Improving SVM through a Risk Decision Rule Running on MATLAB

Xuemei Zhang
School of Economics and Business,
Regional Logistics Planning and Modern Logistics Engineering Laboratory,Fuyang Teachers College,
Fuyang, China, 236037
E-mail: xmz@mail.ustc.edu.cn

Li Yang
School of Economics and Management,
Anhui University of Science & Technoloty, Huainan, China, 232001
E-mail: y321212@163.com

***Abstract*—Support Vector Machine (SVM) is a classification technique based on Structural Risk Minimization (SRM), which can run on MATLAB. For classification of nonseparable samples, conventional SVM needs to select a tradeoff between maximization the margin and misclassification rate. In order to guarantee generalized performance and low misclassification rate of SVM, this paper puts forward an improved SVM through a risk decision rule for the nonseparable samples running on MATLAB. The improved SVM transforms the outputs of the SVM to posterior probabilities belonging to different classes and samples between the support hyper-planes are classified by using risk decision rule of Empirical Risk Minimization (ERM). Computational results show that the proposed approach is better than conventional SVM remarkably when the two classes are easy to separate, and in other condition, its performance is comparable to conventional SVM.**

***Index Terms*—Support Vector Machine; Structural Risk Minimization; Risk decision rule; MATLAB**

I. INTRODUCTION

Developed by Vapnik [1], Support Vector Machine (SVM) is a machine learning technique for classification based on statistical learning theory. In a classification problem with two classes, SVM constructs an optimal separating hyper-plane that maximizes the margin between two classes. SVM embodies Structural Risk Minimization (SRM) principle, which has been shown to be superior to conventional Empirical Risk Minimization (ERM) employed by conventional neural networks. SRM minimizes an upper bound of generalization error as opposed to ERM that minimizes the misclassification rate on training data, which can run on MATLAB.

SVM is gaining popularity due to many attractive features and excellent generalization performance on a wide range of problems. Min et al. [2] and Avci [3] study SVM on the optimal parameters selection aspect. Hsu et al. [4] and Navia-Vazquez [5] expand the binary classification to multi-class classification. Wang et al. [6] and Lu et al. [7] introduce Least-Square-SVM (LS-SVM) by changing inequality constraints to equality constraints. For fuzzy problems, Liu et al. [8] put forward a Fuzzy SVM based on density clustering. Hua et al. [9] proposed an asymmetric support vector machine for the classification problem with asymmetric cost of misclassification.

For nonseparable samples, the optimization model of SVM incorporates a kernel function and a regularization parameter to minimize the SRM. When the regularization parameter is small, SVM has good generalization ability. Meanwhile, the number of samples lying between support hyper-planes increases, which would result in a higher misclassification rate. Therefore, a better classification rule for these samples is required.

In this paper, we propose a novel SVM through a risk decision rule (RD-SVM) for classifying nonseparable samples running on MATLAB. RD-SVM transforms the outputs of samples lying between support hyper-planes into posterior probabilities and computes an optimal probability threshold based on ERM. The optimal probability threshold and optimal separating hyper-plane (OSH) are used to partition the domain between the support hyper-planes into four intervals, and different classification rules are established based on the number of samples belonging to different classes in each interval. Computational results show that the proposed approach is efficient in improving classification performance compared to conventional SVM.

The rest of this paper is organized as follows. In Section 2, a brief review of conventional SVM for binary classification is presented. Section 3 describes the proposed RD-SVM approach, and in Section 4, the experimental results on several benchmark data sets running on MATLAB are reported. The final section ends the paper with some conclusion remarks.

II. A BRIEF REVIEW OF CONVENTIONAL SVM

doi:10.4304/jsw.7.10.2252-2257

In this section, we provide a brief review of conventional SVM for binary classification. More details about SVM can be found in Vapnik [1].

