International Journal of Multimedia and Ubiquitous Engineering
Vol. 9, No. 10 (2014), pp. 79-90
http://dx.doi.org/10.14257/ijmue.2014.9.10.08

Research and Implementation of Support Vector Machine and Its Fast Algorithm

Dong Ai-mei

QiLu University of Technology, JiangNan University
amdong@163.com

Abstract

Support vector machine is an effective pattern classification method. Its time complexity and space complexity is $O(n^3)$ and $O(n^2)$ respectively for training samples with scale of n. Meanwhile, for core vector machine, the relation between time complexity and the scale of training samples is linear and space complexity is independent with the scale of training samples. In this paper, for the problem of big data classification, the concept and principle of support vector machine are described and support vector machine is converted into the form of minimum enclosing ball, consequently core vector machine is used to efficiently obtain the approximate optimal solution. Experiments confirm that core vector machine algorithm can classify the big data quickly and efficiently.

Keywords: *Support Vector Machine (SVM); Minimum Enclosing Ball (MEB); Core Vector Machine (CVM); Big Data (BD)*

1. Introduction

Inducing the motion laws of the system from observed data and then using these motion laws to predict future data or not observed data have always been a focus in the field of artificial intelligence. Traditional learning method using empirical risk minimization rule can minimize the train error, but cannot guarantee to maximize the generalization ability of the system. Therefore, Vapnik [1, 2] proposed the rule of structural risk minimization which can maximize generalization ability by minimizing the upper bound of errors. Support Vector Machine (SVM) proposed by Vapnik is the implementation of structure risk minimization. The basic idea of SVM [3, 4] is to construct the optimal hyperplane in the feature space of samples, and then maximize the distance between hyperplane and different sample sets, therefore can achieve maximum generalization ability. SVM manifests many advantages in solving small sample、 nonlinear and high dimensional pattern recognition problems and overcomes the "curse of dimensionality" and "over learning" problems to a great extent. SVM with the global optimality and good generalization ability has solid theoretical foundations and simple mathematical model. so in areas such as pattern recognition, regression analysis, function estimation, time series prediction SVM has got considerable development.

The standard support vector machine learning algorithm can be reduced to solving a constrained quadratic programming problem. For small-scale quadratic optimization problem, mature classical optimization algorithm such as Newton's method and interior point method can be used to solve it. However, the essence of training process of SVM is to solve a quadratic programming problem, so the time complexity is $O(n^3)$ and space complexity is $O(n^2)$ when handling problems with scale of .If the training set size is very large, the training time of SVM will be too long, at the same time, the size of the kernel matrix will be too big and lead to insufficient memory. Consequently the research of using SVM to handle

ISSN: 1975-0080 IJMUE

large scale problems is imperative. In this paper, the essence of SVM is used to transform the kernel method into geometric problems. Based on approximation of SVM's solving process, I. W. Tsing [5] proposed to transform the kernel method into problem of minimum enclosing ball (MEB) and put forward the core vector machine (CVM) algorithm. The equivalence between SVM and MEB is proved, accordingly the fast algorithm CVM is used to solve the large scale problem. Firstly, CVM selects a sample as the core vector using some heuristic rules, meanwhile calculates the radius and center of the initial MEB; Secondly, CVM finds the sample farthest away from the center of MEB and includes the sample into core vector sets; Thirdly, CVM calculates the radius and center of the new MEB. The above process is repeated until no samples can be included, a MEB of the whole training set is obtained. Finally, based on the equivalence between SVM and MEB, the optimal classification hyperplane is obtained. Because of time complexity of solving MEB is linear with the scale of training sample and space complexity is independent of the scale of training sample, the time complexity of CVM is linear with the scale of training sample and the space complexity is independent of the scale of training samples. Simulated results show that, CVM has the same generalization ability as standard SVM, but a shorter solving time, furthermore is effective for big data.

The remainder of this paper is organized as follows. Section 2 discusses systematically principle and algorithm of SVM. Section 3 studies the thought and theory of MEB. Section 4 reports that two class SVM and one class SVM can be transformed into MEB problem, then can be implemented using CVM algorithm. The experimental results and analysis are given in section 5. Conclusions are given in the final section.

