Proceedings of the 2012 International Conference on Machine Learning and Cybernetics, Xian, 15-17 July, 2012

IMBALANCED EXTREME SUPPORT VECTOR MACHINE

XU ZHOU, SHU-XIA LU, LI-SHA HU, MENG ZHANG

Key Laboratory of Machine Learning and Computational Intelligence, College of Mathematics and Computer Science
Hebei University, Baoding 071002, China
E-MAIL:zhouxu19860901@sina.com, cmclusx@126.com, 295103689@163.com, zhangmeng1987.good@163.com

Abstract:

For the problem of imbalanced data classification which was not discussed in the standard Extreme Support Vector Machines (ESVM), an imbalanced extreme support vector machines (IESVM) was proposed. Firstly, a preliminary normal vector of separating hyperplane is obtained directly by geometric analysis. Secondly, penalty factors are obtained which are based on the information provided by data sets projecting onto the preliminary normal vector. Finally, the final separation hyperplane is got through the improved ESVM training. IESVM can overcome disadvantages of traditional designing methods which only consider the imbalance of samples size and can improve the generalization ability of ESVM. Experimental results show that the method can effectively enhance the classification performance on imbalanced data sets.

Key words:

Imbalanced data; Extreme support vector machine; projection

1. Introduction

Support vector machines [1-3] have been extensively used in widespread applications. It is used to find the separating hyperplane which maximizes the separating margins of two different classes in the feature space, while minimizing the training errors. As explained in [4], SVM can be seen as a specific type of SLFNs, the so called support vector networks. And Liu [5] proposed a new algorithm called Extreme Support Vector machine (ESVM), showing that the ELM [6-7] learning approach can be applied to SVMs directly by simply replacing SVM kernels with random ELM kernels and will achieve better generalization. It is well known that in ESVM classifier the training samples are explicitly mapped into a feature space by a single hidden layer feed forward network (SLFN) with its input weights randomly generated. In theory, this formulation provides better generalization performance and leads to an extremely simple and fast algorithm than traditional nonlinear SVM. One of the most advantages is that the algorithm avoids solving quadratic programming problem, greatly reduces the computational complexity, and it is very fast compared with PSVM [8] and LSSVM [9] in larger data sets. But the standard ESVM did not consider the distribution of positive and negative samples, and give the same penalty factors to all the samples. As we know most traditional learning systems assume that the class distribution in data sets is balanced, however, there are many real-world applications where the data sets are highly imbalanced and the assumption is often violated. In data mining, it is the rare event that is of interest and the cost of misclassifying the rare event is higher than misclassifying the usual event. When the data is highly skewed toward the usual, it can be very difficult for a learning system to accurately detect the rare event. Mining imbalanced data sets has been the focus of much research recently [10-15], and two important directions are researching the imbalanced data influence on traditional classification algorithms. That is reconstructing the training set or directly to improve traditional algorithm to adopt to the imbalance data. And imbalance data considered in this paper mainly has three cases: (1) the differences between the quantity of the two-class data are great; (2) two-class data have the similar quantity of samples, but with different distribution, one class is more concentrated, another is more dispersive; (3) The distribution and the quantity of the two-class data are very different. Based on this focus, a new method was proposed called Imbalanced Extreme Support Vector Machine (IESVM), which adds different penalty factors before the two-class slack variables to control the importance of each class samples. Indeed, experiments have showed IESVM obtains better precision compared with ESVM, and can be better generalization performance.

2. Extreme support vector machine

The linear Extreme Support Vector Machine algorithm has the same form as the linear PSVM, and they have the same process to get their solutions, so we do not present here. Now we introduce the nonlinear ESVM classifier, the

978-1-4673-1487-9/12/$31.00 ©2012 IEEE

formulation of the nonlinear ESVM is the following quadric program problem with a parameter C which is a specified parameter and provides a trade off between the distance of the separating margin and the training errors.

