International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.5 (2016), pp.199-208
http://dx.doi.org/10.14257/ijmue.2016.11.5.18

Privacy-Preserving One-Class Support Vector Machine with Vertically Partitioned Data

Qiang Lin, Huimin Pei, Kuaini Wang and Ping Zhong *

(College of science, China Agricultural University, Beijing, 100083, China)
**Corresponding author. E-mail: zping@cau.edu.cn*

Abstract

We establish a new model of privacy-preserving one-class support vector machine (SVM) based on vertically partitioned data. Every participant holds all the data with a part of attributes. They apply different random matrices to establish their own kernel matrix. By sharing these partial kernel matrices, we construct a global kernel matrix and establish linear and nonlinear privacy-preserving models. Experimental results on benchmark data sets verify the validity of the proposed models.

Keywords*: One-class SVM, random kernel, privacy-preserving, vertically partitioned data*

1. Introduction

Data mining, as a powerful technique, has received significant attention. It can extract the valuable information from a large amount of data. Usually, the data contains a great deal of sensitive information, and it should not be revealed. Therefore, the privacy preservation technique is very important in the data mining.

Support Vector Machine (SVM) [1], based on the statistical learning theory, earns success in many aspects ranging from machine learning, data mining, and knowledge discovery and so on. It can effectively solve the traditional difficult problems, such as over-fitting and dimensional disaster problems. With the rapid developments, many branches of the SVM algorithms have been studied, which include the one-class SVM [2-3]. Schölkopf *et al.* [2] presented a method to adapt the SVM algorithm for one class classification problem. It only uses one class for training, instead of multiple classes. The core idea of one-class SVM is to find the maximum margin hyperplane between the training points and the origin. Recently, there have been more and more researches on one-class SVM [4-7]. One-class SVM has been applied in various fields, such as ecological modeling [8], text clustering [9] and so on. On the other hand, the studies of privacy-preserving SVM (PPSVM) [10-12] have attracted a growing attention. Hwanjo yu et.al applied secure multi-party computation to develop privacy-preserving model [10]. Based on reduced SVM (RSVM) [13-14] and random matrix, Mangasarian *et al.* have established PPSVM models [11-12].

In this paper, we propose a privacy-preserving one-class SVM based on the vertically partitioned data. We introduce the random matrix [15] to make the privacy-preserving model. Every participant holds all the data with a part of attributes. They apply different random matrices to establish their own kernel matrix. By sharing these partial kernel matrices, we construct a global kernel matrix and establish linear and nonlinear privacy-preserving models. Experimental results on benchmark datasets confirm the validity of the proposed algorithms.

The paper is organized as follows. Section 2 introduces one-class SVM. In Section 3, we establish the linear and nonlinear privacy-preserving one-class SVM models. Section 4 presents the experimental results on benchmark datasets. Finally, section 5 gives the conclusions.

ISSN: 1975-0080 IJMUE

2. One-Class SVM

Consider the training set $\{x_1, x_2, ..., x_l\} \subset X$, where X is the input space. One can formulate one-class SVM as follows.

$$\min_{w,\xi_i,\rho} \ \frac{1}{2}\|w\|^2 + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i - \rho$$
$$\text{s.t.} \quad w \cdot \phi(x_i) \geq \rho - \xi_i \tag{1}$$
$$\xi_i \geq 0, \quad i = 1, 2, ..., l$$

Where $\phi(\cdot)$ is a nonlinear map from the input space to the feature space, $\nu \in (0,1]$ is a parameter which can control the fraction of outliers and the fraction of support vectors. In practice, the above optimization problem can be solved by its dual:

$$\min_{\alpha} \ \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j K(x_i, x_j)$$
$$\text{s.t.} \quad 0 \leq \alpha_i \leq \frac{1}{\nu l} \tag{2}$$
$$\sum_{i=1}^{l}\alpha_i = 1$$

Where $K(\cdot,\cdot)$ is a kernel function. The optimal normal vector is given by $w = \sum_{i=1}^{l}\alpha_i\phi(x_i)$, and the resultant one-class SVM model for prediction is

$$f(x) = \text{sgn}(\sum_{i=1}^{l}\alpha_i K(x_i, x) - \rho) \tag{3}$$

Where α_i is the solution of the dual problem and training samples x_i with non-zero α_i are support vectors. Select α_j^* from the components of α in the interval $(0, 1/\nu l)$, then $\rho = \sum_{j=1}^{l}\alpha_j^* K(x_j, x_i)$.