2. Principle of SVM

SVM is proposed with the optimal classification separate plane from the linear perspective. The mechanism of SVM is to seek an optimal classification hyperplane satisfying requirements. In theory, SVM can achieve the optimal classification for linear data. Considering the 2-dimensionality two class linear separate condition shown as Figure 1, solid and hollow points represent two kinds of training samples, H is straight line separating two classes with no errors, H1 and H2 are straight lines respectively by points of two classes nearest and parallel to H, the distance between H1 and H2 is called classification margin of two class. The optimal classification line not only can classify two classes correctly but also maximize the margin between two classes. The former is the guarantee of empirical risk minimization; the latter is to minimize the confidence limits of generalization bounds, so that the real risk is minimized. Generalized to high dimension space, the optimal classification line is called the optimal hyperplane. In this paper, linear and nonlinear conditions for two class are discussed. For training samples $T = \{(x_i, y_i), x_i \in \square^d, y_i \in \{+1, -1\}, i = 1, 2, ..., n\}$, the general form of the classification hyper plane is $g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$. Relevant document confirms that classification hyperplane with form of $g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$ has better performance, so the form is used in this paper.

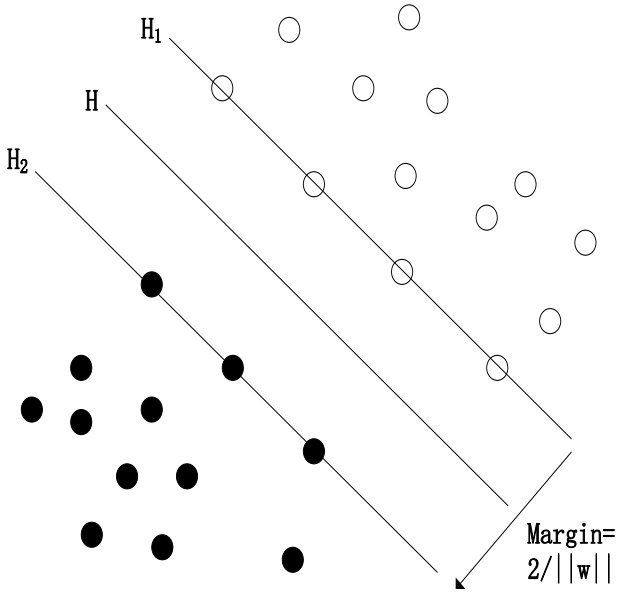

Figure 1. The Optimal Classification Hyperplane of SVM

2.1. The Linear Case

For training samples $T = \{(\boldsymbol{x}_i, \boldsymbol{y}_i), \boldsymbol{x}_i \in \mathbf{x}, \boldsymbol{y}_i \in \mathbf{y}, i = 1, 2, \ldots, n\}$, $\mathbf{x} \in \square^d, \mathbf{y} = \{+1, -1\}$, if there exits classification hyper plane and $\boldsymbol{w} \in \square^d$

$$\boldsymbol{w}^T \boldsymbol{x} = 0 \tag{1}$$

subject to

$$\begin{aligned} &\boldsymbol{w}^T \boldsymbol{x}_i \geq 1, y_i = 1, \\ &\boldsymbol{w}^T \boldsymbol{x}_i \leq -1, y_i = -1, i = 1, 2, \ldots, n \end{aligned} \tag{2}$$

Then T is linearly separable. Equation (2) is rewrite as equation (3):

$$y_i \boldsymbol{w}^T \boldsymbol{x}_i \geq 1, i = 1, 2, \ldots, n \tag{3}$$

Accordingly the discriminant function $y(\boldsymbol{x}) = sign(\boldsymbol{w}^T \boldsymbol{x})$ is obtained. To obtain the optimal hyperplane needs to maximize $\frac{2}{\|\boldsymbol{w}\|^2}$, that is to minimize $\frac{1}{2}\|\boldsymbol{w}\|^2$, so the object function is as follow:

$$\begin{aligned} &\arg\min_{w} J_1 = \arg\min_{w} \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &s.t. \quad y_i \boldsymbol{w}^T \boldsymbol{x}_i \geq 1 \;\; i = 1, 2, \ldots, n \end{aligned} \tag{4}$$