Given a training set $G=\{x_i, y_i\}$ ($i=1,2,\ldots,l$) with input vector $x_i \in R^s$ and corresponding binary class labels $y_i \in \{+1,-1\}$, for nonlinearly nonseparable data set, the classifier can be constructed by solving the following quadratic programming problem:

$$\min_{\omega,b,\xi_i} \Phi(\omega,\xi_i)=\frac{1}{2}\|\omega\|^2+C\sum_{i=1}^{l}\xi_i$$
$$\text{s.t. } y_i(\omega\varphi(x_i)+b)\ge 1-\xi_i, \tag{1}$$
$$\xi_i \ge 0,\ i=1,2,\ldots,l.$$

where $\varphi(\cdot)$ is a nonlinear function which maps the input space into a higher dimensional space. For model (1), by introducing Lagrange multipliers $\alpha_i \ge 0$ and $\beta_i \ge 0$ ($i=1,2,\ldots,l$) associated with the constraints, the above problem can be transformed into its dual problem.

$$\max_{\alpha_i} \Gamma(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$$
$$\text{s.t. } \sum_{i=1}^{l}\alpha_i y_i=0, \tag{2}$$
$$0\le\alpha_i\le C,\ i=1,2,\ldots,l.$$

The kernel function $K(x_i,x_j)=\varphi(x_i)^T\cdot\varphi(x_j)$ must satisfy Mercer's theorem [10]. Some kernel functions in common use are Polynomial and Radial Basis Function.

To solve the model (2), Lemke complementary pivot algorithm [11-13] is employed. The decision function is as follows.

$$y(x)=\text{sgn}\Big(\sum_{i=1}^{l}y_i\alpha_i^*K(x_i,x)+b^*\Big), \tag{3}$$

where, $\omega^*=\sum_{i=1}^{l}y_i\alpha_i^*x_i$,

$$b^*=\frac{1}{2}[(\omega^*\cdot x^*(+1))+(\omega^*\cdot x^*(-1))],$$

$x^*(+1)$ and $x^*(-1)$ are positive support vector and negative support vectors, respectively.

The distance from one sample to OSH is calculated as following:

$$d(x)=\frac{\sum_{i=1}^{l}y_i\alpha_i^*K(x_i,x)+b^*}{\|\omega\|}. \tag{4}$$

For the standardization SVM, the relative distance from one sample to OSH is calculated as following:

$$f(x)=\frac{d(x)}{\frac{1}{\|\omega\|}}=d(x)\cdot\|\omega\|=\sum_{i=1}^{N}y_i\lambda_i^*K(x,x_i)+b^*. \tag{5}$$

III. An Improved SVM Through A Risk Decision Rule

Conventional SVM classifies samples into two classes by the OSH and is sensitive to samples lying near the OSH, especially between the support hyper-planes. We define the domain between the support hyper-planes as nonseparable domain, in which samples are inclined to be misclassified. For the nonseparable domain, this paper puts forward an improved SVM based on the risk decision rule (RD-SVM) that tries to guarantee the generalization ability of SVM and decrease the misclassification rate simultaneously. The RD-SVM technique partitions the nonseparable domain into four intervals by optimal probability threshold and OSH. In each interval, the number of samples belonging to different classes is used to revise the classification rule.

A. The Classification Rule of RD-SVM

The main idea of RD-SVM is to modify classification rule which is obtained by conventional SVM only for samples in nonseparable domain. Given a test sample x_0, we can calculate its relative distance $f(x_0)$ to OSH by Eq. (5), and then apply the following modified rule for classification.

(1) If $f(x_0)\ge 1$, then the sample x_0 is classified to positive class;

(2) If $f(x_0)\le -1$, then the sample x_0 is classified to negative class;

(3) If $-1< f(x_0)<1$, then the sample x_0 is classified to positive class by probability λ_1 and negative class by probability λ_2, with $\lambda_1+\lambda_2=1$ ($0\le\lambda_1,\lambda_2<1$) [14,15].

λ_1,λ_2 are defined as follows.

$$\lambda_1=\begin{cases} f(x_0) & 0\le f(x_0)<1 \\ 1+f(x_0) & -1<f(x_0)<0 \end{cases},$$

$$\lambda_2=\begin{cases} 1-f(x_0) & 0\le f(x_0)<1 \\ |f(x_0)| & -1<f(x_0)<0 \end{cases}. \tag{6}$$

In the following, we will apply the above classification rule to samples in different intervals of nonseparable domain as Hua et al. [16]. These intervals are: $I_1:[\lambda_0^*,1)$, $I_2:[0,\lambda_0^*)$, $I_3:[-\lambda_0^*,0)$, $I_4:(-1,-\lambda_0^*)$, where λ_0^* is an optimal probability threshold based on the risk decision rule of ERM, which will be represented in the next subsection. We calculate the number of positive samples and negative samples in different intervals, respectively. These numbers are denoted by n_{ij}, where $i=1$ represents positive class, $i=2$ represents negative class and $j=1,2,3,4$ represent the four intervals.

