Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Multi-Stage Decision Tree based on Inter-class and Inner-class Margin of SVM

Mingzhu Lu
Department of Electrical and Computer engineering
University of Texas at San Antonio
San Antonio, TX, USA
lwf054@my.utsa.edu

C. L. Philip Chen
Department of Electrical and Computer engineering
University of Texas at San Antonio
San Antonio, TX, USA
Philip.Chen@utsa.edu

Jianbing Huo
Shijiazhuang Central branch
The People's bank of China
Shijiazhuang, China
hbu_hjb@126.com

Xizhao Wang
College of Mathematics and Computer Science
Hebei University
Baoding, China
xizhaowang@ieee.org

***Abstract*—Motivated by overcoming the drawbacks of traditional decision tree and improving the efficiency of large margin learning based multi-stage decision tree when dealing with multi-class classification problems, this paper proposes a novel Multi-stage Decision Tree algorithm based on inter-class and inner class margin of SVM. This new algorithm is well designed for multi-class classification problem based on the maximum margin of SVM and the cohesion and coupling theory of clustering. Considering the multi-class classification problem as a clustering problem, this new algorithm attempts to convert the multi-class classification problem into a two-class classification problem such that the highest cohesion degree within classes while lowest coupling degree between classes, where the margin of SVM is considered as the measurement of the degree. Then for each two-class problem, this paper uses traditional C4.5 algorithm to generate each stage decision tree which splits a dataset into two subsets for the further induction. Recursively, the Multi-stage decision tree is obtained. Numerical simulations and theoretical analysis show this new multi-stage decision tree improves the performance of traditional decision tree and decreases the computational complexity a lot compare with large margin learning based multi-stage decision tree.**

***Keywords*—inter-class margin, inner-class margin, SVM, multi-stage decision tree**

I. INTRODUCTION

Multi-stage decision tree is a new inductive learning algorithm which aims to better deal with multi-class classification problems. Traditional decision tree algorithm is suitable for two-class problem. Due to its robustness to noisy data, capability of learning disjunctive expression and good readability, traditional decision tree plays a very important role in inductive inference and has been successfully applied to broad practical areas. The most famous algorithm among the family of decision tree is Quinlan's C4.5, which is an extension of the basic ID3 algorithm, because its simplicity and practicability. Quinlan's C4.5 algorithm is very efficient to deal with the two-class problems. However, it has a serious disadvantage--the poor ability to solve the multi-class problems [1, 2]. There are two drawbacks of Quinlan's C4.5 when dealing with multi-class problems: the training speed will become slow and the classification accuracy will decrease acutely.

In order to improve the decision tree's ability of cooping with multi-class classification problems, Huo et al. [2] proposed the large margin learning based multi-stage decision tree, each time the algorithm converts the multi-class problem into a two-class problem using the large margin learning of SVM hyper-planes [3,4,5], which improves the accuracy a lot. However, when the number of class becomes large, the enumeration method to search the maximum margin leads to very slow training speed. Motivated by reducing the computational complexity while keeping the high accuracy, this paper presents the multi-stage decision tree based on the inter-class and inner-class margin of SVM, which is simply called IIMDT. Because the conversion of multi-class classification problem into two-class problem can be considered as a clustering problem, this new method combines the large margin of SVM hyper-planes and clustering idea-high cohesion degree within classes and low coupling degree between classes.

The paper has the following organization. Section II briefly reviews some preliminaries about decision tree and SVM. Section III proposes the multi-stage decision tree based on inter-class and inner-class margin of SVM. Section IV gives the simulation results and theoretical discussion. Finally, the last section concludes the paper.

II. DECISION TREE AND SVM

A. Brief Review of Decision tree Algorithm

Decision tree learning methods search a completely expressive hypothesis space, which can avoid the partial optimization. Given a training set, a general procedure for generating a decision tree can be briefly described as follows.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

The entire training set is first considered as the root node of the tree. Then the root node is split into two sub-nodes based on some heuristic information. If the instances in a sub-node belong to one class, then the sub-node is regarded as a leaf node, else continue to split the sub-node based on the heuristic information. This process repeats until all leaf nodes are generated.