Equation (4) is a quadratic programming problem. If T is not linearly separable, non-negative slack variables $\xi_i, i = 1, 2, \ldots, n$ are introduced and the optimization problem to obtain classification hyperplane is as follow:

$$\begin{aligned} &\arg\min_{w,\xi_i} J_2 = \arg\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &s.t. \quad y_i w^T x_i \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0, i = 1,2,\ldots,n \end{aligned} \tag{5}$$

C is penalty parameter, the bigger C value is , the greater the misclassification punishment is. Lagrange multiplier method is used to solve the quadratic programming problem with linear constraints and the dual optimization problem is as follow:

$$\begin{aligned} &\arg\min_{\alpha_i} J_3 = \arg\min_{\alpha_i} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{n}\alpha_i \\ &s.t. \quad 0 \leq \alpha_i \leq C, i = 1,2,\ldots,n \end{aligned} \tag{6}$$

The optimal solution $\alpha^* = (\alpha_1^*,\ldots,\alpha_n^*)^T$ is achieved from object function (6), consequently classification hyperplane parameter is achieved $w = \sum_{i=1}^{n}\alpha_i y_i x_i$.

2.2. The Nonlinear Case

When T is not linearly separable, transformation from input space R^d to Hilbert space H: $\Phi: X \subset R^d \to H$, accordingly, $\overline{T} = \{(\Phi(x_i), y_i), \Phi(x_i) \in H, y_i \in \{+1,-1\}, i = 1,2,\ldots,n\}$ and $x \to X = \Phi(x)$ classification hyperplane is as follow:

$$w^T \varphi(x) = 0 \tag{7}$$

The discriminant function is $y(x) = sign(w^T \varphi(x))$. Object function to solve optimal classification hyperplane is as follow:

$$\begin{aligned} &\arg\min_{w,\xi_i} J_4 = \arg\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &s.t. \quad y_i w^T \Phi(x_i) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0, i = 1,2,\ldots,n \end{aligned} \tag{8}$$

Correspondingly, the dual optimization problem is as follow:

$$\begin{aligned} &\arg\min_{\alpha_i} J_5 = \arg\min_{\alpha_i} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i,x_j) - \sum_{i=1}^{n}\alpha_i \\ &s.t. \quad 0 \leq \alpha_i \leq C, i = 1,2,\ldots,n \end{aligned} \tag{9}$$

$K(x_i,x_j) = [\varphi(x_i)]^T \varphi(x_j)$ is kernel function. $\alpha^* = (\alpha_1^*,\ldots,\alpha_n^*)^T$ is obtained from object function (9) and $w = \sum_{i=1}^{n}\alpha_i y_i \varphi(x_i)$ is obtained, consequently discriminate function is $f(x) = \mathrm{sgn}(\sum_{i=1}^{n}\alpha_i y_i K(x_i,x))$.

T is projected into high dimensionality or infinite feature space through nonlinear transformation $\varphi : X \subset \mathbb{R}^d \to H$ $x \to X = \varphi(x)$ and optimal classification hyperplane is constructed in the feature space, but in the process of solving optimization problem and calculating discriminant function, nonlinear function $\varphi(x)$ need not to explicitly compute and only need to compute the kernel function, accordingly, the curse of dimensionality in feature space is avoided. The choice of kernel function must satisfy the Mercer condition. The commonly used Kernel function are linear function $k(\boldsymbol{x}_i, \boldsymbol{x}) = \boldsymbol{x}_i^T \boldsymbol{x}$, polynomial function $k(\boldsymbol{x}_i, \boldsymbol{x}) = (\boldsymbol{x}_i^T \boldsymbol{x} + 1)^p$, Gauss kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}) = \exp(-\|\boldsymbol{x} - \boldsymbol{x}_i\|^2 / \sigma^2$, and so on.

3. Principle of MEB

3.1. MEB in Computational Geometric

Considering sample sets $T = \{x_1, x_2, ..., x_n\}$, MEB in computational geometric is minimal ball including all samples in T. Approximate MEB based on core set is discussed in the following. $B(c, R)$ denotes a ball with center c and radius R. Given $\varepsilon > 0$, if $R \le r_{MEB(T)}$ and $T \subset B(c,(1+\varepsilon)R)$ then $B(c,(1+\varepsilon)R)$ is a $(1+\varepsilon)$-approximation of MEB(T),shown as Fig.2. The inner circle is the MEB of the set of squares and its (1+e) expansion (the outer circle) covers all the points. The set of squares is thus a core set. In many shape fitting problems, it is found that solving the problem on a subset, called the core set, Q of points from T can often give an accurate and efficient approximation. More formally, if $B(c,R) = MEB(Q)$ and $T \subset B(c,(1+\varepsilon)R)$, then Q is core set of T with $Q \subseteq T$.

