Comparison of performance with support vector machines

Kai Li, Lulu Zhai

College of Computer Science and Technology, Hebei University, Baoding,China,071000

ABSTRACT

Support vector machine (SVM) is an important machine learning method, which has many applications in pattern recognition, network security, etc. However, this method has some shortcomings such as complicated computation of quadratic programming, time-consuming training and low anti-noise performance. To this end, the researchers have proposed some improved methods. In this paper, we select the commonly used SVM classifiers including C-SVM, v-SVM, PSVM and TWSVM to study their performance of classification on the standard data set by experiments. The experimental results are shown that the performance with TWSVM has an advantage over C-SVM, v-SVM and PSVM in selected ten data sets using linear kernel. However, when Gauss kernel is used, accuracy with different support vector machine is almost no differences except data set wdbc.

Keywords: support vector machine, proximal support vector machine, twin support vector machine, kernel function, classification

1. INTRODUCTION

Support Vector Machine (SVM) is a kind of machine learning method proposed by Vapnik et al. [1],which follows the principle of VC dimension and structure risk minimization of statistical learning theory. Its basic idea is to find a hyper-plane with the largest margin and separate the two classes of samples. Since the support vector machine is proposed, this method attracts the attention of many researchers and proposed some different support vector machine algorithms [2-3]. However, there exist some flaws for the traditional support vector machine, for example, the complicated computation of quadratic programming, time-consuming training, the low anti-noise performance, etc. In order to solve them, the researchers proposed the fuzzy support vector machine and the rough support vector machine by introducing the fuzzy theory or the rough set theory into the support vector machine [4-7] to solve the impact of noise or outlier point on the support vector machine. In addition, some researchers use the decomposition method to improve the training speed of support vector machines, such as SMO [8] and SVMlight [9]. As the support vector machine only considers the interclass separation in the sample data and ignores the intraclass structural information in the data, the result of the classification is not optimal. To this end, the researchers introduce structural information into support vector machines to propose the structural support vector machines [10-11]. It can be seen that for the support vector machine and its improved methods described above, the solved method is to convert the primal problem into a dual problem to obtain the required hyper-plane. For this method, when the size of data is very large, it is extremely difficult to solve the optimization problem. Therefore, some researchers use the approximate method to solve the problem. Unfortunately, the obtained solution for solving the dual problem is inferior to the solution for the primal problem [12]. In addition, some researchers have solved the complicated computational problems of quadratic programming by modifying the optimization problem model. On the basis of it, they obtain the support vector machine by solving linear equation systems [13].

It is seen that for the traditional support vector machine and its improved methods, they make the decision of the sample using only a hyper-plane. Recently, Fung et al. [14] constructed two parallel hyper-planes, which hold a farther distance, by solving the linear equation systems. Based on it, a proximal SVM (PSVM) is proposed. To make the decision of sample, it is necessary to calculate the distance of sample from two parallel hyper-planes, and then to use the distance to determine the category of sample. After that, Mangasarian et al. [15] proposed a generalized eigenvalue support vector machine (GEPSVM) based on idea of PSVM by relaxing the parallel constraint on two hyper-planes. In this method, the optimization problem is transformed into a generalized eigenvector problem of the eigen equation. However, each optimization problem uses all the given data, and the computational complexity is very large. Inspired by the GEPSVM approach, Jayadeva et al. [16] proposed the twin support vector machine (Twin SVM). Unlike the GEPSVM, to obtain two hyper-planes, TWSVM does not solve a larger optimization problem but solve two smaller optimization problems such as SVM, and each optimization problem uses only half of the sample data. Therefore, the training time of the method is shortened to 1/4 of the original SVM. Recently, researchers propose some improved methods based on the idea of TWSVM or SVM [17-20]. In view of the importance of support vector machine, in this paper, we select the commonly used support vector machine classifiers, which contain C-SVM, v-SVM, PSVM and TWSVM, and ten data sets from the UCI and Stalog database to study their performance in the experiment.

