Applied Mathematics & Information Sciences
An International Journal

The Upper Preferred Multiple Directed Acyclic Graph Support Vector Machines for Classification

Kan Li, Hang Xu, Wenxiong Huang, Zhonghua Huang

School of Computer, Beijing Institute of Technology, Beijing 100081 P. R. China

Received: 18 Sep. 2012; Revised 22 Nov. 2012; Accepted 29 Nov. 2012
Published online: 1 Mar. 2013

Abstract: The current classification algorithms have weak fault-tolerance. In order to solve the problem, a multiple support vector machines method, called Upper preferred Multiple Directed Acyclic Graph Support Vector Machines (UMDAG-SVMs), is proposed. Firstly, we present least squares projection twin support vector machine (LSPTSVM) with confidence-degree for generating binary classifiers. It uses the idea that "when the confidence-degree outputted from the node in the directed graph, is below the threshold, the decision-making process will go on along with the two branches of the node at the same time.", which strengthens the algorithm's fault-tolerance. In order to select the parameters of the algorithm, we use genetic algorithm to select these parameters. Secondly, according to the minimal hypersphere distance, and the known principle "the upper-level classifiers bring up better performance of classification in DAG-SVMs ", we present a new classification algorithm, called UMDAG-SVMs. This algorithm has two advantages of strong fault-tolerance and high classification accuracy. Finally, we make the experiments to test the performance of the algorithm. Experimental results in public datasets show that our UMDAG-SVMs has comparable classification accuracy to that other algorithms but with remarkable less computation.

Keywords: Support vector machine, classification, confidence-degree

1. Introduction

Directed acyclic graph support vector machines (short as DAG-SVMs) is derived from the Platt's proposed decision-oriented DAG, and is designed to deal with the false and refusing classification problems caused by "one-versus-one" in SVMs [1]. We have known the error accumulation will happen in the DAG-SVMs classification process. In other words, when classification errors occurred at a node, these errors will continue in its further down-level nodes. Meanwhile, if classification errors show up in the closer to the root node, the error accumulation will be more serious and the classification quality will be worse. In the paper, in order to avoid the upper-level nodes' leading wrong classification paths, we put forward least squares projection twin support vector machi-ne with confidence-degree. During decision-making process, this classifier can output classification labels as well as output the decision making confidence-degree. Having confidence-degree as a basis, the multiple classification process can go on simultaneously along the multiple graph paths when a node's confidence level which output from a directed acyclic graph is below a certain threshold. Support vector machine (SVM) in solving the small samples, nonlinearity, high dimension and local minimum problems shows many unique advantages, has become the current hot research, and was widely used in handwriting recognition, face recognition, speech recognition areas [2–5]. However, the current multi-class support vector machine algorithms have weak fault tolerance, which will obviously influence the support vector machine classification accuracy. Other algorithms such as nearest neighbor classification method [6], bayesian classification based SVM [7], latent SVM [8] were not well applied to classification with fault-tolerance. In order to solve the problem of weak fault tolerance, we present strong fault tolerance classification method – the upper preferred directed acyclic graph support vector machines.

[*] Corresponding author: e-mail: likan@bit.edu.cn

2. Least squares recursive projection twin support vector machine with the confidence-degree

2.1. *Brief introduction of LSPTSVM*

The objective function and the constraint condition of the least squares recursive projection twin support vector machine [9] are as follows

(LSPTSVM1)

$$\min_{w_1} \frac{1}{2}\sum_{i=1}^{m_1}(w_1^T x_i^{(1)} - w_1^T \frac{1}{m_1}\sum_{j=1}^{m_1} x_j^{(1)})^2 + \frac{c_1}{2}\sum_{k=1}^{m_2}\xi_k^2 + \frac{c_3}{2}\|w_1\|^2$$

$$s.t., w_1^T x_k^{(2)} - w_1^T \frac{1}{m_1}\sum_{j=1}^{m_1} x_j^{(1)} + \xi_k = 1, k = 1,2,\ldots,m_2$$

(LSPTSVM2)

$$\min_{w_2} \frac{1}{2}\sum_{i=1}^{m_2}(w_2^T x_i^{(2)} - w_2^T \frac{1}{m_2}\sum_{j=1}^{m_2} x_j^{(2)})^2 + \frac{c_2}{2}\sum_{k=1}^{m_1}\eta_k^2 + \frac{c_4}{2}\|w_2\|^2$$

$$s.t., w_2^T x_k^{(1)} - w_2^T \frac{1}{m_2}\sum_{j=1}^{m_2} x_j^{(2)} + \eta_k = 1, k = 1,2,\ldots,m_1, \tag{1}$$

where $c_1 > 0, c_2 > 0, c_3 > 0, c_4 > 0$ are the parameters.