For samples in the nonseparable domain, we use the following rule for classification:

$$y=\begin{cases} 1 & if\ (f(x_0)\in I_j)\,\&\,(n_{1j}\ge n_{2j}) \\ -1 & if\ (f(x_0)\in I_j)\,\&\,(n_{1j}\ge n_{2j}) \end{cases},\ j=1,2,3,4. \tag{7}$$

B. The Optimization Probability Threshold

Optimization probability threshold λ_0^* is a critical parameter in the decision rule. For any given optimization probability threshold λ_0^*, there exist two types of errors. Similar to statistical process control, we call them type I error and type II error. Type I error means the negative sample is wrongly classified to positive class, while type II error means the positive sample is wrongly classified to negative class. As presented in [17],

The probability of type I error is: $\lambda_1 \cdot \Pr(\lambda_1 < \lambda_0)$,

The probability of type II error is: $(1-\lambda_1) \cdot \Pr(\lambda_1 \geq \lambda_0)$.

Denote by $F(\cdot)$ the distribution function and $f(\cdot)$ the density function of variable λ_1, the expectation of the two types of errors is as following:

$$E_1(\lambda_0) = F(\lambda_0) \cdot \int_0^{\lambda_0} \lambda_1 f(\lambda_1) d\lambda_1 , \quad (8)$$

$$E_2(\lambda_0) = (1-F(\lambda_0)) \cdot \int_{\lambda_0}^{1} (1-\lambda_1) f(\lambda_1) d\lambda_1 . \quad (9)$$

Assume that variable λ_1 submits to uniform distribution on interval $[0,1]$. The distribution function $F(x)$ and density function $f(x)$ are

$$F(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x < 1 , \\ 1, & x \geq 1 \end{cases} \quad (10)$$

$$f(x) = \begin{cases} 1, & 0 < x < 1 \\ 0, & otherwise \end{cases} . \quad (11)$$

The risk decision rule of ERM minimizes the expectation of the two types of errors by

$$\min_{\lambda_0} E(\lambda_0) = F(\lambda_0) \cdot \int_0^{\lambda_0} \lambda_1 f(\lambda_1) d\lambda_1$$

$$+ (1-F(\lambda_0)) \cdot \int_{\lambda_0}^{1} (1-\lambda_1) f(\lambda_1) d\lambda_1 \quad (12)$$

s.t. $0 \leq \lambda_0 < 1$.

When λ_1 submits to uniform distribution, the optimization model is equivalent to

$$\min_{\lambda_0} E(\lambda_0) = 3(\lambda_0)^2 - 3\lambda_0 + 1 . \quad (13)$$

Solving problem (12) would result in $\lambda_0^* = 0.5$ and $E^*(\lambda_0^*) = 0.25$.

C. The Applicability of RD-SVM

RD-SVM mainly focuses on samples in the nonseparable domain; therefore, its applicability should be investigated. We define percentage of samples belonging to each class in different intervals as follows

$$p_{ij} = n_{ij} \Big/ \sum_{k=1}^{4} n_{ik} , \quad (14)$$

where $i = 1,2$ represents the two classes, $j = 1,2,3,4$ represent the four intervals, and n_{ij} represents the number of class samples belonging to class i in the jth interval. Define

$p_1(j^*) = \max_j \{p_{1j}\}$, $j = 1,2,3,4$;

$p_2(k^*) = \max_k \{p_{2k}\}$, $k = 1,2,3,4$,

where $p_i(j^*)$ represents the maximal frequency of samples belonging to class i that fall in the j^*th interval. There are three possible situations in classification.

If samples distribute in four intervals equally, it means that two classes are not easy to separate;

If $p_1(j^*) >= 70\%$, $p_2(k^*) >= 70\%$ and $j^* \neq k^*$, it means that two classes are easy to separate;

If $p_1(j) > p_2(j)$ or $p_1(j) < p_2(j)$, it means that every interval contains mostly one class samples.

In the condition that two classes are easy to separate and every interval contains mostly one class, the performance of RD-SVM would be better than conventional SVM, which is demonstrator in computational results presented in the next section.