3. Privacy-Preserving One-Class SVM

In this section, we present linear privacy-preserving one-class SVM (VLPPOCSVM) and nonlinear privacy-preserving one-class SVM (VNPPOCSVM) based on vertically partitioned data. Denote A a real $l \times n$ matrix, which is shown in Figure 1. There are N data sets $A_1, A_2, ..., A_N$, where A_i is an $l \times n_i$ matrix, which is the i-th column or i-th block of columns of A. A_i is the data that is held by the i-th participant, and it contains only a part of attributes. $n_1 + ... + n_N = n$. Then $A = (A_1, A_2, ..., A_N)$. Denote $B = (B_1^{\mathrm{T}}, B_2^{\mathrm{T}}, ..., B_N^{\mathrm{T}})^{\mathrm{T}}$ an $n \times k$ random real matrix, where B_i is an $n_i \times k$ random matrix with rank k. According to [16], we know that such B_i exists. When $x, y \in R^n$, $K(x, y)$ is a real number, $K(x, B)$ is a row vector in R^k, and $K(A, B)$ is an $l \times k$ matrix.

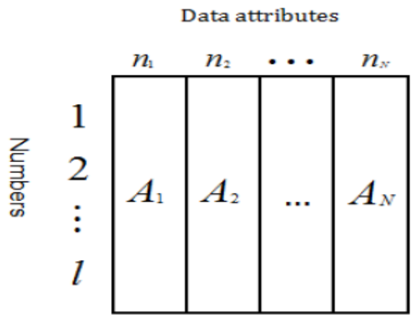

Figure 1. Vertically Partitioned Data

3.1 Linear Privacy-Preserving One-Class SVM

First we propose a linear privacy-preserving one-class SVM. Suppose $w = Bu$. We formulate the privacy-preserving model for one-class SVM as follows.

$$\begin{aligned} &\min_{u,\xi,\rho} \ \frac{1}{2}u^{\mathrm{T}}\left(B_1^{\mathrm{T}}B_1 + B_2^{\mathrm{T}}B_2 ... + B_N^{\mathrm{T}}B_N\right)u + \frac{1}{\nu l}e^{\mathrm{T}}\xi - \rho \\ &\text{s.t. } \left(A_1B_1 + A_2B_2 + ... + A_NB_N\right)u \geq \rho e - \xi \\ &\xi \geq 0 \end{aligned} \tag{4}$$

Where $e = (1,1,...,1)^{\mathrm{T}} \in R^l$. In order to solve the optimization problem (4), we introduce the Lagrangian function:

$$\begin{aligned} L(u,\xi,\rho,\alpha,\beta) = &\frac{1}{2}u^{\mathrm{T}}\left(B_1^{\mathrm{T}}B_1 + B_2^{\mathrm{T}}B_2 ... + B_N^{\mathrm{T}}B_N\right)u + \frac{1}{\nu l}e^{\mathrm{T}}\xi - \rho \\ &- \alpha^{\mathrm{T}}\left[\left(A_1B_1 + A_2B_2 + ... + A_NB_N\right)u - \rho e + \xi\right] - \beta^{\mathrm{T}}\xi \end{aligned} \tag{5}$$

Where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_l)^{\mathrm{T}} \geq 0,\ \beta = (\beta_1, \beta_2, ..., \beta_l)^{\mathrm{T}} \geq 0$ are Lagrange multipliers. By Karush-Kuhn-Tucker (KKT) conditions, we derive

$$\frac{\partial L}{\partial u} = \left(B_1^{\mathrm{T}}B_1 + B_2^{\mathrm{T}}B_2 ... + B_N^{\mathrm{T}}B_N\right)u - \left(A_1B_1 + A_2B_2 + ... + A_NB_N\right)^{\mathrm{T}}\alpha = 0 \tag{6}$$