There are extra two modifications. The first one is that in the objective functions of Eq.(1),the regularization terms $\frac{c_3}{2}\|w_1\|^2$ and $\frac{c_4}{2}\|w_2\|^2$ are introduced. It leads to be more theoretically sound than classical SVM [10] and twin support vector machine [11]. The second one is that the loss function in Eq.(1) is the square of 2-norm of slack variables ξ and η instead of 1-norm of ξ and η. It allows us to solve the dual quadratic programming problems (QPPs) by solving a simultaneous system of linear equations.

According to Eq.(1), the solving of the optimal projection axes w_1 and w_2 are given by

$$w_1 = (\frac{S_1}{c_1} + (-B + \frac{1}{m_1}e_2 e_1^T A)^T(-B + \frac{1}{m_1}e_2 e_1^T A) + \frac{c_3}{c_1}I)^{-1} \times (B - \frac{1}{m_1}e_2 e_1^T A)^T e_2$$

$$w_2 = -(\frac{S_2}{c_2} + (A - \frac{1}{m_2}e_1 e_2^T B)^T(S - \frac{1}{m_2}e_1 e_2^T B) + \frac{c_4}{c_2}I)^{-1} \times (A - \frac{1}{m_2}e_1 e_2^T B)^T e_1, \tag{2}$$

where I is an identity matrix of appropriate dimensions.

After the optimal projection axes are obtained according to Eq.(2), the training stage of LSPTSVM is completed. For testing, the label of a new coming data point x is determined depending on the distance between the projection of x and the projected class mean which is expressed as

$$label(x) = arg\min_{i=1,2} |w_i^T x - w_i^T \frac{1}{m_i}\sum_{j=1}^{m_i} x_j^{(i)}|.$$

2.2. *Parameter Selection Algorithm of LSPTSVM*

LSPTSVM needs four parameters c_1 c_4, whose values reflect the classification performance. These parameters generally range from 2^{-8} to 2^8 based on practical cases that even if the parameters take discrete values, the enumeration method is difficult to find the solution in the acceptable time. So enumeration method is not suit for the LSPTSVM parameter selection problem.

Genetic algorithm provides a common framework to solve the complex system optimization problem. Genetic algorithm learns from biological natural selection and natural genetic mechanism, and it is a highly parallel, randomized, adaptive search method. Genetic algorithm works with the individuals in the generation, operating with three basic operations-selection, cross and variation-combining with the fitness function. The scale of the LSPTSVM parameter selection problem is too large to calculate the optimal solution using enumeration method, so that we pay attention to find the acceptable solution. Genetic algorithm is one of the better methods to find these kinds of acceptable solutions.

Algorithm 1. Parameter selection algorithm of LSPTSVM based on genetic algorithm.

Step 1. Discretization. Let the LSPTSVM parameter selection range in the set $\Phi = \{2^{-8}, 2^{-7}, \ldots, 2^7, 2^8\}$.

Step 2. Parameter initialization. Initialize the population number N, the generation number gen, the cross probability $cross\_p$, the metamorphosis probability $mutate\_p$, and set the current generation $gen\_now = 1$.

Step 3. Population initialization. Pick four values from Φ for each individual as its four chromosome that we get the initial population $(v_1, v_2, \ldots, v_N)$.

Step 4. Calculate fitness degree. Run LSPTSVM using the parameters stand for each individual on the same data set and calculate the classification accuracy acc_i. Then calculate fitness degree $adapt_i$ for each individual by

$$adapt_i = acc_i / \sum_{j=1}^{N} acc_j.$$

Step 5. Selection. Firstly, select the "elite individual" which has the highest fitness degree and let it be into the next generation. Then use the roulette selection method to select the other individuals: break the interval $[0,1]$ into pieces and assign them to each individual v_i, where the size of each piece is proportional to the fitness degree of the corresponding individual. And then, generate a random number ranging in $[0,1]$. If this number locates in the small interval assigned to v_i, v_i is selected into the next generation. Repeat the above process $N-1$ times to get the new generation $(v_1^{'}, v_2^{'}, \ldots, v_N^{'})$.