$$\min_{(w,b,\xi_i)} \frac{1}{2}(w^T w + b^2) + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \qquad (1)$$
$$s.t. \quad y_i(w^T\phi(x_i) - b) = 1 - \xi_i$$

where $\phi(x): R^m \to R^N$ is a map function, which can map the samples in the input space into a high-dimension space. To the best of our knowledge almost all previous nonlinear SVM algorithms make use of a kernel function $K(x',x)$, ($K(x',x) = \phi(x)\bullet\phi(x)$). But we usually don't know the map function ϕ and many of its properties in the nonlinear SVM algorithms. However the ϕ in the ESVM can be constructed by the hidden layer of a random SLFN just as in the article [5] and its properties are obviously known for us. The map function can be obtained as follows:

$$\phi(x) = G(XM) = (g(XM_1), \cdots, g(XM_N)) \qquad (2)$$

where N, n and m are the number of the hidden layers, training samples and the features of samples, respectively. $x \in R^m$ is the input vector and $X_i = [x,1]$. $M = (M_1, \cdots, M_N) \in R^{(m+1)\times N}$ is a matrix whose elements are generated randomly, and the $g(\bullet)$ is the nonlinear active function which can map a matrix Z with elements z_{ij} to another matrix which has the same size with elements $g(z_{ij})$. ξ_i is the slack variable for the training sample x_i.

We can get the normal vector w and the bias b of the separation hyperplane by using the method of Lagrangian for the quadric program problem (1).

Now for an arbitrary unseen sample x, firstly the sample is mapped into the feature space by ϕ, then the nonlinear ESVM classifier works as follow:

$$\phi(x)^T w - b = \begin{cases} >0, \text{then} & x \in \text{positive class} \\ <0, \text{then} & x \in \text{negtive class} \\ =0, \text{then} & x \in \text{positive class or } x \in \text{negtive class} \end{cases} \qquad (3)$$

3. Imbalanced extreme support vector machine

3.1. Determine the preliminary normal vector

In this paper, we only consider a binary classification problem—positive class and negative class. The concept of clustering center is introduced first. Suppose the distribution of the binary classification data is independent in each dimension space.

For the linear separable problem, given the training samples $\{x_1, x_2, \cdots x_n\}$, y_i is the label of x_i, $y_i \in \{1,-1\}$, $i = 1, \cdots, n$. Let the numbers of positive and negative class be n^+ and n^-, so $n = n^+ + n^-$. The positive clustering center and negative clustering center are listed as follows:

$$X_+ = \frac{1}{n^+}\sum_{y_i=1} x_i, \quad X_- = \frac{1}{n^-}\sum_{y_j=1} x_j \qquad (4)$$

For the nonlinear separable problem, we first map the given training samples in the input space into $\{\phi(x_1), \phi(x_2), \cdots, \phi(x_n)\}$ in a high-dimensional feature space by using a nonlinear map function ϕ. The positive and negative clustering centers are presented respectively as follows:

$$X_+ = \frac{1}{n^+}\sum_{y_i=1} \phi(x_i), \quad X_- = \frac{1}{n^-}\sum_{y_i=-1} \phi(x_j) \qquad (5)$$

Actually, ESVM is fitting two parallel hyperplanes by using the positive samples and the negative samples. Moreover it should guarantee to make each of the two-class samples have the minimum sum error. Then according to the empirical risk minimization principle the two hyperplanes will pass through the positive and negative clustering center, respectively. And the parallel hyperplanes in the ESVM will maximize the separating margin of the two-classes by the structural risk minimization principle. Then according to the Geometric Axiom, the maximum distance between two parallel hyperplanes, which pass through two arbitrary points, is the distance of these two points. So the normal vector of the hyperplane is the direction of a line passing through the two points. The preliminary normal vector of the optimal hyperplane can be determined as follows:

$$w_0 = \frac{1}{n^+}\sum_{y_i=1}\phi(x_i) - \frac{1}{n^-}\sum_{y_j=-1}\phi(x_j) \qquad (6)$$