$$\frac{\partial L}{\partial \xi} = \frac{1}{\nu l}e - \alpha - \beta = 0 \tag{7}$$

$$\frac{\partial L}{\partial \rho} = -1 + e^{\mathrm{T}}\alpha = 0 \tag{8}$$

$$\alpha_j((A_1B_1 + A_2B_2 + ... + A_NB_N)_j u - \rho + \xi_j) = 0,\ \ j = 1,2,...,l \tag{9}$$

$$\beta_j\xi_j = 0,\ \ j = 1,2,...,l \tag{10}$$

Where $(A_1B_1 + A_2B_2 + ... + A_NB_N)_j$ denotes the j -th row of $\left(A_1B_1 + A_2B_2 + ... + A_NB_N\right)$. The dual of (4) is

$$\begin{aligned} &\min_{\alpha} \ \frac{1}{2}\alpha^{\mathrm{T}}\left(A_1B_1 + A_2B_2 + ... + A_NB_N\right)\left(B_1^{\mathrm{T}}B_1 + B_2^{\mathrm{T}}B_2 ... + B_N^{\mathrm{T}}B_N\right)^{-1} \\ &\qquad (A_1B_1 + A_2B_2 + ... + A_NB_N)^{\mathrm{T}}\alpha \\ &\text{s.t.} \quad 0 \leq \alpha \leq \frac{1}{\nu l}e \\ &\qquad e^{\mathrm{T}}\alpha = 1 \end{aligned} \tag{11}$$

The optimal normal vector is given by

$$u^* = \left(B_1^{\mathrm{T}}B_1 + B_2^{\mathrm{T}}B_2 ... + B_N^{\mathrm{T}}B_N\right)^{-1}(A_1B_1 + A_2B_2 + ... + A_NB_N)^{\mathrm{T}}\alpha^* \tag{12}$$

where α^* is the optimal solution of (11). Select $\alpha_j^* \in (0,1/\nu l)$ from the components of α^*. According to (7), we get the corresponding $\beta_j > 0$. Thus by (9) and (10), we have $(A_1B_1 + A_2B_2 + ... + A_NB_N)_j u - \rho = 0$. Hence, we get

$$\rho^* = (A_1B_1 + A_2B_2 + ... + A_NB_N)_j u^* \tag{13}$$

The decision function is

$$f(x) = \text{sgn}(x^T B u^* - \rho^*) = \text{sgn}\left(\left(x_1^T B_1 + x_2^T B_2 + ... + x_N^T B_N\right)u^* - \rho^*\right) \tag{14}$$

Where $x^T = \left(x_1^T, x_2^T, ..., x_N^T\right), x_j \in R^{n_j}, j = 1, 2, ..., N$.

Linear algorithm

(1) All N participants generate their own random matrix $B_i \in R^{n_i \times k}$ with rank k, $i = 1, 2, ..., N$.

(2) Each participant i makes its linear kernel A_iB_i and $B_i^T B_i$ public, and does not reveal A_i and B_i.

(3) Given parameter ν, solve the quadratic programming (11) and get the optimal solution α^*.

(4) Calculate (12) and (13) to get u^* and ρ^*.

(5) Each participant provides $x_i^T B_i$ for a new sample, compute (14) to assign a new sample.

3.2 Privacy-Preserving Nonlinear One-Class SVM

In order to extend the linear model to the nonlinear case, we express w in terms of the mapped $B_i = (z_{i1}, z_{i2,...,} z_{ik}), i = 1, ..., N$ as follows.

$$w = \sum_{i=1}^{N}\sum_{j=1}^{k} u_j \phi(z_{ij}) \tag{15}$$

The privacy-preserving nonlinear one-class SVM is formulated as follows.

$$\begin{aligned} \min_{u,\xi,\rho} \quad & \frac{1}{2} u^T K(B^T, B) u + \frac{1}{\nu l} e^T \xi - \rho \\ \text{s.t.} \quad & K(A, B) u \geq \rho e - \xi \\ & \xi \geq 0 \end{aligned} \tag{16}$$