Step 6. Cross. Divide the samples into groups and each group contains 2 samples. For each group, generate a random number ranging in $[0,1]$. If $r < cross\_p$, do the cross operation: generate a random number $r^{'}$ for each chromosome. If $r^{'} < cross\_p$, exchange this chromosome into the sample.

Step7. Metamorphosis. For each chromosome in each individual, generate a random number r ranging in $[0,1]$. If $r < mutate\_p$, replace the chromosome by picking a value in Φ randomly.

Step 8. If $gen\_now < gen$, turn to step4.

Step 9. Output the optimal samples, and then end the algorithm.

2.3. LSPTSVM with confidence-degree

In the decision-making process, the distance d_1 of every new data point x projecting onto the center of positive class is as follow

$$d_1 = |w_1^T x - w_1^T \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)}|,$$

where m_1 is the number of positive sample, and $x_j^{(1)}$ is the j^{th} point of positive sample. Therefore, we can get that the distance d_2 of every projected new data point x to the projected center of negative class is as follow

$$d_2 = |w_2^T x - w_2^T \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)}|.$$

Definition 1. The confidence-degree of LSPTSVM.

The confidence-degree of LSPTSVM refers to the positive degree of the results on decision-making in the training stage. Its expression is as follow

$$\sigma = \frac{\max\{d_1, d_2\}}{d_1 + d_2}. \quad (3)$$

From Eq.(3), we can see that the confidence-degree is high when the projected data x is close to the projected center of one class and far from the projected center of another class (the maximum is 1).

3. The upper preferred multiple directed acyclic graph support vector machines

In order to state the upper preferred multiple directed acyclic graph support vector machine, firstly we give several definitions.

Definition 2.Inaccessible nodes, trusted nodes and trustless nodes.

In the DAG-SVMs classification process, inaccessible node is the node which is not involved in the directed graph classification. Trusted node is the node which confidence value is no less than the threshold of the classifier. Trustless node is the node which confidence value is less than the threshold of classifier node.

Definition 3. Feasible path.

Feasible path is the path starting from the root node in the diagram, to leaf node as the end point without including inaccessible node in the DAG-SVMs.

Definition 4. Trusted leaf node.

Trusted leaf node is the leaf node in the directed graph of DAG-SVMs, which can be reached starting from the root node along a feasible path. Notice that the trusted leaf node is the decision-making results, not the classifier.

Definition 5. Total degree of confidence.

For a trusted leaf node, total degree of confidence is the product of the confidence degree of total nodes including trusted nodes and trustless nodes in a path ending in the trusted leaf nodes. When a number of feasible paths end in the same trusted leaf node, the total degree of confidence is the maximum feasible path confidence value.

Multiple DAG-SVMs is short as MDAG-SVMs. Its decision-making process starts from the root, when faced with a trusted node, decision-making process is along the high confidence branch of the node; When faced with trustless nodes, decision-making process goes along the two branches of the node, until it reaches the leaf node of the directed graph, each feasible path produces a trusted leaf node (maybe repeat). If there is only one trusted leaf node, the classification label of the leaf node value is the classification result. If there is more than one trusted leaf node, calculate the total degree of confidence of trusted leaf node respectively, the classification label of the leaf node, which has the highest total degree of confidence, is the classification result.

3.1. The upper preferred multiple directed acyclic graph support vector machine

In the Multiple DAG-SVMs (short as MDAG-SVMs), classification error occurred in the closer place to the root node, the easier it will eventually sample normalized to the wrong category. At the same time, the closer to the low degree of confidence of the classification of the root node, the more branch path through the nodes are, and the longer the decision time is. Therefore, in the formation process of the directed acyclic graph, we should let the classification performance of a good classifier in the upper nodes of a directed acyclic graph, improve classification accuracy, and reduce the decision-making time.

Definition 6. Class-contained minimal hypersphere.

The minimal hypersphere which can contain all the data belonging to the same classification is called class-contained minimal hypersphere.