2. SUPPORT VECTOR MACHINE AND ITS IMPROVED METHOD

Given a sample set $X=\{(x_1,y_1),(x_2,y_2)\cdots,(x_l,y_l)\}$, where $x_i\in R^n$, class label $y_i\in\{-1,+1\}$, $i=1,2,\cdots,l$.

2.1 C-SVM

C-SVM[1] is a support vector classifier which is presented by Vapnik et al. It finds the maximal margin separating hyper-plane $w^T\varphi(x)+b=0$ between two classes of samples in the feature space H maximizing the margin $2/\|w\|^2$ and satisfying $y_i(w^T\varphi(x_i)+b)\ge 1, i=1,2,\cdots,l$, where $\varphi(.)$ maps X into H and $w\in H$. For the linear case, we have $\varphi(x)=x$. C-SVM can be expressed as the following optimization problem:

$$\begin{aligned}&\min_{w,b,\xi_i}\ \frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}\xi_i\\ &s.t.\ \ y_i\left(\left(w^T\cdot\varphi(x_i)\right)+b\right)\ge 1-\xi_i,\xi_i\ge 0,i=1,2,...,l\end{aligned}, \qquad (1)$$

where C>0 is the pre-specified parameter and ξ_i is slack variable.

By introducing the Lagrangian multipliers $\alpha_i(i=1,2,\cdots,l)$, we obtain the dual problem with (1) as follows:

$$\begin{aligned}&\max_{\alpha}\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i y_i\alpha_j k(x_i,x_j)y_j\\ &s.t.\ \ 0\le\alpha_i\le C,i=1,2,\cdots,l\\ &\sum_{i=1}^{l}\alpha_i y_i=0\end{aligned}. \qquad (2)$$

It can be seen that (2) is a convex quadratic programming problem. After optimizing this dual quadratic programming problem, we obtain the solution of Lagrangian multipliers $\alpha_i(i=1,2,\cdots,l)$, namely $\alpha_i^*(i=1,2,\cdots,l)$. At this time, the decision function is expressed as

$$\begin{aligned}f(x)&=sgn\left(\sum_{i=1}^{n}\alpha_i^*y_i\varphi(x_i)\cdot\varphi(x)+b^*\right)\\ &=sgn\left(\sum_{i=1}^{l}\alpha_i^*y_i k(x_i,x)+b^*\right)\end{aligned}, \qquad (3)$$

where $w=\sum_{i=1}^{l}y_i\alpha_i^*\varphi(x_i)$, b^* is derived by KKT condition.

2.2 *v*-SVM

In C-SVM, how to select an appropriate parameter C is a very difficult. So, Scholkopf et al.[2] presented the *v*-SVM, in which the selection of the parameter $v\in[0,1]$ is more intuitive. In the *v*-SVM, the primal optimization problem is in the following:

$$\begin{aligned}&\min_{w,b,\xi,\rho}\ \frac{1}{2}\|w\|^2-v\rho+\frac{1}{l}\sum_{i=1}^{l}\xi_i\\ &s.t.\ \ y_i(w^T\varphi(x_i)>+b\ge\rho-\xi_i\\ &\qquad \xi_i\ge 0,\ \rho\ge 0,i=1,2,...,l\end{aligned}, \qquad (4)$$

where w,b,ξ,ρ are variables to be optimized.

By introducing Lagrange multipliers method, we derive the dual problem of (4) in the following:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j k(x_i,x_j) \\ s.t. \quad & \sum_{i=1}^{l}\alpha_i y_i = 0,\ 0 \le \alpha_i \le \frac{1}{l}, \\ & \sum_{i=1}^{l}\alpha_i \ge v, i=1,2,...,l. \end{aligned} \tag{5}$$

where $\alpha_i (i=1,2,\cdots,l)$ is the Lagrange multiplier associated with the constraint, and $k(x_i,x_j)$ represents the kernel function which gives the dot product $\varphi(x_i)\cdot\varphi(x_j)$ in the feature space.