3.2. Theory analysis and determine the penalty factors

It is well known that the constraint condition in ESVM is $y_i(w^T\phi(x_i) - b) = 1 - \xi_i$ instead of the inequality $y_i(w^T\phi(x_i) - b) \geq 1 - \xi_i$ in SVM. ESVM with a linear kernel or a nonlinear kernel tries to find the proximal planes $w^T x - b = \pm 1$ in the input space or $w^T\phi(x) - b = \pm 1$ in the feature space. The standard ESVM algorithm gives the same penalty factor to the slack variable factor for each sample. However, the algorithm doesn't provide satisfactory performance for the problems on imbalanced data that are often encountered in real-world applications. So the final separation hyperplane will be out-of-balance and the classification performance of ESVM algorithm will fall sharply inevitably when ESVM used on imbalanced data. According to the theory of probability knowledge in [16-17], the final separating hyperplane will move to the class with fewer samples if the two-class training samples are similar in dispersion degree but largely different in quantity. However, the final separating hyperplane will move to the class with higher dispersion degree if the two-class training samples are roughly equal in quantity but greatly different in dispersion degree. In order to deal with this problem, we introduce unequal penalty factors γ^+ and γ^- for the positive and negative slack variables. We increase the penalty factors of the class which has fewer samples and is more easily mistaken, moreover, reduce the penalty factors of the class which has larger samples and is not easily mistaken, so we know $\frac{\gamma^+}{\gamma^-} \infty \frac{n^-}{n^+}$.

Method 1: Just consider the size of the two-class training samples. we order $\gamma^+ = C$, $\gamma^- = C\frac{n^+}{n^-}$ and C is a constant.

(Note: this is a common method using penalty factors to deal with imbalanced data in many classification algorithms except ESVM. This paper quotes from this method to construct IESVM1 which is compared with ESVM and method 2 (IESVM2) in the nest.)

Method 2: Consider both the differences in size and dispersion degree on datasets. Here we mainly consider the dispersion degree along with the normal vector of separating hyperplane. Therefore, we can use the variance of the two-class data projected on the normal vector to represent the dispersion degree. Firstly, the given training samples in the input space are mapped into a high dimension space. Secondly, one-dimensional data was generated in (7) by projecting the high dimensional data onto the preliminary normal vector. Finally, by using the information provided by the standard deviation of the one-dimensional data and the difference of two-class sample sizes, the proportion of the two-class penalty factors is determined.

$$\begin{aligned} w_0\phi(x_l) &= \left[\frac{1}{n^+}\sum_{y_i=1}\phi(x_i) - \frac{1}{n^-}\sum_{y_j=-1}\phi(x_j)\right]\phi(x_l) \\ &= \frac{1}{n^+}\sum_{y_i=1}k(x_i,x_l) - \frac{1}{n^-}\sum_{y_j=-1}k(x_j,x_l) \quad l=1,\cdots,n \end{aligned} \tag{7}$$

The positive standard deviation S^+ and negative standard deviation S^- are calculated, respectively.

$$S^+ = \sqrt{\frac{1}{n^+-1}\sum_{y_l=1}\left\{\begin{aligned}&\left[\frac{1}{n^+}\sum_{y_i=1}k(x_i,x_l) - \frac{1}{n^-}\sum_{y_j=-1}k(x_j,x_l)\right] - \\ &\frac{1}{n^+}\sum_{y_l=1}\left[\frac{1}{n^+}\sum_{y_i=1}k(x_i,x_l) - \frac{1}{n^-}\sum_{y_j=-1}k(x_j,x_l)\right]\end{aligned}\right\}^2} \tag{8}$$

$$S^- = \sqrt{\frac{1}{n^--1}\sum_{y_l=-1}\left\{\begin{aligned}&\left[\frac{1}{n^+}\sum_{y_i=1}k(x_i,x_l) - \frac{1}{n^-}\sum_{y_j=-1}k(x_j,x_l)\right] - \\ &\frac{1}{n^-}\sum_{y_l=-1}\left[\frac{1}{n^+}\sum_{y_i=1}k(x_i,x_l) - \frac{1}{n^-}\sum_{y_j=-1}k(x_j,x_l)\right]\end{aligned}\right\}^2} \tag{9}$$

We order $\gamma^+ = \gamma\frac{s^+}{n^+}$, $\gamma^- = \gamma\frac{s^-}{n^-}$, γ is a specific constant, because we know $\frac{\gamma^+}{\gamma^-} \infty \frac{n^-}{n^+}$ and $\frac{\gamma^+}{\gamma^-} \infty \frac{S^+}{S^-}$ from the above analysis above. The algorithm constructed using these penalty factors is called IESVM2 in this paper.