Assume that the class S has n samples $x_1, x_2, \ldots, x_n$, the minimal radius of the hypersphere containing the class of these samples is as follows

$$R = \max_{x_i \in S} \{\|\bar{x} - x_i\|\}, \quad (4)$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the center of such a sample set, $\|\|$ is the Euclidean distance operator.

Definition 7. Minimal hypersphere distance.

Suppose there are two classifications of S_1 and S_2, the centers of the class-contained the minimal hypersphere are O_1 and O_2 respectively, and the intersection points of these two class-contained minimal hypersphere and the segment O_1O_2 are P_1 and P_2. The minimal hypersphere distance of S_1 and S_2 is the Euclidean distance of vertex P_1 and P_2. The expression is as follows

$$D = \|O_1O_2\| - R_1 - R_2, \qquad (5)$$

where $\|\|$ is Euclidean distance computing, R_1 and R_2 are class-contained the minimal hypersphere radius of S_1 and S_2 respectively.

The deployment strategy of two classifiers in the upper preferred multiple DAG-SVMs(Up-preferred Multiple DAG-SVMs, short as UMDAG-SVMs) is as follows: (1) calculate the various types of centers and the radius of the class-contained minimal hypersphere; (2) calculate the minimal hypersphere distance between each classification and other classifications; (3)calculate the average value of the minimal hypersphere distance for each classification and other classifications, compare the average value, and select the bigger value to generate the upper classifier of a directed acyclic graph.

Here we give the complete UMDAG-SVMs training algorithm and decision-making algorithm.

Algorithm 2. Training algorithm of UMDAG-SVMs.

Step 1. According to Eq.(4), calculate the radius of all the class-contained minimal hypersphere R_i.

Step 2. According to Eq.(5), calculate the minimal hypersphere distance between each classification and other classifications and generate the distance matrix D_{ij}.

Step 3. Calculate mean value of each line of the matrix D_{ij} and generate the matrix $\overline{D_i}$.

Step 4. Compare the average value of minimal hypersphere distance of each classification, and order the corresponding classification by the above average values, get all the classifications $n_1, n_2, \ldots, n_k$, where $n_m \in \{1, 2, \ldots, k\}$, $m = 1, 2, \ldots, k$ is the classification label. Here, the order of classifications $n_1, n_2, \ldots, n_k$ is as follows.

$$\begin{array}{cccc} n_1 & n_2 & \ldots & n_{k-1} & n_k \\ | & | & & | & | \\ I & III & \ldots & IV & II \end{array}$$

Here I means the classification n_1 has the largest average value of minimal hypersphere distance. II means the classification n_k has the second largest average value of minimal hypersphere distance. III means the classification n_k has the third largest average value of minimal hypersphere distance. IV means the classification n_k has the third largest average value of minimal hypersphere distance.

Step 5. LSPTSVM training algorithm of Binary classification is used to generate the optimal projection axis of each internal node for decision-making oriented cyclic graph. In the root node, the classification n_1 is selected to the positive sample set, and the classification n_k is selected to the negative sample set, LSPTSVM training algorithm generates the optimal projection axis. In the nodes in the second layer, select classification n_1 and classification n_{k-1}, classification n_2 and classification n_k as positive samples and negative samples respectively from the sample set. LSPTSVM training algorithm is used to generate the optimal projection axis. Turn down, and finally the results are as shown in Figure 1.

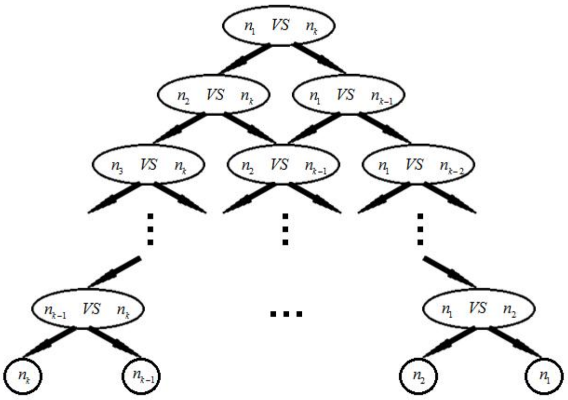

Figure 1 UMDAG-SVMs classification model

Algorithm 3. Recursive decision algorithm of UMDAG-SVMs. Step 1. Initialization. Initialization sample of the total number of categories $class\_len$, Threshold θ, The current classifier line number is $rol = 1$, column number is $col = 1$, the total degree of confidence is $con = 1$.