Let $\alpha^* = (\alpha_1^*, \alpha_2^*, ..., \alpha_l^*)$ represent the optimal solution of the dual optimization (5) and $w = \sum_{i=1}^{l}\alpha_i^* y_i \varphi(x_i)$. Then, the optimal separating decision function is given by

$$f(x) = \mathrm{sgn}(\sum_{i=1}^{l}\alpha_i^* y_i \varphi(x)\cdot\varphi(x_i) + b) = \mathrm{sgn}(\sum_{i=1}^{l}\alpha_i^* y_i k(x,x_i) + b), \tag{6}$$

where $b^* = -\frac{1}{2}\sum_{i=1}^{l}\alpha_i^* y_i (k(x_i,x_j) + k(x_i,x_k))$,here $j \in \{i \mid \alpha_i^* \in (0,1/l), y_i = +1\}$,and $k \in \{i \mid \alpha_i^* \in (0,1/l), y_i = -1\}$.

2.3 PSVM

In proximal support vector machine (PSVM)[14], two parallel planes are generated such that each plane is closest to one of two data sets to be classified and the two planes are as far apart as possible. This PSVM mainly modifies the objective function in optimization problem (1) and replaces the inequality constraint by the equality. For generating a linear or nonlinear classifier, it merely requires the solution of a single system of linear equations. The optimization problem for PSVM is repressed as follows:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}(w^T w + b^2) + C\frac{1}{2}\|\xi\|^2 \\ s.t \quad & y_i(w^T\varphi(x_i) - b) + \xi_i = 1, i = 1,2,\cdots,l \end{aligned} \tag{7}$$

Now, let the $l \times n$ matrix A represent sample set X, where each row of A denotes a sample point. In addition, class label of each sample is depicted by the $l \times l$ diagonal matrix D with plus ones or minus ones along its diagonal. The optimization problem (7) is modifying as (8) in the following.

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}(w^T w + b^2) + C\frac{1}{2}\|\xi\|^2 \\ s.t. \quad & D(K(A,A^T)Dw - eb) + \xi = e \end{aligned} \tag{8}$$

where $K(A,B) = (K_{ij})_{l\times l}$ is a $l \times l$ matrix whose element is expressed as $K_{ij} = \exp(-\sigma \| A_i^T - B_{.j} \|^2)$. A_i is the ith row of A while $B_{.j}$ is the jth column of A. The decision function is represented as

$$f(x) = (K(x^T, A^T)K(A,A^T)^T + e^T)Dv,$$

where $v = (\frac{I}{C} + GG^T)^{-1}e$ and $G = D[K \quad -e]$.

2.4 TWSVM

For performance of improvement on SVM, the researchers proposed the twin support vector machine (TWSVM) [16] by deeply study. TWSVM seeks two nonparallel proximal hyper-planes such that each hyper-plane is closest to one of two classes and as far as possible from the other class and solves two smaller sized quadratic programming problems other than a quadratic programming problem like SVM.

Suppose that all of the sample points in class +1 and class -1 are denoted by matrixes $A_+ \in R^{m_1\times n}$ and $A_- \in R^{m_2\times n}$,respectively. Different from traditional SVM, TWSVM seeks a pair of nonparallel hyper-planes $K(x^T,C^T)w^{(1)} + b^{(1)} = 0$ and $K(x^T,C^T)w^{(2)} + b^{(2)} = 0$, Such that each hyper-plane is proximal to the sample points of one class and far from the sample points of the other class, where $C^T = [A_+ \quad A_-]$.