Where

$$K(B^T, B) = K(B_1^T, B_1) + K(B_2^T, B_2) + ... + K(B_N^T, B_N) \tag{17}$$

With

$$K(B_i^T, B_i) = \begin{bmatrix} (\phi(z_{i1}) \cdot \phi(z_{i1})) & ... & ... & (\phi(z_{i1}) \cdot \phi(z_{ik})) \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ (\phi(z_{ik}) \cdot \phi(z_{i1})) & \cdots & \cdots & (\phi(z_{ik}) \cdot \phi(z_{ik})) \end{bmatrix} \tag{18}$$

$$K(A, B) = K(A_1, B_1) + K(A_2, B_2) + ... + K(A_N, B_N) \tag{19}$$

With

$$K(A_i,B_i)=\begin{bmatrix}(\phi(A_{i1})\cdot\phi(z_{i1})) & \cdots & \cdots & (\phi(A_{i1})\cdot\phi(z_{ik}))\\ \vdots & & & \vdots\\ (\phi(A_{ir})\cdot\phi(z_{i1})) & \cdots & \cdots & (\phi(A_{ir})\cdot\phi(z_{ik}))\\ \vdots & & & \vdots\\ (\phi(A_{il})\cdot\phi(z_{i1})) & \cdots & \cdots & (\phi(A_{il})\cdot\phi(z_{ik}))\end{bmatrix} \quad (20)$$

Where $A_{ir}, r=1,...,l$ is the r-th row of A_i.The Lagrangian function of (16) is

$$L(u,\xi,\rho,\alpha,\beta)=\frac{1}{2}u^{\mathrm{T}}K(B^{\mathrm{T}},B)u+\frac{1}{\nu l}e^{\mathrm{T}}\xi-\rho-\alpha^{\mathrm{T}}(K(A,B)u-\rho e+\xi)-\beta^{\mathrm{T}}\xi \quad (21)$$

Where $\alpha=(\alpha_1,\alpha_2,...,\alpha_l)^{\mathrm{T}}$ and $\beta=(\beta_1,\beta_2,...,\beta_l)^{\mathrm{T}}$ are Lagrange multipliers. Differentiating L with respect to u,ξ,ρ and setting the results to zero, we obtain

$$\frac{\partial L}{\partial u}=K(B^{\mathrm{T}},B)u-\left(K(A,B)\right)^{\mathrm{T}}\alpha=0 \quad (22)$$

$$\frac{\partial L}{\partial \xi}=\frac{1}{\nu l}e-\alpha-\beta=0 \quad (23)$$

$$\frac{\partial L}{\partial \rho}=-1+e^{\mathrm{T}}\alpha=0 \quad (24)$$

$$\alpha_j[(K(A,B))_j u-\rho+\xi_j]=0,\ \ j=1,2,...,l \quad (25)$$

$$\beta_j\xi_j=0,\ \ j=1,2,...,l \quad (26)$$

Where $(K(A,B))_j$ denotes the j-th row of $K(A,B)$. The quadratic program (16) can be solved by its dual:

$$\begin{aligned}\min_{\alpha}\ \ & \frac{1}{2}\alpha^{\mathrm{T}}K(A,B)K(B^{\mathrm{T}},B)^{-1}\left(K(A,B)\right)^{\mathrm{T}}\alpha\\ \text{s.t.}\ \ & 0\le\alpha\le\frac{1}{\nu l}e\\ & e^{\mathrm{T}}\alpha=1\end{aligned} \quad (27)$$

Then we get

$$u^*=K(B^{\mathrm{T}},B)^{-1}\left(K(A,B)\right)^{\mathrm{T}}\alpha^* \quad (28)$$

Where α^* is the optimal solution of (27). Select $\alpha_j\in(0,1/\nu l)$ from the components of α^*. By (23), (25), and (26), we have the corresponding $\beta_j>0$ and $(K(A,B))_j u-\rho=0$. Hence, we get

$$\rho^*=(K(A,B))_j u^* \quad (29)$$

The decision function is

$$\begin{aligned}f(x)&=\mathrm{sgn}(K(x^{\mathrm{T}},B)u^*-\rho^*)\\ &=\mathrm{sgn}\left(\left(K(x_1^{\mathrm{T}},B_1)+K(x_2^{\mathrm{T}},B_2)+...+K(x_N^{\mathrm{T}},B_N)\right)u^*-\rho^*\right)\end{aligned} \quad (30)$$

Where $K(x_i^{\mathrm{T}},B_i)=\left[\left(\phi(x_i)\cdot\phi(z_{i1})\right),...,\left(\phi(x_i)\cdot\phi(z_{ik})\right)\right]$.