IV. Experiments

In order to investigate the validity and applicability of RD-SVM proposed in this paper, three benchmark data sets [18] are tested on RD-SVM and SVM, respectively. We assume that probability λ_1 of a sample belonging to different classes submits to uniform distribution on interval $[0,1]$. The optimal probability threshold $\lambda_0^* = 0.5$ is computed based on risk decision rule of ERM. In RD-SVM and SVM, we use Radial Basis Function (RBF) as the kernel function with parameter $\gamma = 1$ and $C = 0.1$ for good generalized classification ability. The experiments are conducted using MATLAB 7.1. The main program of RD-SVM in MATLAB 7.1 is following as

```
{ZERO=0.00001;K=3;gamma=1;r=1;d=2;C=0.1;
aver_error=0;averageerr=zeros(1,100);
averagn1_1=zeros(1,100);averagn1_2=zeros(1,100);
averagn2_1=zeros(1,100);averagn2_2=zeros(1,100);
averagn3_1=zeros(1,100);averagn3_2=zeros(1,100);
averagn4_1=zeros(1,100);averagn4_2=zeros(1,100);
 for n=1:100
   str1='data/thyroid\thyroid_train_data_';
    str1=strcat(str1,num2str(n));
    str1=strcat(str1,'.asc');
    inFile=str1;TrainData=load(inFile);
    [xN,xV]=size(TrainData);
    for i=1:xN
        xSVM(i,:)=TrainData(i,:);
    end
   str2='data/thyroid\thyroid_train_labels_';
    str2=strcat(str2,num2str(n));
    str2=strcat(str2,'.asc');
    inFile=str2;TrainLabels=load(inFile);
    [yN,yV]=size(TrainLabels);
    for i=1:yN
        ySVM(i)=TrainLabels(i,1);
    end
   str3='data/thyroid\thyroid_test_data_';
    str3=strcat(str3,num2str(n));
    str3=strcat(str3,'.asc');
    inFile=str3;TestData=load(inFile);
    [vxN,vxV]=size(TestData);
    for i=1:vxN
        vxSVM(i,:)=TestData(i,:);
    end
   str4='data/thyroid\thyroid_test_labels_';
    str4=strcat(str4,num2str(n));
    str4=strcat(str4,'.asc');
    inFile=str4;TestLabels=load(inFile);
```

```
    [vyN,vyV]=size(TestLabels);
    for i=1:vyN
        vySVM(i)=TestLabels(i,1);
    end
[yesno,Alpha,Epsilon,Omiga,b] =
CommonSVM(K,xSVM',ySVM,C,gamma,r,d);
    if (yesno==0)
        disp('There are no solutions to QP!');
        return;
    end
[MaxOBJ,SemiMinDist,NormOmiga,HisDist] =
CalHisDist(K,xSVM',ySVM,C,gamma,r,d,Alpha,Epsilo
n,b);
ValDis=CalValDis(vxSVM,xSVM,ySVM,NormOmiga,Alpha,
gamma,b);
Middle=[];
[N,V]=size(HisDist);
RelativeHisDist=zeros(1,V);
    for iter=1:V
RelativeHisDist(iter)=HisDist(iter)*NormOmiga;
        if
(RelativeHisDist(iter)<1)&(RelativeHisDist(iter)
>-1)
Middle=[Middle;[RelativeHisDist(iter)
ySVM(iter)]];
        else
        end
    end
lambda=solvelambda(Middle);
 [valN,valV]=size(Middle);
    for j=1:valN
        if
(Middle(j,1)>=lambda)&(Middle(j,1)<1)&(Middle(j,
2)==1)
            averagn1_1(n)=averagn1_1(n)+1;
        else if
(Middle(j,1)>=lambda)&(Middle(j,1)<1)&(Middle(j,
2)==-1)
                averagn1_2(n)=averagn1_2(n)+1;
            else if
(Middle(j,1)>=0)&(Middle(j,1)<lambda)&(Middle(j,
2)==1)

averagn2_1(n)=averagn2_1(n)+1;
                else if
(Middle(j,1)>=0)&(Middle(j,1)<lambda)&(Middle(j,
2)==-1)
averagn2_2(n)=averagn2_2(n)+1;
                    else  if
(Middle(j,1)>=lambda-
1)&(Middle(j,1)<0)&(Middle(j,2)==1)

averagn3_1(n)=averagn3_1(n)+1;
                        else if
(Middle(j,1)>=lambda-
1)&(Middle(j,1)<0)&(Middle(j,2)==-1)
averagn3_2(n)=averagn3_2(n)+1;
                            else if
(Middle(j,1)>-1)&(Middle(j,1)<lambda-
1)&(Middle(j,2)==1)

averagn4_1(n)=averagn4_1(n)+1;
                                else if
(Middle(j,1)>-1)&(Middle(j,1)<lambda-
1)&(Middle(j,2)==-1)
averagn4_2(n)=averagn4_2(n)+1;
                                    else
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
RalativeValDis=zeros(1,vxN);
    for i=1:vxN
        RelativeValDis(i)=ValDis(i)*NormOmiga;
        if RelativeValDis(i)>=1
            xySVM(i)=1;
        else
        end
        if RelativeValDis(i)<=-1
            xySVM(i)=-1;
        else
        end
        if
RelativeValDis(i)>=lambda&RelativeValDis(i)<1
            if averagn1_1(n)>=averagn1_2(n)
                xySVM(i)=1;
            else
                xySVM(i)=-1;
            end
        end
        if
RelativeValDis(i)>=0&RelativeValDis(i)<lambda
            if averagn2_1(n)>=averagn2_2(n)
                xySVM(i)=1;
            else
                xySVM(i)=-1;
            end
        end
        if RelativeValDis(i)>=lambda-
1&RelativeValDis(i)<0
            if averagn3_1(n)>=averagn3_2(n)
                xySVM(i)=1;
            else
                xySVM(i)=-1;
            end
        end
        if ValDis(i)>-1&ValDis(i)<lambda-1
            if averagn4_1(n)>=averagn4_2(n)
                xySVM(i)=1;
            else
                xySVM(i)=-1;
            end
        end
    end
    error_no=0;
    for i=1:vxN
        if(xySVM(i)~=vySVM(i))
            error_no=error_no+1;
        else
        end
    end
    aver_error=error_no/vxN;
    averageerr(n)=aver_error;
end
sum=0;
average=0;
for m=1:n
    sum=sum+averageerr(m);
end
average=sum/n;disp('The average error rate is ');
average}
```