For TWSVM classifier, their pairs of quadratic programming problems are written as

$$(TSVM1)\quad \min_{w^{(1)},b^{(1)},q} \quad \frac{1}{2}(K(A_+,C^T)w^{(1)}+e_1b^{(1)})^T(K(A_+,C^T)w^{(1)}+e_1b^{(1)})+c_1e_2{}^Tq \qquad (9)$$
$$s.t.\quad -(K(A_-,C^T)w^{(1)}+e_2b^{(1)})+q\geq e_2,\quad q\geq 0,$$
$$(TSVM2)\quad \min_{w^{(2)},b^{(2)},q} \quad \frac{1}{2}(K(A_-,C^T)w^{(2)}+e_2b^{(2)})^T(K(A_-,C^T)w^{(2)}+e_2b^{(2)})+c_2e_1{}^Tq \qquad (10)$$
$$s.t.\quad (K(A_+,C^T)w^{(2)}+e_1b^{(2)})^T+q\geq e_1,\quad q\geq 0.$$

where $c_1>0$ and $c_2>0$ are parameters and e_1 and e_2 are vectors of ones of appropriate dimensions.
By using Lagrange method and KKT conditions, their dual problems are obtained in the following.

$$(DTSVM1)\quad \max_{\alpha}\quad e_2{}^T\alpha-\frac{1}{2}\alpha^TR(S^TS)^{-1}R^T\alpha \qquad s.t.\quad 0\leq\alpha\leq c_1, \qquad (11)$$

where $S=[K(A_+,C^T)\ \ e_1]$ and $R=[K(A_-,C^T)\ \ e_2]$.
Similarly, the dual problem for $TSVM2$ is obtained as follows.

$$(DTSVM2)\quad \max_{\gamma}\quad e_1{}^T\gamma-\frac{1}{2}\gamma^TL(N^TN)^{-1}L^T\gamma \qquad s.t.\quad 0\leq\gamma\leq c_2, \qquad (12)$$

where $L=[K(A_+,C^T)\ \ e_1]$ and $N=[K(A_-,C^T)\ \ e_2]$.

3. EXPERIMENTAL RESULTS AND ANALYSIS

For comparison of the different SVM classifier, ten data sets are chosen from the UCI and Stalog database. The detailed characteristics are as shown in Table 1. In the following experiments, we extract 70% of data set as the training set using random method for each data set, respectively, and the remaining 30% of data set is viewed as the testing set. The experimental results are the average accuracies after running 10 times.

Table 1: Features of datasets

Number	Dataset	Number of	Number of
1	ionosphere	351	33
2	bupa	345	6
3	sonar	208	60
4	heart	270	13
5	wpbc	194	33
6	australian	695	14
7	cancer	683	7
8	german	1000	24
9	wdbc	569	30
10	pima	768	8

Table 2: Classification performance of C-SVM with gauss kernel and linear kernel using different value of parameter C