Nonlinear algorithm

(1) All N participants generate their own random matrix $B_i\in R^{n_i\times k}$ with rank k.

(2) Each participant i makes public its nonlinear kernel $K(A_i,B_i)$ and $K(B_i^{\mathrm{T}},B_i)$, and does not reveal A_i and B_i.

(3) Calculate $K(B^{\mathrm{T}},B)$ according to (17), and compute $K(A,B)$ by (19).

(4) Given the parameter ν, solve the optimization problem (27) and get the optimal

solution α^*.

(5) Calculate (28) and (29) to get u^* and ρ^*, respectively.

(6) Each participant provides $K(x_i^{\mathrm{T}}, B_i)$ for a new sample $x_i \in R^n$, compute (30) to assign a new sample.

4. Experiments

In order to test the validity of the proposed algorithms, we compared vertically linear privacy-preserving one-class SVM (VLPPOCSVM) and vertically nonlinear privacy-preserving one-class SVM (VNPPOCSVM) with linear and nonlinear OCSVMs by experiments on a collection of six benchmark data sets from UCI Machine Learning Repository, which are Heart, Ionosphere, Bupa, WDBC, Pima and German. Table 1 gives the description of these data sets. For each data set, the positive class was chosen as the target class and the negative class as outliers.

Table 1. Description of Benchmark Data Sets

Data set	DataNum	Attribute	Positive instances	Negative instances
Heart	270	13	150	120
Ionosphere	351	34	225	126
Bupa	345	6	200	145
WDBC	569	30	357	212
Pima	768	8	500	268
German	1000	24	700	300

First, we divided the data into several parts according to the attribute. For example, in the first column of Table 2, the number “5” indicates that we divided the Heart data into 5 parts $A_1,...,A_5$, where $A_1,...,A_4 \in R^{270\times2}$, $A_5 \in R^{270\times5}$. This simulates that there are 5 participants and they own the part data sets $A_1,...,A_5$, respectively. For each data set, two kinds of grouping were given, shown in Table 2.

Table 2. Data Grouping

	Heart	Ionosphere	Bupa	WDBC	Pima	German
Data grouping	5	5	5	5	5	5
	4	11	2	10	2	8

We selected positive samples of each data set as the training set and used the ten-fold cross validation method for parameter optimization. All the positive samples were randomly divided into 10 disjoint subsets $s_1, s_2, ..., s_{10}$, with each subset of roughly equal size, and then 10 iterations were operated. $s_1, s_2, ..., s_{i-1}, s_{i+1}, ..., s_{10}$ were the training sets for the i-th iteration process; s_i and all negative samples were used as the test samples. At i-th iteration, we got the misclassification point number m_i and calculated the iterative error ratio $R_i = {}^{m_i}\!/_{m}$, where m is the number of the test samples. At last, we got $R_1, R_2, ..., R_{10}$ and their average ratio after 10 iterations:

$$r = \sum_{i=1}^{10} \frac{R_i}{10}, \tag{31}$$

Where r is an evaluation index of model. In addition, we adopted G-means as another evaluation index.

$$\text{G-means} = \sqrt{acc^+ \times acc^-} \tag{32}$$

$$acc^{+} = \frac{TP}{TP + FN}$$
$$acc^{-} = \frac{TN}{TN + FP} \quad (33)$$

Where TP is the number of positive samples which are predicted as positive samples; FN is the number of positive samples which are predicted as negative samples; TN is the number of negative samples which are predicted as negative samples; FP is the number of negative samples which are predicted as positive samples. The higher G-means is, the higher acc^{+} and acc^{-}. The parameter ν was chosen from $\{0.1, 0.2, ..., 1\}$ in the models. In nonlinear model, the Gauss kernel function $K(x, x') = \exp(-\|x - x'\| / 2\sigma^2)$ was employed. The optimal parameters (ν^*, σ^*) were chosen from $\{0.1, 0.2, ..., 1\} \times \{2^{-6}, 2^{-5}, ..., 2^{6}\}$.