Step 2. Start recursion. Go to step 3 , and if get the final classification results $label$ and confidence $confidence$, then turn to step 5.

Step 3. Use the classifier of line number rol and column number col to classify unlabeled samples x, get the classifier label not_label (in this classification given sample does not belong to a classification) and confidence-degree c, if the current classification is the leaves in the hierarchical classifiers, that is $rol = class\_len - 1$, according to the judgment not_label, the current sample label $label$ is decided, at the same time , set $confidence = con \times c$, return to step 2. Otherwise, go to step 4.

Step 4. If the confidence level c is greater than the threshold θ, the classifier of the next level should be chosen according to not_label. If the classifier in the right branch is selected, $rol = rol + 1$, $col = col + 1$, and $con = con \times c$, otherwise, $rol = rol + 1$, $col = col + 1$, and $con = con \times c$, go to step 3. If the confidence level c is less than or equal to the threshold θ, unlabeled samples should be classified along the two branches recursively, In the left branch, $con = con \times c$, and in the right branch, $con = con \times (1-c)$, recursively run step 3 respectively, until get the classification results $label1$, $confidence1$ and $label2$, $confidence2$ respectively, return to the classification labels of samples corresponding to the greater confidence-degree value.

Step 5. End.

4. Experimental results

4.1. LSPTSVM parameters selection based on genetic algorithms

In this experiment, the values of all parameters of genetic algorithms are as follows: population size $N = 20$, generations $gen = 20$, crossover probability $cross\_p = 0.6$, and mutation probability $mutate\_p = 0.1$. For fear of the partial results, we use the 10-folds to calculate accuracy and ten independent runs are performed to obtain the mean accuracy. Figure 2 shows the relationship between the generations and the accuracy in the process of selecting the 4 parameters of LSPTSVM by genetic algorithms on complex Xor dataset. The results are $c_1 = 0.25, c_2 = 0.0039, c_3 = 0.125, c_4 - 0.25, accuracy = 98.94\%$, the runtime is 42 seconds. This experiment shows that, on one hand, genetic algorithms can find the parameters which close to the optimal solution in a more acceptable time on the comparison with the enumeration algorithms, on the other hand, the results and the runtime of this algorithm are related to the population size and generations, so this algorithm has controllability to a certain degree.

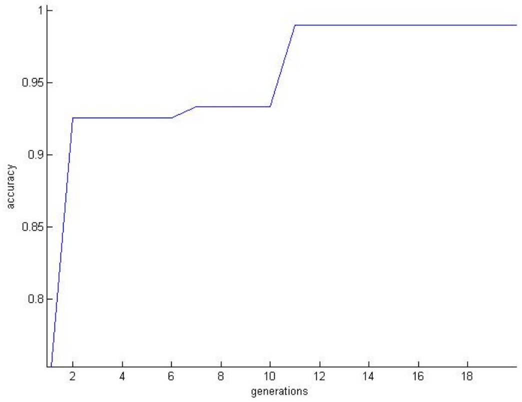

Figure 2 The relationship between generations and accuracy in the genetic algorithm on complex Xor dataset

4.2. LSPTSVM with confidence-degree

According to Eq.(3), the confidence-degree provided by the classifier will be the lowest on the condition that the distance between the projection of one sample point and the projected class mean $d_1 = d_2$. Therefore, the distribution of the point with minimum confidence-degree is expressed as

$$|w_1^T x - w_1^T \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)}| = |w_2^T x - w_2^T \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)}|. \quad (6)$$

Eq. (6) is also expressed as

$$(w_1^T - w_2^T)x = w_1^T \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} - w_2^T \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)}$$

$$(w_1^T + w_2^T)x = w_1^T \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} + w_2^T \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)}. \quad (7)$$

In order to simplify the formulas above, we give the following definition

$$\begin{cases} C_1 = w_1^T \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} - w_2^T \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)}. \\ C_2 = w_1^T \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} + w_2^T \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)} \end{cases} \quad (8)$$

According to Eq.(8), Eq.(7) can be simplified as

$$(w_1^T - w_2^T)x = C_1 or (w_1^T + w_2^T)x = C_2. \quad (9)$$

From Eq.(9), we can see that the distribution of the sample point with minimum confidence-degree, actually, is two hyperplanes S_1 and S_2 which are related to the projection axis W_1, W_2 and the center of the two classifications, which is illustrated in Figure 3 and Figure 4.