Datasets		C=10	C=100	C=500	C=1000	C=5000	C=10000
ionosphere	Gauss kernel	0.9510±0.0139	0.9434±0.0204	0.9311±0.0295	0.9255±0.0346	0.9509±0.0116	0.9359±0.0188
	Linear kernel	0.8698±0.0226	0.8594±0.0396	0.8613±0.0327	0.8547±0.0205	0.8613±0.0361	0.8651±0.027
bupa	Gauss kernel	0.7173±0.0408	0.7173±0.0408	0.7442±0.0333	0.7267±0.0352	0.7183±0.0276	0.7±0.0455
	Linear kernel	0.6615±0.0347	0.6817±0.057	0.6809±0.0301	0.6817±0.0487	0.6577±0.0311	0.6712±0.0422
sonar	Gauss kernel	0.8746±0.033	0.8619±0.0375	0.8556±0.0452	0.8714±0.0542	0.8635±0.0456	0.8825±0.06
	Linear kernel	0.7524±0.0368	0.7254±0.0508	0.7317±0.0663	0.7556±0.0475	0.7365±0.048	0.7302±0.0359
heart	Gauss kernel	0.5543±0.031	0.5123±0.0895	0.5210±0.0584	0.5556±0.0476	0.5432±0.0372	0.5617±0.0413
	Linear kernel	0.8185±0.0436	0.7988±0.0469	0.8395±0.0285	0.826±0.0282	0.8272±0.054	0.7543±0.1439
wpbc	Gauss kernel	0.7559±0.0537	0.778±0.0533	0.7441±0.0521	0.7424±0.0373	0.7661±0.0643	0.7746±0.0506
	Linear kernel	0.4373±0.2747	0.6458±0.2255	0.5946±0.3034	0.6915±0.2694	0.5729±0.2506	0.5424±0.2853
australian	Gauss kernel	0.5404±0.0271	0.5471±0.0428	0.5466±0.0307	0.5404±0.0281	0.5572±0.0271	0.5644±0.0278
	Linear kernel	0.5246±0.079	0.4808±0.0728	0.5312±0.0767	0.5173±0.081	0.5442±0.0632	0.5284±0.0835
cancer	Gauss kernel	0.9532±0.0218	0.941±0.0215	0.9449±0.0146	0.9415±0.0134	0.9415±0.0145	0.939±0.0197
	Linear kernel	0.962±0.0081	0.9668±0.0102	0.9663±0.0118	0.9585±0.0096	0.9615±0.0104	0.964±0.0132
german	Gauss kernel	0.6917±0.0312	0.708±0.0245	0.702±0.0279	0.6933±0.0211	0.7097±0.0215	0.686±0.0284
	Linear kernel	0.757±0.0217	0.7713±0.0256	0.772±0.0215	0.7413±0.0232	0.7547±0.0234	0.7677±0.0268
wdbc	Gauss kernel	0.6269±0.0273	0.6146±0.036	0.607±0.0368	0.6257±0.0294	0.6445±0.027	0.6205±0.0247
	Linear kernel	0.3632±0.0327	0.3813±0.0379	0.3801±0.0381	0.3754±0.0313	0.3755±0.0227	0.3796±0.0223
pima	Gauss kernel	0.6524±0.0278	0.6567±0.0166	0.6416±0.0263	0.6615±0.0205	0.6593±0.0184	0.6559±0.0191
	Linear kernel	0.7671±0.0268	0.7489±0.0608	0.7623±0.0351	0.6775±0.173	0.6134±0.2016	0.6277±0.1695