C4.5 is extended from the basic ID3, and their main ideas of are similar. The difference between them is C4.5 algorithm is more applicable and overcome some drawbacks of traditional ID3 [6]. Both of them generate a single decision tree, and the tree divides the training set into several subsets. Each subset is a leaf node, and each path from the root to leaf node can convert into a rule. The decision tree can be expressed by a set of rules finally. Quinlan's C4.5 works well in managing two-class problem, however, when it comes to a dataset with multi-class, it often generates a tree with too many leaf nodes. So the decision tree will become serious complicated, which leads to poor generalization capability.

B. The basic problem of SVM

Considering a two-class classification problem, the basic problem of SVM is to construct an optimal hyper-plane that maximizes the margin between the two classes [3,7,8,9]. Given a training set $U = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ where $x_i \in R^n$ and $y_i \in \{-1, 1\}$ for $i = 1, 2, ...N$. The optimization problem for linear SVM is

$$\min \frac{1}{2}(w \cdot w) \quad \text{s.t. } y_i(w \cdot x_i + b) \geq 1, i = 1, 2, ..., N \tag{1}$$

The optimal hyper-plane of U is defined as $f(x) = 0$,

$$\text{where } f(x) = (w_0 \cdot x) + b_0 \tag{2}$$

$$\text{with } w_0 = \sum_{i=1}^{N} y_i \alpha_i^0 x_i \tag{3}$$

$$b_0 = y_i - x_i \sum_{j=1}^{N} y_j \alpha_j^0 x_j \tag{4}$$

where Eq. (3) and Eq. (4) are obtained by the dual problem of maximizing the margin. For the original problem is an optimization problem with inequality constraints, Lagrange method is used to convert it into its dual problem, which is easier to solve. In Eq. (3), $(w_0 \cdot x) = \sum_{i=1}^{N} w_0^i \cdot x^i$ is the inner product of the two vectors, where $w_0 = (w_0^1, w_0^2, ..., w_0^n)$, and $x = (x^1, x^2, ..., x^n)$.

For the non-linearly separable dataset, the slack variable ξ_i is introduced as the bound of the number of errors. The parameter C is a constant and works as a tradeoff parameter between error and margin. The optimization problem becomes

$$\begin{aligned} &\min \frac{1}{2}(w \cdot w) + C \sum_{i=1}^{N} \xi_i \\ &\text{s.t. } y_i(w \cdot x_i + b) \geq 1 - \xi_i, i = 1, 2, ..., N \end{aligned} \tag{5}$$

Use Lagrange method, its dual problem is obtained:

$$\begin{cases} \text{Maximum } W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) \\ \text{Subject to } \sum_{j=1}^{N} y_j \alpha_j = 0; C \geq \alpha_i \geq 0, \ i = 1, 2, \cdots, N \end{cases} \tag{6}$$

where x_i is a support vector. Then the vector w_0 can be determined according to the quadratic programming (6) and the separating function becomes:

$$f(x) = \sum_{i=1}^{N} y_i \alpha_i^0 (x_i \cdot x) + b_0 \tag{7}$$

By checking whether the following inequalities,

$$y_i[(w_0 \cdot x_i) + b] \geq 1, \quad i = 1, 2, ..., l. \tag{8}$$

holds well or not, we can get whether the two subsets are separable or not. It has been proved that the margin of hyper-plane which classify the samples without error can be calculated by

$$\text{margin} = \frac{1}{\|w\|} \tag{9}$$

The procedure to compute maximum margin for two subsets is described below [5].

Procedure1. The constant C in Eq. (6) is selected to be large at first.

Step1. Solve the quadratic programming (6).

Step2. Determine the separating hyper-plane (7) according to (4).