3.3. Establishment of the model

Here we mainly consider the two-class classification problem of classifying n training samples in m -dimensional real space R^m represented by the $n\times(m+1)$ matrix A which will be explained later. And the model of Imbalance Extreme Support Vector machine can be presented as follows:

$$\begin{aligned} \min_{(w,b,\xi)} \quad & \frac{1}{2}(w^T w + b^2) + \frac{1}{2}\xi^T W\xi \\ s.t. \quad & D(w^T\phi(A) - b) + \xi = e \end{aligned} \tag{10}$$

Where D is a diagonal matrix with the elements +1 or -1 which are determined by the class A_+ or A_- of each sample A_i. And $A = [X_1,\cdots,X_n]^T \in R^{n\times(m+1)}$ is the matrix of input vector X_i which represented each training sample plus the bias of one for the last term just as $X_i = [x_i, 1]$, $i = 1,2,\cdots,n$ .and $\xi = [\xi_1,\cdots,\xi_n]^T_{n\times 1}$ is the slack vector. A diagonal matrix $W = [\gamma_1,\cdots,\gamma_n]^T \in R^{n\times n}$ is with elements γ_i.

$\gamma_i = \gamma^+$ if $D_{ii} = 1$ otherwise $\gamma_i = \gamma^-$. The lagrangian for (10) can be written as follows:

$$L(w,b,\xi,\alpha)=\frac{1}{2}(w^T w+b^2)+\frac{W}{2}\|\xi\|^2-\alpha\left(D(w^T \bullet \phi(A)-b)+\xi-e\right) \quad (11)$$

Here α is the lagrangian multiplier with the equality constraints. In order to find the optimal solutions of (10), according to Karush-Kuhn-Tucher (KKT) optimality conditions we have: $\frac{\partial L}{\partial w}=0, \frac{\partial L}{\partial b}=0, \frac{\partial L}{\partial \xi}=0, \frac{\partial L}{\partial \alpha}=0,$ and we can further get:

$$w=\phi(A)'D\alpha \quad (12)$$

$$b=-e'D\alpha \quad (13)$$

$$\alpha = W\xi \quad (14)$$

$$D(\phi(A)w-eb)+\xi-e=0 \quad (15)$$

Substituting the three expressions (12), (13) and (14) in the (15), we can get an explicit expression for $D\alpha$:

$$D\alpha=(\phi(A)\phi(A)^T+ee^T+W^{-1})^{-1}De=(E_\phi E_\phi^T+W^{-1})^{-1}De \quad (16)$$

$E_\phi = [\phi(A),-e] \in R^{n\times(N+1)}$. It obviously can be seen that the formula (16) of $D\alpha$ still entails the inversion of a possibly massive matrix of order $n\times n$. We can make use of the Sherman-Morrison-Woodbury (SMW) formula to overcome this problem, so we get the results as follows:

$$D\alpha=(W-WE_\phi(I+E_\phi^T WE_\phi)^{-1}E_\phi^T W)De \quad (17)$$

Where I is the unit matrix with the dimension of $(N+1)\times(N+1)$, so we can get the solution of $D\alpha$ just involves the inversion of a matrix of order $(N+1)\times(N+1)$. Usually N can be much smaller than n and is independent of n which is the number of the training samples. Then, it is easy to get the normal vector w and the bias b, if we substitute the expression (17) in the (12) and (13).

$$\begin{bmatrix} w \\ b \end{bmatrix}=E_\phi^T(W-WE_\phi(I+E_\phi^T WE_\phi)^{-1}E_\phi^T W)De \quad (18)$$

Now for an unseen sample x, which is to map into the feature space by ϕ. The nonlinear IESVM classifier works as follows:

$$\phi(x)^T w-b=\begin{cases} >0, & then \quad x\in A_+ \\ <0, & then \quad x\in A_- \\ =0, & then \quad x\in A_+ \ or\ x\in A_- \end{cases} \quad (19)$$

Algorithm1: The Nonlinear Imbalanced Extreme Support Vector Machine Classifier

Input: Training set: $x_i \in R^m$, $y_i \in \{1,-1\}$ $i=1,\cdots,n$

Output: IESVM algorithm: $H(x)$.