Table 3: Classification performance of *v*-SVM with gauss kernel and linear kernel using different value of parameter *v*

Datasets		v=0.1	v=0.2	v=0.3	v=0.4	v=0.5	v=0.6
ionosphere	Gauss kernel	0.9377±0.0167	0.9585±0.0184	0.933±0.018	0.8708±0.123	0.933±0.0175	0.933±0.0175
	Linear kernel	0.8236±0.0547	0.8651±0.0385	0.8519±0.0327	0.8218±0.0977	0.8472±0.0338	0.8585±0.0278
bupa	Gauss kernel	0.5356±0.0499	0.5788±0.037	0.6048±0.0625	0.6817±0.0364	0.701±0.0305	0.7241±0.0413
	Linear kernel	0.4702±0.0558	0.5±0.0751	0.5086±0.0745	0.4683±0.1027	0.5317±0.1066	0.5317±0.094
sonar	Gauss kernel	0.8603±0.0579	0.8635±0.0418	0.8788±0.0503	0.8714±0.0413	0.8318±0.0425	0.8318±0.0657
	Linear kernel	0.7111±0.0489	0.7572±0.0644	0.7095±0.056	0.6937±0.0565	0.7762±0.0628	0.7572±0.0318
heart	Gauss kernel	0.5482±0.0408	0.55543±0.0375	0.5494±0.0544	0.5815±0.0574	0.579±0.0405	0.5309±0.0651
	Linear kernel	0.5333±0.1121	0.6247±0.141	0.6222±0.1905	0.8272±0.0273	0.8025±0.0591	0.7704±0.0591
wpbc	Gauss kernel	0.7729±0.05	0.756±0.0467	0.7627±0.0492	0.7509±0.0358	0.7712±0.0312	0.7746±0.0358
	Linear kernel	0.6576±0.2002	0.6034±0.21	0.6526±0.2181	0.7034±0.1529	0.5356±0.2686	0.5949±0.234
australian	Gauss kernel	0.5538±0.0388	0.5433±0.0163	0.5529±0.0625	0.5572±0.026	0.5524±0.0339	0.5601±0.0321
	Linear kernel	0.4793±0.1007	0.5245±0.0982	0.5053±0.0895	0.4817±0.0944	0.4942±0.0791	0.4798±0.0972
cancer	Gauss kernel	0.9454±0.0156	0.9488±0.0142	0.9303±0.0126	0.9107±0.0212	0.8994±0.0271	0.8688±0.0288
	Linear kernel	0.9571±0.0121	0.9546±0.0103	0.9615±0.0121	0.96±0.0125	0.9503±0.0156	0.9293±0.018
german	Gauss kernel	0.6835±0.0331	0.6217±0.1725	0.545±0.2056	0.702±0.0126	0.7±0.0236	0.664±0.1279
	Linear kernel	0.6037±0.1252	0.609±0.1459	0.679±0.0536	0.7057±0.0524	0.7047±0.0411	0.701±0.0323
wdbc	Gauss kernel	0.6328±0.0213	0.6099±0.0248	0.6216±0.0247	0.6404±0.0269	0.6228±0.029	0.6187±0.0405
	Linear kernel	0.6287±0.1598	0.5573±0.2794	0.6334±0.1889	0.614±0.1748	0.666±0.0709	0.5532±0.2671
pima	Gauss kernel	0.6559±0.0289	0.6541±0.0287	0.6533±0.0271	0.6385±0.0121	0.642±0.0278	0.6597±0.029
	Linear kernel	0.5139±0.134	0.5261±0.1866	0.5061±0.1451	0.5481±0.1967	0.6576±0.1491	0.761±0.0185