Firstly, we compute the average number of positive and negative class in different intervals introduced in RD-SVM. The results are summarized in Table 1 and Table 2.

TABLE I.
DIFFERENT CLASSES FOR TRAINING SAMPLES

	An_{11}	An_{21}	An_{12}	An_{22}	An_{13}	An_{23}	An_{14}	An_{24}
Thyroid	0	0	19.9	0	22.7	28.8	0.2	68.5
Titanic	5.4	0.1	12.9	3.0	9.2	11.4	19.3	81.0
Twonorm	116.9	0	8.0	107.1	77.2	8.0	0	82.8

TABLE II.
DIFFERENT CLASSES FOR TESTING SAMPLES

	An_{11}	An_{21}	An_{12}	An_{22}	An_{13}	An_{23}	An_{14}	An_{24}
Thyroid	0	0	4.8	0.03	17.4	19.5	0.17	33.1
Titanic	56.6	2.1	164.7	53.6	132.2	167.5	284.9	1089
Twonorm	104.5	105.5	1985	1866	1376	1493	35.2	34.8

In Tables 1 and 2, An_{ij} represents average number of samples belonging to class i in the jth interval. The results show that train and test samples are correlative. For Thyroid set and Titanic set, two classes are easy to separate, while for Twonorm set, two classes are not easy to separate.

RD-SVM and SVM are tested on the three benchmark data sets and the average misclassification rates are summarized in Table 3.

TABLE III.
THE AVERAGE MISCLASSIFICATION RATE

	MR_{RD-SVM} (%)	MR_{SVM} (%)	IR(%)	Paired 2-tailed t test ($\alpha = 0.05$)
Thyroid	5.36	23.45	77.14	0.000<0.05
Titanic	23.17	24.24	4.41	0.000<0.05
Twonorm	48.14	48.32	0.37	0.000<0.05

In Table 3, MR is the misclassification rate and IR is the improved rate, and $IR(\%)=\frac{MR_{SVM}-MR_{RD\text{-}SVM}}{MR_{SVM}}\times 100\%$.

The results in Table 3 show that the RD-SVM can improve classification performance of conventional SVM remarkably when two classes are easy to separate. When the two classes are not easy to separate, the performance of RD-SVM is comparable to conventional SVM.