Step 1: Generate a matrix $M \in R^{(m+1)\times N}$ randomly, and choose an appropriate activation function g, typically the sigmoid function, to construct ϕ.

Step 2: Map the training sample x_i into the feature space by ϕ to get $\phi(x_i)$.

Step 3: Calculate the positive clustering center X_+ and the positive clustering center X_-, then the preliminary normal vector w_0 can be get by (6).

Step 4: Project all the training samples onto the normal vector w_0 to get the one-dimension data.

Step 5: Calculate the two-class's standard deviation S^+ and S^-.

Step 6: Determine the penalty factors γ^+ and γ^- by the standard deviation and the sizes of the two-class samples.

Step 7: Construct the optimization problem (10).

Step 8: Get the final normal vector w and the bias b by solving the dual optimization problem of (10).

Output the final function:

$$H(x)=sign(\phi(x)^T w-b)$$

4. Experimental results

In this section, we compare the performance of the ESVM, IESVM1, IESVM2, ELM and SVM, which is implemented by LIBSVM with the Gaussian kernel $\exp(-\beta\|u-v\|^2)$. All these algorithms are implemented in MATLAB7.1, 2.8 GHz, Pentium 4 CPU, 512 MB memory. Firstly, three artificial data sets are generated to test the performance of ESVM, IESVM1and IESVM2 algorithm, respectively. The artificial data in the three Figures are generated randomly and described in table 1:

TABLE 1. BASIC INFORMATION OF THE ARTIFICIAL DATA SETS

Data description		NO. of data	mean value	variance
Figure 1	*Positive data*	10	0	1
	Negative data	100	5	2
Figure2	*Positive data*	100	0	1
	Negative data	100	5	2
Figure3	*Positive data*	10	0	2
	Negative data	100	5	2

(Note: the red line represents the hyperplane IESVM2, the blue line represents hyperpane IESVM1 and the green line represents hyperplane ESVM in the following three pictures)

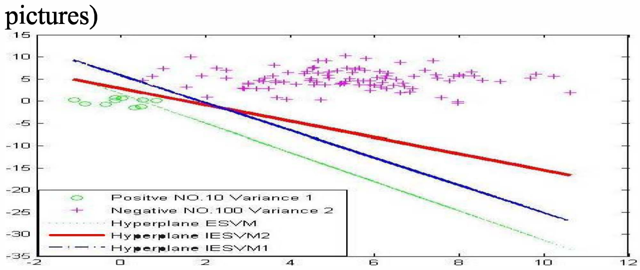

Figure 1. Different classifiers comparison on two-class data with different NO. and variance

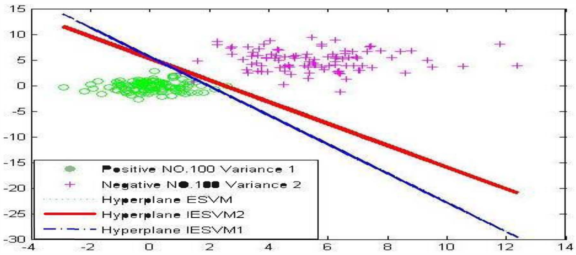

Figure 2. Different classifiers comparison on two-class data with the same NO. but different variance

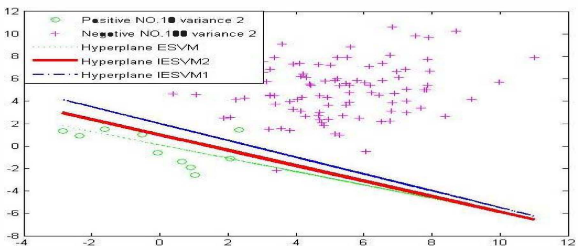

Figure 3. Different classifier comparison on two-class data with the same variance but different NO.