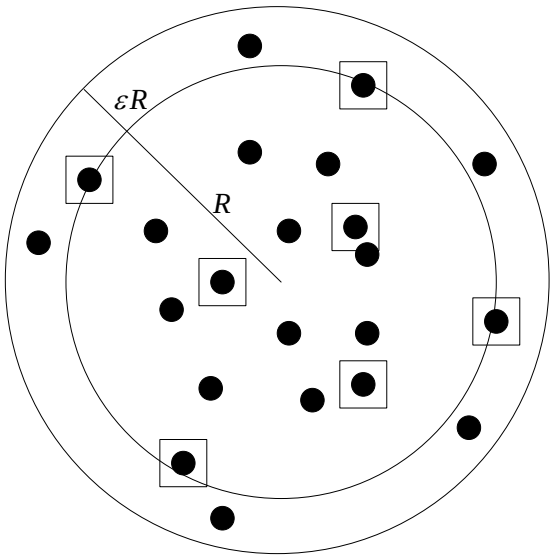

Figure 2. The Demonstration of MEB in Computational Geometric

In order to obtain the core set, Badoiu and Clarkson in 2002 proposed a simple iterative algorithm [6] to calculate an (1+ε)-approximation of MEB. The specific steps are as follows: At the *t*th iteration, the current estimate $B(c_t, r_t)$ is expanded

incrementally by including the furthest point outside the (1+ε)-ball $B(c_t,(1+\varepsilon)r_t)$. This is repeated until all the points in T are covered by $B(c_t,(1+\varepsilon)r_t)$. Despite its simplicity, the number of iterations, and the size of the final core set, depends only on ε but not on dimensionality d or the size of T n. The independence of d is important on applying this algorithm to kernel methods as the kernel-induced feature space can be infinite-dimensional. As for the remarkable independence on n, it allows both the time and space complexities of algorithm to grow slowly.

3.2. The Theory and Kernel Method of MEB

The primal object function of MEB is as follow:

$$\begin{aligned} &\arg\min_{c,R} J_5 = \arg\min_{c,R} R^2 \\ &s.t. \quad \left\|c - \varphi(\boldsymbol{x}_i)\right\|^2 \le R^2 \quad i = 1,\ldots,m \end{aligned} \tag{10}$$

φ is feature mapping corresponding to kernel k, $B(c,R)$ is MEB in kernelized feature space. The corresponding dual problem is quadratic programming problem as follow:

$$\begin{aligned} &\arg\max_{\alpha} J_6 = \arg\max_{\alpha} \boldsymbol{\alpha}^T diag(K) - \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \\ &s.t. \quad \boldsymbol{\alpha}^T 1 = 1, \quad \boldsymbol{\alpha} \ge \boldsymbol{0} \end{aligned} \tag{11}$$

$\boldsymbol{\alpha} = [\alpha_1,\ldots,\alpha_m]^T$ is Lagrange multiplier vector, $K_{m\times m} = \left[k(\boldsymbol{x}_i,\boldsymbol{x}_j)\right] = \left[\varphi(\boldsymbol{x}_i)^T\varphi(\boldsymbol{x}_j)\right]$ is the corresponding kernel matrix. Given k satisfying

$$k(\boldsymbol{x},\boldsymbol{x}) = \kappa \tag{12}$$

is constant. Equation (11) can be rewrite as:

$$\begin{aligned} &\arg\max_{\alpha} J_7 = \arg\max_{\alpha} -\boldsymbol{\alpha}^T K \boldsymbol{\alpha} \\ &s.t. \quad \boldsymbol{\alpha}^T 1 = 1, \quad \boldsymbol{\alpha} \ge \boldsymbol{0} \end{aligned} \tag{13}$$

The primitive variable of MEB can be obtained by solving the optimal value of equation (13):

$$c = \sum_{i=1}^{n} \alpha_i \varphi(\boldsymbol{x}_i), \qquad R = \sqrt{\alpha' diag(K) - \alpha' K \alpha} \tag{14}$$

Thus, MEB can be transformed into quadratic programming problem in kernelized feature space. Theory of MEB points out, quadratic programming problem being form of equation (13) with kernel functions satisfying equation (12) can be regarded as MEB, which can use approximate MEB algorithm to solve quadratic programming problem, so time complexity and space complexity solving quadratic programming problem can be decreased.