Table 3 reports the experimental results of two linear models VLPPOCSVM and linear OCSVM. Note that for each data set, our method VLPPOCSVM completed two sets of experiments. For example, the lines of VLPPOCSVM (5) and VLPPOCSVM (4) for Heart data set show the results of VLPPOCSVM when there are 5 and 4 participants, respectively. From Table 3, we can see that the error ratios of two linear models are similar except that the error ratios of linear OCSVM are lower on two data sets. G-means values of VLPPOCSVM and linear OCSVM are approximate. We note that unlike OCSVM, in VLPPOCSVM model, the data multiplied by random matrix instead of real data was used to do experiments. The VLPPOCSVM model has the effect of privacy preservation.

Table 4 shows the results of the error ratios and G-means values of two nonlinear models VNPPOCSVM and nonlinear OCSVM. Comparing the two nonlinear models, we conclude that the error ratios of VNPPOCSVM and nonlinear OCSVM are almost nearly. The G-means values of VNPPOCSVM are slightly lower than those of nonlinear OCSVM. However, we notice that unlike nonlinear OCSVM, the kernel matrix generated by the original data was not used in VNPPOCSVM model to solve the one class classification problem. Instead, the whole kernel matrix employed by VNPPOCSVM is protected by the random matrices, which makes an influence on the classification accuracy. But the gap between the two models is very small. The VNPPOCSVM model has the effect of privacy preservation.

5. Conclusion

We have presented two privacy-preserving models VLPPOCSVM and VNPPOCSVM to solve one class classification problems. The different random matrices are employed to calculate the participants' kernel matrices. For linear case, each participant makes public only the data multiplied by the random matrix instead of the real data. For nonlinear case, the participants generate the kernel matrices by their private random matrices. A global kernel matrix can be generated by the combination of these partial kernel matrices. Partial kernel matrix can protect the privacy of the participants, and the global kernel matrix can ensure the classification accuracy. Experimental results indicate that VLPPOCSVM and VNPPOCSVM not only have good classification accuracy, but also realize the data privacy preservation.

Table 3. Experimental Results of VLPPOCSVM and Linear OCSVM

Data sets	Algorithm	ν^*	r	G-means
Heart	OCSVM	1.0	0.1113	0.7235
	VLPPOCSVM(5)	0.9	0.2237	0.7072
	VLPPOCSVM(4)	1.0	0.2326	0.7091
Ionosphere	OCSVM	0.9	0.1480	0.7348
	VLPPOCSVM(5)	1.0	0.1459	0.7150
	VLPPOCSVM(11)	1.0	0.1480	0.7345
Bupa	OCSVM	1.0	0.0761	0.7271
	VLPPOCSVM(5)	1.0	0.0872	0.7166
	VLPPOCSVM(2)	1.0	0.1480	0.7345
WDBC	OCSVM	1.0	0.5661	0.4596
	VLPPOCSVM(5)	1.0	0.5676	0.4331
	VLPPOCSVM(10)	1.0	0.5576	0.4423
Pima	OCSVM	1.0	0.1601	0.7360
	VLPPOCSVM(5)	1.0	0.1612	0.7290
	VLPPOCSVM(2)	1.0	0.1601	0.7361
German	OCSVM	1.0	0.2021	0.7325
	VLPPOCSVM(5)	1.0	0.1946	0.7345
	VLPPOCSVM(8)	1.0	0.1962	0.7419