TABLE 2 BASIC INFORMATION OF THE DATA SETS

Dataset	# P_NO./N_NO.	#Attributes	#Classes
pima	186/351	8	2
Diabetes	263/505	8	2
thyroid	65/ 150	5	2
Heart	120/150	13	2
Ionosphere	126/225	34	2
German	300/700	20	2
cancer	237/446	7	2
ecoli	86/150	8	5
Iris	50/100	4	3

The experimental results show that the hyperplane ESVM moves to the class with less samples on the imbalanced data sets. So it is inevitable to mistake some samples in the class with fewer samples. These results are not what we want; however, the performance of IESVM1 and IESVM2 is less influenced by the imbalanced data sets. Thus IESVM1 and IESVM2 can effectively guarantee the classification results. Figure 2 shows that hyperplane ESVM and hyperplane IESVM1 are overlap what means IESVM1 algorithm doesn't work, when the two-class samples are equal in number, but different in scattered degrees. However IESVM2 algorithm still shows a good performance. Figure 3 shows that: when the samples have the same scattered degrees but are greatly different in numbers, the results of the classification of ESVM are bad, but at the moment IESVM1 and IESVM2 can still have a good classification effect.

To further test the performance of these algorithms, we run them on several publicly available benchmark datasets. In the simulations, we consider the sigmoid function: $g(x) = 1/(1 + \exp(-x))$ as active function. We set N to 100, and fix the parameter C and parameter γ to 100 for all these algorithms.

Table 3 presents the average learning results of these algorithms with 10-fold cross-validation on these benchmark datasets. It can be seen that IESVM1 and IESVM2 demonstrate better test accuracy than ESVM and ELM. And ELM has better training accuracy but lower test accuracy relatively. It also can be seen that IESVM2 take more learning CPU time than IESVM1. This is because we consider the scattered degree in the two-class data sets additionally. That is the reason why IESVM2 has much better test accuracy and stability than the other algorithms on imbalanced data sets. Considering from the whole table 3, we can observe that the IESVM2 can achieve comparable accuracy to LIBSVM most of the time, and the test accuracy and stability of IESVM2 are much better than ESVM, ELM and IESVM1 obviously.

TABLE 3. TRAINING AND TESTING ACCURACY AND TRAINING TIME OF THE FIVE ALGORITHMS

Data sets	ESVM Train Test time	IESVM1 Train Test time	IESVM2 Train Test time	ELM Train Test time	SVM Train Test time
	80.26	82.68	83.43	82.87	77.83
Pima	74.45	74.89	75.61	73.16	78.35
	0.0156	0.2500	34.375	0.0938	0.0312
	81.56	77.65	83.05	81.05	70.21
Diabetes	73.29	76.63	74.459	71.00	68.83
	0.0781	0.2656	70.313	0.0625	0.0938
	97.35	97.35	94.18	94.71	86.24
Heart	80.25	80.27	86.42	77.78	83.95
	0.0156	0.0625	15.437	0.0313	0.0156
	99.18	98.36	99.58	99.18	95.51
Ionosphere	86.79	87.76	90.57	85.19	92.45
	0.0156	0.0313	30.313	0.0313	0.0156
	92.67	95.33	93.31	97.33	98.67
Thyroid	93.84	93.85	95.38	89.23	93.84
	0.0156	0.0468	5.7968	0.0313	0.0156
	79.28	78.43	78.28	81.00	82.00
German	71.67	72.13	73.66	73.23	76.67
	0.1094	0.4531	183.61	0.1094	0.1563
Cancer	98.32	97.90	98.74	97.70	98.58

	94.63	95.12	95.12	94.63	95.12
	0.0469	0.1875	6.2031	0.0781	0.0469
Ecoli	90.64	91.06	91.06	95.74	88.94
	85.15	87.12	91.08	85.15	91.09
	0.0156	0.0781	23.078	0.0313	0.0156
Iris	100.00	100.00	100.00	100.00	100.00
	100.00	100.00	100.00	97.78	100.00
	0.0156	0.0156	2.5310	0.0313	0.0156

5. Conclusions

An improved ESVM is proposed to deal with imbalanced data sets in this paper. This new method uses different penalty factors to the positive and negative slack variables respectively in the optimal model, and can make the adjustment of penalty factors more flexible. What is more, when determining the penalty factors, we not only consider the quantity of the positive and negative samples but also take into account the different scattered degrees based on the samples in the projection of the normal vector between the positive and the negative data sets. The experiments show that IESVM2 has better stability and classification accuracy than ESVM, IESVM1 and ELM in the UCI data sets and artificial data set.