3.3. The Relationship between SVM and MEB

In literature [7], Tsang reveals that quadratic programming problem satisfying certain conditions is equivalent to MEB problem and can be solved by the iterative algorithm which

is proposed by Badoiu in 2002 for solving big data problems and calculates approximate MEB problem.

In nonlinear SVM, the classification intervals is set from 1 to a soft interval and the objective function becomes slightly as follow:

$$\begin{aligned} &\arg\min_{w,\xi_i} J_8 = \arg\min_{w,\xi} \frac{1}{2}\|w\|^2 - v\rho + C\sum_{i=1}^{n}\xi_i^2 \\ &s.t. \quad y_i w^T \Phi(x_i) \geq \rho - \xi_i \end{aligned} \tag{15}$$

Its dual function is:

$$\begin{aligned} &\arg\max_{\alpha_i} J_9 = \arg\max_{\alpha_i} -\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j(y_i y_j K(x_i,x_j) + y_i y_j + \frac{\delta_{ij}}{2c}) \\ &s.t. \quad \sum_{i=1}^{n}\alpha_i = v, \alpha_i \geq 0, i = 1,2,\ldots,n \end{aligned} \tag{16}$$

In Eq.(16), $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$, Letting $\frac{\alpha_i}{v} \to \alpha_i$ and Eq.(16) becomes

$$\begin{aligned} &\arg\max_{\alpha_i} J_{10} = \arg\max_{\alpha_i} -\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j(y_i y_j K(x_i,x_j) + y_i y_j + \frac{\delta_{ij}}{2c}) \\ &s.t. \quad \sum_{i=1}^{n}\alpha_i = 1, \alpha_i \geq 0, i = 1,2,\ldots,n \end{aligned} \tag{17}$$

Letting $K(x_i,x_j) = y_i y_j K(x_i,x_j) + y_i y_j + \frac{\delta_{ij}}{2c}$ then $K(x,x) = K(x,x) + 1 + \frac{1}{2c}$ .If kernel function $K(x,x) = \kappa$,then $K(x,x) = \kappa + 1 + \frac{1}{2c}$ is a constant, thus the dual problem of nonlinear SVM satisfies Eq.(12) (13) and a conclusion that classification problem in one feature space becomes MEB problem in another space is got.

4. CVM Algorithm

4.1. The Step of CVM Algorithm

Classification problem in one feature space becomes MEB problem in another space and can be solved quickly and approximately by CVM algorithm. For training set $T = \{z_i = (x_i, y_i), x_i \in R^d, y_i \in (+1,-1), i = 1,2,\ldots,n\}$, mapping kernel function φ is adopted corresponding to transformed kernel K .The CVM algorithm is as follow:

(a) Initialize S_0, c_0, R_0

(b)Terminate if there is no training point z such that $\varphi(z)$ falls outside the (1+ ε)-ball $B(c_t, (1+\varepsilon)R_t)$.

(c)Find z such that $\varphi(z)$ is furthest away from c_t . Set $S_{t+1} = S_t \cup \{z\}$.

(d)Find the new MEB(S_{t+1}) .Set $c_{t+1} = c_{\mathrm{MEB}(S_{t+1})}, R_{t+1} = R_{\mathrm{MEB}(S_{t+1})}$.

(e) $t = t + 1$ and return to step(b)

4.2. The Detailed Process of CVM Algorithm

4.2.1. Initialization: We start with an arbitrary point $z \in S$ and find $z_a \in S$ that is furthest away from z . Then, we find another point $z_b \in S$ that is furthest away from z_a . Obviously, the initial core set is $S_0 = \{z_a, z_b\}$, $c_0 = \frac{1}{2}(\varphi(z_a) + \varphi(z_b))$, $\alpha_a = \alpha_b = \frac{1}{2}$, $R_0 = \frac{1}{2}\left\|\varphi(z_a) - \varphi(z_b)\right\|$.