V. CONCLUSION

Conventional SVM classifies samples using optimal separating hyper-plane (OSH) obtained by support hyper-planes. For the nonseparable domain in risk management, SVM is sensitive to samples near the OSH, which would result in the increase of misclassification. In this paper, a new classification rule is proposed which modifies the outputs of SVM aiming at reducing the misclassification of samples in nonseparable domain using MATLAB. Computational results show that the proposed approach is better than conventional SVM remarkably when two classes are easy to separate, and in other condition, its performance is comparable to conventional SVM.

Our study has the following limitations that need further research. First, the parameters selection in the SVM for best prediction performance is also needed in the new RD-SVM technique. The second issue for future research relates to a method of estimating the distribution of the probability of samples belonging to different classes.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China(71071003), the MOE Project of Youth Foundation of Humanities and Social Science（09YJC630004）, and the Natural Science Foundation key project of Anhui University (KJ2009A59)

REFERENCES

[1] V.N. Vapnik, The nature of statistical learning theory, Springer, 1995.

[2] J.H. Min, and Y.C. Lee, Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters, Expert Systems with Applications, 2005, vol. 28, pp. 603-614.

[3] E. Avci, Selecting of the optimal feature subset and kernel parameters in digital modulation classification by using hybrid genetic algorithm-support vector machines, Expert Systems with Applications, 2007, vol. 36, no. 2, pp. 1391-1402..

[4] C.W. Hsu, and C.J. Lin, A comparison of methods for multiclass support vector machines, IEEE Transactions on Neural Networks, 2002, vol. 2, pp. 415-425.

[5] A. Navia-Vazquez, Compact multi-class support vector machine, Neurocoputing, 2007, vol.71, pp. 400-405.

[6] X.D. Wang, W.F. Liang, and X.S. Cai, Application of adaptive least square support vector machines in nonlinear system identification, Proceeding of the 6th World Congress on Intelligent Control and Automation, 2006, pp. 1897-1900.

[7] S.X. Lu, and X.Z. Wang, A comparison among four SVM classification methods: LSVM, NLSVM, SSVM and NSVM, Proceeding of the Third International Conference on Machine Learning and Cybernetics, 2004, pp. 4277-4282.

[8] H.B. Liu, and S.W. Xiong, Fuzzy support vector machines based on density clustering, IEEE International Conference on Control and Automation, 2007, pp. 784-787.

[9] Z.S. Hua, X.M. Zhang, and X.Y. Xu, Asymmetric support vector machine for the classification problem with asymmetric cost of misclassification, International Journal of Innovative Computing, Information and Control, 2011, vol. 6, no. 12, pp. 5597-5608.

[10] A.S. Ghamdi, Using logistic regression to estimate the influence of accident, Accid Anal Prev, 2002, vol. 34, pp. 729-741.

[11] C.E. Lemke, Bimarix equilibrium points and mathematical programming, Management Science, 1965, vol. 11, pp. 681-698.

[12] B. Zhang, and Z.S. Hua, An improved Lemke algorithm for convex quadratic programming with equality constrains, Journal of University of Science and Technology of China, 2004, vol. 6, pp. 668-677.

[13] Z.S. Hua, B. Zhang, J. Yang, and D.S. Tan, A new approach of forecasting intermittent demand for spare parts inventories in the process industries, Journal of the Operational Research Society, 2007, vol. 58, pp. 52-61.

[14] A. Madevska-Bogdanova, D. Nikolik, and L. Curfs, Probabilistic SVM outputs for pattern recognition using analytical geometry, Neurocomputing, 2004, vol. 62, pp. 293-303.

[15] X. Xue, and G.P. He, New method for decision the sort of the test sample based on SVM decision tree, Computer Applications, 2007, vol. 1, pp. 84-85.

[16] Z.S. Hua, Y. Wang, X.Y. Xu, B. Zhang, and L. Liang, Predicting corporate financial distress based on integration of support vector machines and logistic regression, Expert Systems with Application, 2007, vol. 33, pp. 434-440.

[17] Z.S. Hua, S.J. Li, and Z. Tao, A rule-based risk decision-making approach and its application in China's customs inspection decision, Journal of the Operational Research Society, 2006, vol. 57, pp. 1313-1322.

[18] http://ida.first.fhg.de/projects/bench/benchmarks.htm.

Dr. Xuemei Zhang is a Lecturer in School of Economics and Business, Fuyang Teachers College, Fuyang, Anhui, China.
Her major field of study includes supply chain management, operation research, product design, and risk decision
(E-mail: xmz@mail.ustc.edu.cn).