The characteristics of this new proposed algorithm in this paper are listed as follows:

(1) Determine the preliminary normal vector of the separation hyperplane directly by the clustering centers, which has obviously geometric meaning compared with the previously established normal vector, and it does not require initial training samples.

(2) We not only take into account the differences in the number of samples but also take into account the scattered degree of the samples.

(3) Without modifying the original samples, and make use of the information provided by the data set.

The next step is parameter selection and structure selection to get much better classification performance and accuracy.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China (61170040), the Natural Science Foundation of Hebei Province (F2011201063, 2010000323), and by the Plan of the Natural Science Foundation of Hebei University (doctor project) (Y2008122, Y2011-228).

References

[1] Li Zhang, and Weida Zhou, “Density-induced margin support vector machines”, Pattern Recognition, Volume 44, Issue: 7, Pages: 1448-1460, July 2011.

[2] Ertekin, S. Bottou, L. and Giles, C.L. “Nonconvex online support vector machines”, IEEE transactions on Pattern Analysis and Machine intelligence, Volume 33 Issue 2, Pages 368-381, February 2011

[3] Xiaoming Wang, Fulai Chung, and Shitong Wang. “Theoretical analysis for solution of support vector data description”, Neural Networks, Volume: 24, Issue: 4, Pages: 360-369, May 2011.

[4] Guangbin Huang, Xiaojian Ding and Hongming Zhou, “Optimization method based extreme learning machine for classification”, Neurocomputing. Volume: 74, Issue:1-3 Pages:155-163, May. 2010.

[5] Qiuge Liu, Qing He, and Zhongzhi Shi, “Extreme Support Vector Machine Classifier”, Proceedings of the 12th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining, pages: 222-233, 2008.

[6] Guangbin Huang, Qinyu Zhu, and Siew C. K., “Extreme learning machine: Theory and applications”, Neurocomputing, Volume: 70, Issue: 1-3, Pages: 489-501, 2006.

[7] Nan Liu, and Han Wang, “Ensemble based extreme learning machine”, IEEE Signal processing letters, Volume: 17, Issue: 8, Pages: 754-757, August 2011.

[8] P.Venkatesan, and G.Meena Devi, “Proximal support vector machine for disease classification”, International Journal of Science and Technology, Volume: 2, Issue: 1, Pages 61-64, January 2012.

[9] Yucheng Liu, and Yubin Liu, “Incremental learning method of least squares support vector machine”, Proceeding of ICICTA 2010 International Conference, pp. 529-532, May 2010.

[10] Batista. G. E. A. P. A, Prati. R. C, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” ACM SIGKDD Explorations Newsletter, Volume: 6, Issue: 1, Pages: 20–29, June 2004.

[11] Shoushan Li, Zhongqing Wang, Guodong Zhou and Sophia Yat Mei Lee. “Semi-Supervised Learing for Imbalanced Sentiment Classification”, Proceeding of the Twenty-Second International Joint Conference on Artificial Intelligence. Pages: 1826-1831, 2011.

[12] Tong Liu, Yongquan Liang, and Weijian Ni, “A learningstrategy for highly imblanced classification”, Proceedings of the third international conference on

internet multimedia computing and service, Pages: 116-119, 2011.

[13] Koknar-tezel. S., and Latecki. L.J., “Improving SVM classification on imbalanced data sets in distance spaces”, Proceeding of the Ninth IEEE International conference, Pages: 259-267, December 2009.

[14] Jae Pil Hwang, Seongkeun Park, and Euntai Kim. “A new weighted approach to imbalanced data classification problem via support vector machine with quadratic cost functions”, Volume: 38, Issue: 70, Pages 8580-8585, July 2011.

[15] Thai-Nghe, N., Gantner, Z., and Schmidt-thieme, L. “Cost-sensitive learning methods for imbalanced data”, Neural Networks, the 2010 internetional joint conference on, Pages: 1-8, July 2010.

[16] Zhenxia Xue, Sanyang Liu, and Wangli Liu, “Unbalanced least squares support vector machines”, Journal of system simulation, Volume: 21, Issue: 14, Pages: 4324-4327, 2009.

[17] Haibo He, and Edwardo A. Garcia, “Learning from imbalanced data”, IEEE transactions on knowledge and data engineering, Volume: 21, Issue: 9, September 2009.