4.2.2. Distance Computations: Step (b) and (c) involving computing $\left\|c_t - \varphi(z_l)\right\|^2$ for $z_l \in S$. On using $c = \sum_{z_i \in S_t} \alpha_i \varphi(z_i)$, we have $\left\|c_t - \varphi(z_l)\right\|^2 = \sum_{z_i, z_j \in S_t} \alpha_i \alpha_j k(z_i, z_j) - 2\sum_{z_i \in S_t} \alpha_i k(z_i, z_l) + k(z_l, z_l)$. Hence, computations are based on kernel evaluations instead of the explicit $\varphi(z_l)$, which may be infinite-dimensional. However, calculating point farthest away from the center point requires calculating the distance between all the other points and the center point. When the number of samples is very large, the time cost is very huge. In this paper, CVM uses an accelerated method[8] proposed by Smola and Scholkopf in 2000. The idea is to randomly sample a sufficiently large subset S_0 from S , and then take the point in S_0 that is furthest away from c_t as the approximate furthest point over S . As shown in (Smola and Sch¨olkopf, 2000), by using a small random sample of, say, size 59, the furthest point obtained from S_0 is with probability 0.95 among the furthest 5% of points from the whole S .This method can greatly reduce the time complexity and can also be used for the initialization step.

5. Experiments and Analysis

5.1. Experiment Settings

The experiment result is reported from two aspects: firstly, the performance of CVM and SVM is compared on accuracy and solving time; secondly, the sensitivity of parameter ε on performance of CVM is analyzed. The banana datasets with different scales are adopted in experiment.

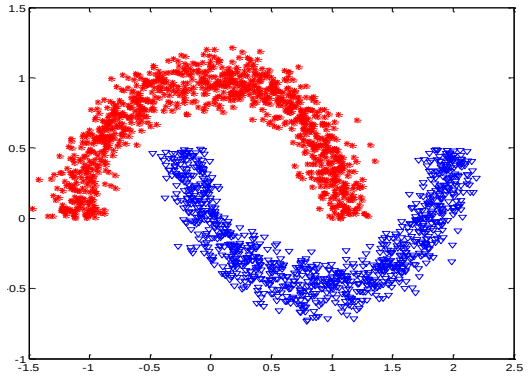

Figure 3. Artificial Banana Dataset

C parameter and kernel parameter γ are needed in SVM; approximation parameter ε, C parameter and kernel parameter γ are needed in CVM. Followed as experience, the domain of C parameter is $\{0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100\}$; Gaussian kernel function, *i.e.*, $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{\gamma})$, is adopted by CVM and SVM, and the kernel width γ is determined from $\left\{\frac{\sigma^2}{32}, \frac{\sigma^2}{16}, \frac{\sigma^2}{8}, \frac{\sigma^2}{4}, \frac{\sigma^2}{2}, \sigma^2, 2\sigma^2, 4\sigma^2, 8\sigma^2, 16\sigma^2\right\}$, where σ is the standard deviation of the corresponding dataset; the domain of approximation parameter ε is $\{1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7, 1e-8, 1e-9, 1e-10\}$.

5.2. Experiment and Result Analysis

Table 1 records training time, test time and classification accuracy of SVM and CVM with different training set size. The size of core set of CVM algorithm is also recorded. For all the algorithms, the related parameters are determined by the grid-search strategy. When these parameters are fixed, each algorithm is carried out ten times and the corresponding average classification performance is recorded.

Table 1. Performance Comparison of CVM and SVM

size of sample		size of core set	solving time(unit:sec)		classification accuracy	
training set	testing set		SVM	CVM	SVM	CVM
100	1000	62.5±4.3	16.6292±0.0023	22.3848±0.0023	0.8558±0.0226	0.8421±0.0012
500	1000	189±13.7	21.9042±0.0047	22.8808±0.0047	0.8921±0.0009	0.9013±0.0200
1000	1000	253.7±8.9	31.1062±0.0102	23.5883±0.0209	0.9120±0.0032	0.9238±0.0201
2000	1000	432.2±18.1	41.0539±0.0034	23.5507±0.0109	0.9239±0.0012	0.9169±0.0089
3000	1000	498.2±30.1	72.2946±0.0056	24.0588±0.0067	0.9421±0.0035	0.9478±0.0019
4000	1000	623.6±6.9	166.8772±0.0019	26.3078±0.0900	0.9419±0.0123	0.9398±0.0104
5000	1000	671.4±3.8	-	26.3657±0.0701	-	0.9538±0.0203
7500	1000	948±23.1	-	28.7238±0.0602	-	0.9609±0.0078
10000	1000	1547±19.8	-	83.0154±0.0023	-	0.9529±0.0830
12500	1000	2269±12.3		161.5609±0.0067		0.9610±0.0023
15000	1000	2269±14.6	-	238.4546±0.0028	-	0.9489±0.0109