It can be seen from Table 2 that for different value of C, there are obvious differences of accuracy for nine data sets including ionosphere, bupa, sonar, wpbc, australian, cancer, german, wdbc and pima. However, for some data sets, for instance heart, german and pima, performance of C-SVM using linear kernel is superior to that of using Gauss kernel. For example, when C=500, accuracy is 0.8395 using linear kernel whereas accuracy is 0.5210 using Gauss kernel. On the whole, for selected ten data sets, performance of C-SVM using Gauss kernel is superior to that of C-SVM using linear kernel except data sets heart, german and pima. In addition, for different data set, to find an appropriate value of parameter C is also a difficult thing for obtaining better performance of classification and it's a big range for the value of parameter C. Now, let us turn to Table 3. For different value of v, there are some differences of accuracy for all data sets. And performance of v-SVM using Gauss kernel is superior to that of v-SVM using linear kernel except data set cancer. However, to find a better value of v is not a difficult thing as its range is smaller.

For classifiers PSVM and TWSVM, in experiment, the better values of parameters are obtained by cross-validation method. The experimental results are shown in Figure 1 and Figure 2, where x-axis represents different data set, which is denoted as the number of data set, y-axis represents the average accuracy and standard deviation.

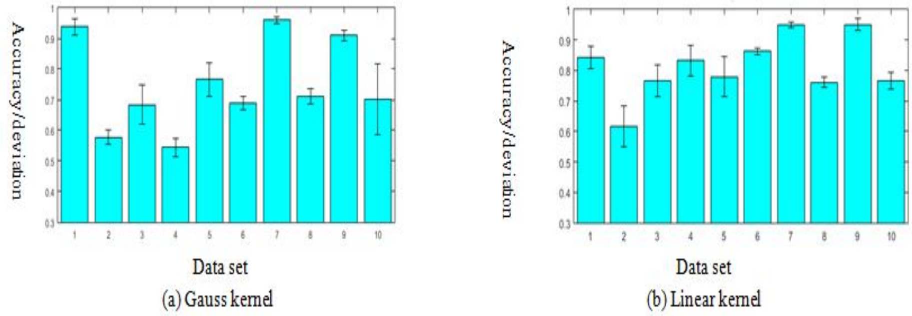

Figure 1 Classification performance of PSVM

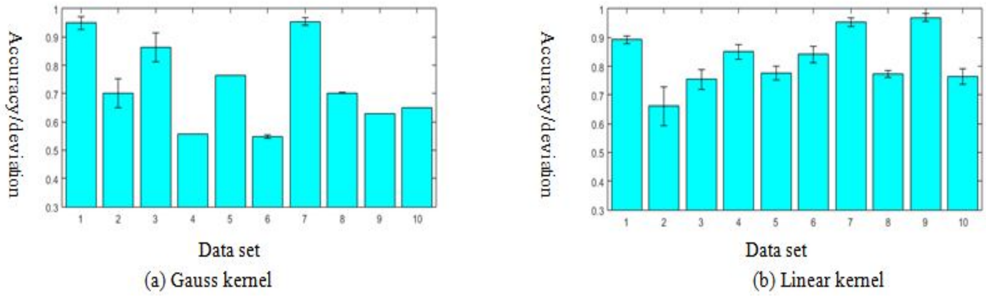

Figure 2 Classification performance of TWSVM

It can be seen from Figure 1 that performance of PSVM using Gauss kernel almost equals to that of PSVM using linear kernel in most data sets. In Figure 2, performance of TWSVM using linear kernel is superior to that of TWSVM using Gauss kernel in most data sets.

Moreover, we give the comparison of performance with different support vector machine containing C-SVM, *v*-SVM, PSVM and TWSVM as shown in Figure 3, where x-axis represents different data set, which is denoted as the number of data set, y-axis represents the average accuracy. Among it, for C-SVM and *v*-SVM, we select the accuracies with C=10 and *v*=0.2, respectively. It is known from Figure 3(a) that for selected ten data sets, accuracy with different support vector machine is almost no differences except data set wdbc using Gauss kernel. When linear kernel is used, the accuracy with TWSVM is superior to other three kinds of support vector machines for ten data sets.

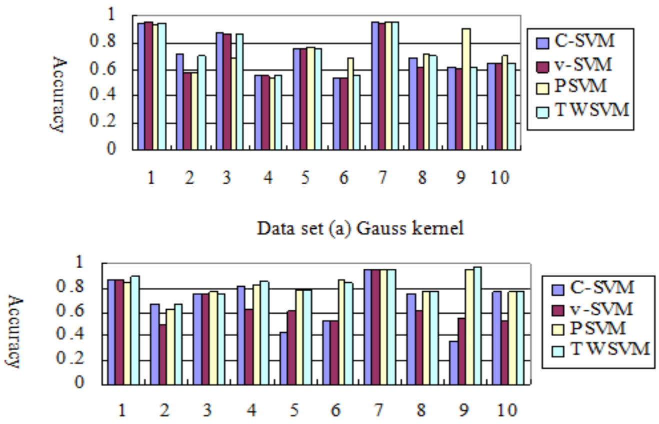

Figure 3 Comparison of performance for different support vector machine

4.CONCLUSION

In this paper, we select the four commonly used support vector machine classifiers including C-SVM, v-SVM, PSVM and TWSVM to conduct to experiment with ten data sets from UCI and Stalog database. For C-SVM and v-SVM, we test their accuracies and standard deviation with different values of parameters C and v using Gauss kernel and linear kernel. For PSVM and TWSVM, the better values of parameters are determined by cross-validation method. It is known from experiment that the performance with TWSVM has an advantage over C-SVM, v-SVM and PSVM in selected ten data sets using linear kernel. However, when Gauss kernel is used, accuracy with different support vector machine is almost no differences except data set wdbc. Certainly, for C-SVM and v-SVM, we only use the selected value to experiment. In future, we select the more support vector machine, for instance, the structural support vector machine etc., to study their performance.

Acknowledgement
This work is supported by Natural Science Foundation of Hebei University under grant 799207217074.