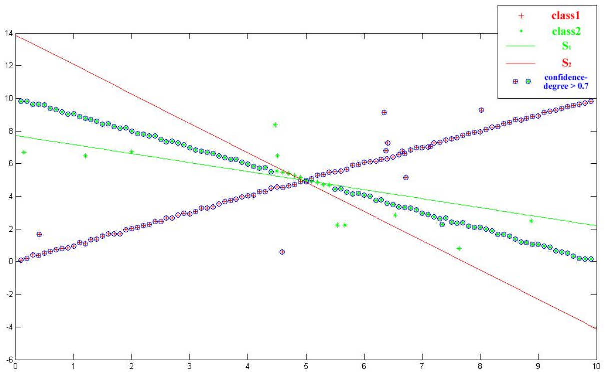

Figure 3 The feature of the confidence-degree on crossplane dataset

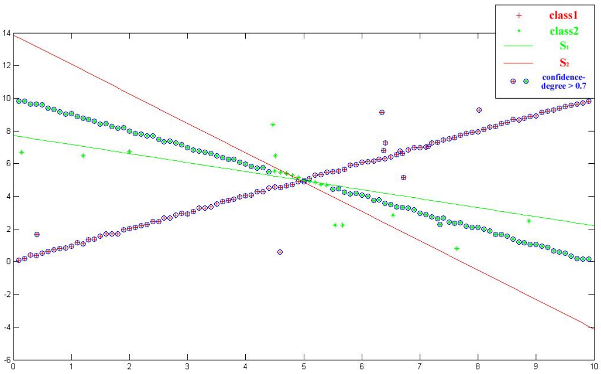

Figure 4 The feature of the confidence-degree on complex Xor dataset

Figure 3 and Figure 4 show that when the sample is closer to hyperplanes S_1 and S_2, it will be harder to classify and its confidence-degree will be lower. So, intuitively, the solution of confidence-degree given by Eq.(3) is reasonable.

4.3. *Comparison among multi-classification algorithms*

We further experimented with 4 UCI benchmark datasets and report the results of the four algorithms on the selected datasets in Table 4.1. All the classification algorithms are implemented in Matlab 7.11.0 environment on a PC with 1.86GHZ processor with 2GB RAM. In order to avoid the partial results, we employ standard 10-fold cross-validation technique.

Table 1 Comparison between four multi-classification algorithms on 4 UCI benchmark datasets

Datasets $(instances \times attributes)$ (categories)		Iris (150×5) (3)	Glass (214×11) (6)	Segmentation (2100×20) (7)	Letter (1160×17) (15)
DAG-SVMs	Accuracy(%)	98	87.73	94.15	83.89
	Training time(s)	0.0224	0.0677	0.571	0.5507
	Decision time(s)	0.0116	0.0334	0.228	0.2057
MDAG-SVMs	Accuracy (%)	98	89.35	97.27	89.06
	Training time(s)	0.0233	0.1102	0.6459	2.7705
	Decision time(s)	0.0156	0.0312	0.2094	0.2475
	Suspect nodes	0	0.3	2.1	37.2
UMDAG-SVMs	Accuracy (%)	98	89.35	96.98	89.91
	Training time(s)	0.039	0.1796	0.8774	3.3270
	Decision time(s)	0.012	0.0265	0.1541	0.1597
	Suspect nodes	0	0	1.3	25.8
UMDAG-SVMs with noise reduction	Accuracy (%)	98	91.97	97.73	92.38
	Noise reduction time(s)	0.0796	0.2459	2.8233	7.133
	Training time(s)	0.0365	0.1694	0.7056	2.846
	Decision time(s)	0.0111	0.0233	0.1051	0.1146

We can infer from the comparison between DAG-SVMs and MDAG-SVMs that,

(1) In the course of decision, the MDAG-SVMs generates a little of suspect nodes when the categories of samples are small. Under this circumstance, the decision process of DAG-SVMs and MDAG-SVMs are similar. (2) The MDAG-SVMs generates a lot of suspect nodes if there are large numbers of categories of samples, contrarily. In such a case, accuracy the MDAG-SVMs achieves is higher than which the DAG-SVMs obtains, but the decision time of the MDAG-SVMs is longer than that of DAG-SVMs.