Note: "-" represents out of memory in matlab when SVM runs.

CVM is run using banana set with scale of 10000, Table 2 shows the influence of approximation parameter with different values.

Table 2. The Influence of Approximation Parameter ε with Different Values on CVM ($C = 10, \gamma = \frac{\sigma^2}{2}$)

ε	classification accuracy	size of core set	solving time(unit:sec)
1e-1	0.8997	130	0.8976
1e-2	0.9087	352	58.7865
1e-3	0.9100	649	295.3524
1e-4	0.9178	1025	1534.6548
1e-5	0.9230	1345	1647.3965
1e-6	0.9500	1520	1749.3621
1e-7	0.9526	1548	1802.1400
1e-8	0.9538	1987	2350.8649
1e-9	0.9620	3978	10367.4789
1e-10	0.9620	5134	29569.3420

From Table 1, the following observations can be made: for small datasets, the classification accuracy of CVM is almost the same as that of SVM and the solving time of SVM is less than that of CVM; for big datasets, the solving time of CVM is much less than that of SVM and the classification accuracy of CVM is almost the same as that of SVM. Using the core set instead of all the samples, the storage space is reduced on a large scale. So the CVM algorithm can solve problem of big data.

As can be seen from Table 2, the smaller approximation parameter ε is, the smaller classification error is and the longer the solving time is. In this experiment, the parameter ε with $1e-7$ can get the best coordination between classification accuracy and training time.

6. Conclusion

Pattern classification is an important research branch in the field of pattern recognition and SVM method is an effective method to solve the problem of pattern classification, but in background of big data , due to the constraints of time and space complexity , SVM cannot solve pattern classification problems efficiently; CVM is a new pattern classification method, it transforms kernel method into an equivalent MEB problem and is combined with computational geometry, thus CVM solves the problem of time and space requiring in processing big data. Experiments show that compared with SVM, CVM can not only ensure the classification accuracy but also economize running time and storage space.

Acknowledgements

This work was supported by the Project of Shandong Province Higher Educational Science and Technology Program under Grant J14LN05.

References

[1] V. N. Vapnik, “The nature of statistical learning”, Berlin: Springer (**1995**).
[2] V. N. Vapnik, “Statistical learning theory”, New York: John Wiley & Sons (**1998**).
[3] N. Cristianini and J. S. Taylor, “An introduction to support vector machines and other kernel-based learning methods”, New York: Cambridge university press (**2000**).

[4] X.-G. Zhang, "Introduction to statistical learning theory and support vector machines", Acta Automatica Sinica, vol. 1, no. 26, (**2000**).
[5] I. Tsang, J. Kwok and P. Cheung, "Core vector machines: Fast SVM training on very large data sets", Journal of Machine Learning Research, vol. 4, no. 6, (**2005**).
[6] M. Badoiu and K. L. Clarkson, "Optimal core sets for balls", Computational Geometry: Theory and Applications, vol. 1, no. 40, (**2008**).
[7] I. Tsang, J. Kwok and J. Zurada, "Generalized core vector machines", IEEE Transactions on Neural Networks, vol. 5, no. 17, (**2006**).
[8] A. Smola and B. Schokopf, "Sparse greedy matrix approximation for machine learning", Proceedings of the 17th International Conference on Machine Learning, (**2000**) August 21-23, San Francisco, America.

Author

Dong Ai-mei, she was born in 1978. She is a lecture in QiLu University of Technology and received M.S degree in Computer Science from Ocean University of China, China, in 2004. Now she is a Ph.D. candidate at the School of Digital Media, Jiangnan University. She has published more than ten papers in international/national authoritative journals. Her research interests include data mining, pattern recognition, intelligent computation and their applications.