 ACM SIGMOD Conference on Management of Data. May 14-19, ACM, Dallas, TX, pp: 439-450.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov., 2: 121-167.

Chen, K. and L. Liu, 2005. Privacy preserving data classification with rotation perturbation. Proceedings of the 5th International Conference of Data Mining (ICDM'05). 2005 IEEE Computer Society Washington, DC, USA., pp: 589-592.

Fung, G. and O.L. Mangasarian, 2001a. Proximal support vector machine classifiers. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. August 2001, San Francisco, USA., pp: 77-86.

Fung, G. and O.L. Mangasarian, 2001b. Semi-supervised support vector machines for unlabeled data classification. Optimization Methods Software, 15: 29-44.

Ioannidis, I., A. Grama and M. Atallah, 2002. A secure protocol for computing dot-products in clustered and distributed environments. The 2002 International Conference on Parallel Processing. Aug. 18-21, Vancouver, Canada, pp: 379-384.

Kantarcioglu, M. and C. Clifton, 2004. Privacy-preserving distributed mining of association rules on horizontally partitioned data. IEEE Trans. Knowledge Data Eng., 16: 1026-1037.

Lee, Y.J. and S.Y. Huang, 2007. Reduced support vector machines: A statistical theory. IEEE Trans. Neural Networks, 18: 1-13.

Vaidya, J. and C. Clifton, 2002. Privacy-preserving association rule mining in vertically partitioned data. The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. July 2002, Edmonton Alberta, Canada, pp: 639-644.

Vapnik, V.N., 2000. The Nature of Statistical Learning Theory. 2nd Edn., Springer, Publishing, New York, USA, ISBN: 978-0-387-98780-4 pp: 175.

Yu, H., J. Vaidya and X. Jiang, 2006a. Privacy-preserving SVM classification on vertically partitioned data. Proceedings of PAKDD '06, LNCS 3918, January 2006, Springer-Verlag, pp: 647-656.

Yu, H., X. Jiang and J. Vaidya, 2006b. Privacy-preserving SVM using nonlinear on horizontally partitioned data. Proceedings of the 2006 ACM Symposium on Applied Computing, April 23-27, ACM New York, USA., pp: 603-610.