The comparison between MDAG-SVMs and UMDAG-SVMs shows that: The UMDAG-SVMs, putting the classifiers which have better performance to upper layer of multi-classifier, reduces the quantity of suspect nodes in the course of decision. From Table 1, we can see that UMDAG-SVMs decreases the decision time effectively on the condition that its accuracy is comparable to the accuracy of MDAG-SVMs.

The comparison between UMDAG-SVMs and UMDAG-SVMs with noise reduction indicates that: UMDAG-SVMs with noise reduction can further increase the accuracy, but its training time is longer due to the distance matrixes of every sample in the same class which needs to be solved in the course of decision.

5. Conclusions

In order to solve that the current classification algorithms have weak fault-tolerance, we propose the UMDAG-SVMs. For binary classification, we present LSPTSVM with confidence-degree. In the algorithm, when the confidence-degree outputted from the node in the directed graph is below the given threshold, the decision-making process will go on along with the two branches of the node at the same time, which strengthens the algorithm's fault-tolerance. We use genetic algorithm to select these parameters of the algorithm. Next, according to the minimal hypersphere distance, we present a new classification algorithm called UMDAG-SVMs. This algorithm has two advantages of strong fault-tolerance and high classification accuracy. Finally, Experimental results in publicly available datasets indicate that our UMDAG-SVMs has comparable classification accuracy to that other algorithms.

Acknowledgement

The paper was supported in part by the National Natural Science Foundation of China (No.60903071).

References

[1] J. C. Platt, N. Cristianini, J. Shawe-Taylor, Large margin DAGs for multiclass classification, Advances in Neural Information-Processing Systems. 547-553 (2000).

[2] C. Gruber, T. Gruber, S. krinninger, et al, Online signature verification with support vector machines based on LCSS kernel functions, IEEE Transactions on Systems, Man, and Cybernatics, Part B: Cybernetics, **40**(4), 1088-1100(2010).

[3] I. Kotsia, I. Pitas, Facial expression recognition in image sequences using geometric deformation features and support vector machines, IEEE Transactions on Image Processing, **16**(1), 172-187(2007).

[4] R. Solera-Urena, A. I. Garcia-Moral, Real-time robust automatic speech recognition using compact support vector machines. IEEE Transactions on Audio, Speech, and Language Processing, **20**(4), 1347-1361(2012).

[5] A. C. Lorena, C. P. L. F. de Carvalho, J. M. P. Gama, A review on the combination of binary classifiers in multiclass problems, Artificial Intelligence Review, **30**(1/4), 19-37(2008).

[6] F. Angiulli, Fast nearest neighbor condensation for large data sets classification, IEEE Transactions on Knowledge and Data Engineering, **19**(11), 1450-1464(2007).

[7] D. Isa, L. H. Lee, V. P. Kallmani, et al. Text document preprocessing with the bayes formula for classification using the support vector machine, IEEE Transactions on knowledge and Data Engineering, **20**(9), 1264-1272(2008).

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, et al, Object detection with discriminatively trained part-based models, IEEE Transactions on Knowledge and Data Engineering, **32**(9), 1627-1645(2010).

[9] Y. H. Shao, N. Y. Deng, Z. M. Yang, Least squares recursive projection twin support vector machine for classification. Pattern Recognition, **45**, 2299-2307(2012).

[10] N. Cristianini, J. Shawe-Taylor, An Introduction to support vector machines: and other kernel-based learning methods, Cambridge university Press. Cambridge, New York, (2000).

[11] Y. H. Shao, C. H. Zhang, X. B. Wang, et al, Improvements on twin support vector machines, IEEE Transactions on Neural Network, **22**(6), 962-968(2011).

Kan Li, Ph.D. He is currently a professor in Beijing Institute of Technology. His research interests are in the areas of machine learning, and data mining.

Hang Xu, Master student. His research interests are in the areas of machine learning, and data mining.

Wenxiong Huang, Master student. His research interests are in the areas of machine learning, and data mining.

Zhonghua Huang, Ph.D. He is currently an associate professor in Beijing Institute of Technology. His research interest is intelligent systems.