Step3. Check the separability between two subsets according to inequalities (8).

*Step4.*Let the margin be 0 if the two subsets are not separable.

Step5. Compute the maximum margin according to (9) for the separable case where the vector w is determined by (3).

To improve the performance of SVM, Vapnik extended the SVM from the original space to the feature space [3] by using kernel function, which is a nonlinear function $K(x_1, x_2)$. Then the optimal separating hyper-plane becomes the following form:

$$f(x) = \sum_{i=1}^{N} y_i \alpha_i^0 K(x_i, x) + b_0 \tag{10}$$

C. The inverse problem of SVM

The basic problem of SVM is given labeled data to calculate the maximum margin, while, the inverse problem of SVM [5] is given unlabeled data, finding maximum margin (large margin) separating hyper-plane. The whole process is called large margin learning. It is motivated by finding an optimization solution to split the dataset without class labels into two subsets. For a given dataset without class labels, it can be randomly split into two subsets, the inverse problem of SVM is to solve how to split the dataset such that the margin attains maximum. The inverse problem of SVM can be mathematically formulated as follows:

The dataset $U = \{x_1, ..., x_N\}$, where $x_i \in R^n, i = 1, ..., N$ and $\Omega = \{f | f \text{ is a function from } U \text{ to } \{-1, 1\}\}$. Given a function $f \in \Omega$, the dataset can be split into two subsets and then the margin can be calculated by Procedure1. The

calculated margin is denoted by $\text{Margin}(f)$. Then the inverse problem of SVM is formulated as

$$\text{Maximum}_{f\in\Omega}(\text{Margin}(f)) \tag{11}$$

Due to the increased complexity, it is not feasible to enumerate all possible function in Ω when the number of samples are very large.

III. MULTI-STAGE DECISION TREE BASED ON INTER-CLASS AND INNER-CLASS MARGIN OF SVM

The main idea of multi-stage decision tree is as follows: For a given dataset with multi-class, it first converts the multi-class problem into a two-class problem and re-labels each sample positive or negative. Then for each two-class problem, it uses traditional decision tree algorithm to generate a decision tree, which splits a dataset into two subsets for the further induction. Therefore each stage it generates a decision tree, do this recursively, it obtains a set of decision trees finally.

The inverse problem of SVM was applied to the conversion of multi-class to two-class classification problem [2]. Its drawback is the number of possible conversion solutions by enumeration increase dramatically when the number of class increases. It needs enumerate $2^{(k-1)}-1$ possibilities when deal with an k-class problem [2]. Thus, the inter-class and inner-class margins are proposed to solve the conversion problem in this section. We first construct the inter-class margin matrix by calculating the margin between each two class. Then the definitions of inter-class and inner-class margin are given. Their reciprocals are called coupling degree between classes and cohesion degree within classes respectively. Our aim is to convert the multi-class problem into two-class problem such that the largest inter-class margin between classes and the minimum inner-class margin within classes, i.e. the lowest coupling degree inter classes and the highest cohesion degree inner classes [10, 11, 12].

A. Conversion algorithm based on inter-class margin and inner-class margin

Suppose given a dataset $U=\{x_1,x_2,...,x_N\}$ with classes $C=\{c_1,c_2,....,c_k\}$

Definition1. The number of classes for dataset U is represented by N_c.

$$N_c(U)=k \ , U \text{ is a dataset} \tag{12}$$

If $N_c(U)=k$, the conversion problem needs calculate $2^{(k-1)}-1$ times. When k becomes large, there is a great need to have a new algorithm to enhance the computational efficiency. Even the number of class k is always small enough, each time calculating the margin of $f\in\Omega$ needs to solve a quadratic programming problem as Eq. (6), which costs much time and with exponential complexity.