Reference

[1] V. N. Vapnik. The nature of statistical learning theory, Springer, New York, 1995.
[2] B. Scholkopf, A. J. Smola, R. C. Williamson, et al., "New support vector algorithms," Neural Computation, 12(5), pp.1207–1245, 2000.
[3] V. Bloom, I. Griva, B. Kwon, et al., "Exterior-point method for support vector machines," IEEE Transactions on Neural Networks and Learning Systems, 25(7), pp.1390-1393, 2014.
[4] C. F. Lin, S. D. Wang, "Fuzzy support vector machines," IEEE Transaction on Neural Networks, 13, pp.464-471, 2002.
[5] X. W. Yang, G. Q. Zhang, J. Lu, "A kernel fuzzy c-means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises," IEEE Transactions on Fuzzy Systems, 19(1),pp. 105-115, 2011.
[6] Y. T. Xu, "A rough margin-based linear υ support vector regression," Statistics and Probability Letters, 82(3), pp.528-534, 2012.
[7] D. G. Chen, Q. He and X. Z. Wang, "FRSVMs: Fuzzy rough set based support vector machines," Fuzzy Sets and Systems, 161, pp.596–607, 2010.
[8] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, pp. 185–208,1999.
[9] T. Joachims, "Making large-scale SVM learning practical," in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Machine, MIT Press, Cambridge, MA, pp. 169–184, 1999.
[10] D. Yeung, D. Wang, W. Ng, E. Tsang, et al., "Structured large margin machines: sensitive to data distributions," Machine Learning, 68(2), pp.171–200, 2007.
[11] H. Xue, S. C. Chen, Q. Yang, "Structural regularized support vector machine: a framework for structural large margin classifier," IEEE Transactions on Neural Networks,22(4), pp.573-587,2011.
[12] O. Chapelle, "Training a Support Vector Machine in the Primal," Neural Computation, 19(5), pp.1155-1178,2007.
[13] J. Suykens, J. Vandewalle, "Least squares support vector machine classifiers," Neural Processing Letters, 9(3), pp. 293-300, 1999.
[14] G. Fung,O. L. Mangasarian, "Proximal support vector machine classifiers," Proceedings of the 7th International Conference on Knowledge and Data Discovery, pp.77–86,2001.
[15] O. L. Mangasarian, E. W. Wild, "Multisurface proximal support vector classification via generalized eigenvalues," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28: 69-74.
[16] R. K. Jayadeva, R. Khemchandani, S. Chandra, "Twin support vector machine for pattern classification," IEEE Transactions on Pattern Analysis and Machine Intelligence, 29, pp. 905-910,2007.
[17] J. A. Nasiri, N. M. Charkari, S. Jalili, "Least squares twin multi-class classification support vector machine," Pattern Recognition, 48(3), pp.984-992, 2015.
[18] R. Khemchandani, S. Sharma, "Robust least squares twin support vector machine for human activity recognition," Applied Soft Computing,47, pp.33-46,2016.
[19] X. L. Pan, L. Yao, Y. T. Xu, "K-nearest neighbor based structural twin support vector machine," Knowledge-Based Systems,88, pp.34-44, 2015.

[20] L. Zhang, W. D. Zhou, "Fisher-regularized support vector machine," Information Sciences,343-344, pp.79-93, 2016.

AUTHOR

Kai Li received the B.S. and M.S. degrees in mathematics department electrical engineering department from Hebei University, Baoding, China, in 1982 and 1992,respectively. He received the Ph.D. degree from Beijing Jiaotong University, Beijing, China, in 2001.He is currently a Professor in college of computer science and technology, Hebei University. His current research interests include machine learning, data mining, computational intelligence, and pattern recognition.

Lulu Zhai received the bachelor degree in college of information science and technology from Jiangxi Normal University, Nanchang, Jiangxi, China, in 2016, and she is currently pursuing the M.E. degree in the college of computer science and technology, Hebei University, Baoding, Hebei, China. Her research interests include machine learning, data mining, and pattern recognition.