Table 4. Experimental Results of VNPPOCSVM and Nonlinear OCSVM

Data sets	Algorithm	υ^*	σ^*	r	G-means
Heart	OCSVM	0.2	2^4	0.1111	0.7320
	VNPPOCSVM(5)	1.0	2^2	0.1267	0.7033
	VNPPOCSVM(4)	0.5	2^4	0.1104	0.7275
Ionosphere	OCSVM	0.5	2^3	0.1480	0.6970
	VNPPOCSVM(5)	1.0	2^3	0.1486	0.7349
	VNPPOCSVM(11)	1.0	2^3	0.1507	0.7296
Bupa	OCSVM	0.2	2^3	0.0650	0.7190
	VNPPOCSVM(5)	1.0	2^4	0.0701	0.6920
	VNPPOCSVM(2)	1.0	2^3	0.0652	0.7031
WDBC	OCSVM	0.5	2^3	0.1417	0.7336
	VNPPOCSVM(5)	1.0	2^2	0.2146	0.7235
	VNPPOCSVM(10)	1.0	2^4	0.1413	0.7335
Pima	OCSVM	0.1	2^{-2}	0.1572	0.7367
	VNPPOCSVM(5)	0.8	2^5	0.1569	0.7310
	VNPPOCSVM(2)	1.0	2^4	0.1555	0.7359
German	OCSVM	0.9	2^3	0.1892	0.7431
	VNPPOCSVM(5)	0.1	2^5	0.1902	0.7230
	VNPPOCSVM(8)	1.0	2^{-3}	0.1878	0.7235

Acknowledgments

The work is supported by National Natural Science Foundation of China Grant No.11171346.

References

[1] V. N. Vapnik, "The Nature of Statistical Learning", Berlin: Springer, **(1995)**.
[2] B. Schölkopf, R. C. Williamson, A. J. Smola, J. S. Taylor and J. C. Platt, "Support vector method for novelty detection", Neural Information Processing Systems, vol. 12, **(1999)**, pp. 526-532.
[3] D. Tax and R. Duin, "Support vector domain description", Pattern Recognition Letters, vol. 20, no. 11, **(1999),** pp. 191-199.
[4] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification", Journal of Machine Learning Research, vol. 2, **(2001)**, pp. 139-154.
[5] L. Zhuang and H. Dai, "Parameter optimization of kernel-based one-class classifier on imbalance learning", Journal of Computers, vol. 1, **(2006)**, pp. 32-40.
[6] W. Zhu and P. Zhong, "A new one-class SVM based on hidden information", Knowledge-Based Systems, vol. 60, **(2014)**, pp. 35-43.
[7] Z. Zhao, P. Zhong and Y. Zhao, "Reduced least squares one class SVM in empirical feature space for imbalanced Data", ICIC Express Letters, vol. 5, no. 11, **(2010)**, pp. 4115-4121.
[8] G. Foody, A. Mathur, H. C. Sanchez and D. S. Boyd, "Training set requirements for classification of a specific class", Remote sensing of Environment, vol. 104, no. 1, **(2006)**, pp. 1-14.
[9] L. Manevitz and M. Yousef, "One class SVMs for document classification", Journal of machine learning research, vol. 2, **(2002)**, pp. 139-154.
[10] H. Yu, X. Q. Jiang and J. Vaidya, "Privacy-preserving SVM using nonlinear kernels on vertically partitioned data", Advances in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, **(2006)**, pp. 647-656.
[11] O. L. Mangasarian and E. W. Wild, "Privacy-preserving classification of vertically partitioned data via random kernels", ACM Transactions on Knowledge Discovery from Date(TKDD), vol. 2, no. 3, **(2008)**, pp. 147-149.
[12] O. L. Mangasarian and E. W. Wild, "Privacy-Preserving classification of horizontally partitioned data via random kernels", Proceedings of the 2008 International Conference on Data Mining, **(2008)**, pp. 473- 479.
[13] Y. J. Lee and O. L. Mangasarian, "RSVM: reduced support vector machines", Proceedings of the First SIAM International Conference on Data Mining, **(2001)**, pp. 57-64.
[14] K. Lin and C. J. Lin, "A study on reduced support vector machines", IEEE Transactions on Neural Network, vol. 45, no. 2, **(2003)**, pp. 199- 204.
[15] W. Hu and W. Wang, "Privacy protection of SVM classification method", Electronic journals, vol. 2, no. 2, **(2012)**, pp. 281-286.
[16] X. Feng and Z. Zhang, "The rank of random matrix", Applied Mathematics and Computation, vol. 2, no. 1, **(2007)**, pp. 689-694.