Definition2. Assume m_{ij} (where $i,j\in 1,2,...,k$) denotes the margin between data belongs to class c_i and class c_j, which is calculated by only using samples belong to c_i and c_j. The inter-class margin matrix M is represented as follows (the matrix is symmetrical, so for simplicity we just keep the upper triangular matrix):

$$M=\begin{pmatrix} 0 & m_{12} & ... & m_{1k} \\ 0 & 0 & ... & m_{2k} \\ ... & ... & ... & ... \\ 0 & 0 & ... & m_{(k-1)k} \\ 0 & ... & ... & 0 \end{pmatrix} \tag{13}$$

Definition3. Given a dataset $U=\{x_1,x_2,...,x_N\}$ with classes $C=\{c_1,c_2,....,c_k\}$, C is divided into two subsets (C_p,C_n) and accordingly U is divided into two subsets (U^+,U^-), the **inter-class margin** is defined as:

$$\text{Margin}_{\text{inter}}(C_p,C_n)=\min\{m_{ij},c_i\in C_p,c_j\in C_n\} \tag{14}$$

where C_p is the positive super class set and C_n represents the negative super class set; all the data belong to class C_p forms U^+ and the rest dataset belong to class C_n is called U^-. The reciprocal of $\text{Margin}_{\text{inter}}(C_p,C_n)$ is called the coupling degree between class C_p and C_n. Obviously, the margin between two super classes should be larger than any margin inner the super class, which can be formulated:

$$\text{Max}(m_{ij},\ c_i,c_j\in C_p \text{ or } c_i,c_j\in C_n)\le \text{Margin}_{\text{inter}}(C_p,C_n) \tag{15}$$

Definition4. Given a dataset $U=\{x_1,x_2,...,x_N\}$ with classes $C=\{c_1,c_2,....,c_k\}$, C is divided into two subsets (C_p,C_n) and accordingly U is divided into two subsets (U^+,U^-), the **inner-class margin** is defined as:

$$\text{Margin}_{\text{inner}}(C)=\frac{\sum m_{ij}^{+}+\sum m_{ij}^{-}}{l^{+}+l^{-}} \tag{16}$$

where m_{ij}^{+} (where $i,j\in 1,2,...,k$) represents the elements of M^+ and M^+ denotes the inter-class margin matrix of U^+; l^+ is the number of nonzero elements in M^+; M^- represents the inter-class margin matrix of U^-, and m_{ij}^{-} (where $i,j\in 1,2,...,k$) denotes its elements, l^- is the number of nonzero elements in M^-. The reciprocal of $\text{Margin}_{\text{inner}}(C)$ is also called cohesion degree within class.

Our task is to convert the multi-class into a two-class classification problem such that the maximum of inter-class margin and meanwhile minimal of inner-class margin. In the point view of clustering, it means the high cohesion degree within each super class and low coupling degree between super classes. The new conversion algorithm will first assume all the classes belong to C_p, and the negative super class is null. Each step it chooses a proper class to add to negative super class. Then check whether the two new super classes satisfy the inequality in Eq. (15), if not, continue this step

because this means there are still some classes in C_p can be separated into C_n. Otherwise, the conversion algorithm stops. Thus the criterion of selecting the class which is added to negative super class becomes an important topic. The class which minimizes the inner-class margin will be added to negative super class. The change of inner-class margin will be the measurement.

Definition5. Given a dataset $U = \{x_1, x_2, ..., x_N\}$ with $C = \{c_1, c_2, ..., c_k\}$, C is divided into two subsets (C_p, C_n), the original inner-class margin is $\text{Margin}_{\text{inner}}(C)$, then choose a class $c_i (i = 1, 2, ..., k)$ from super positive class set C_p and add it to super negative class set C_n, the inner-class margin becomes $\text{Margin}_{\text{inner}}(C)'$, the change of inter-class margin is defined as:

$$\Delta\text{Margin}_{\text{inner}}(C, c_i) = \text{Margin}_{\text{inner}}(C) - \text{Margin}_{\text{inner}}(C)' \quad (17)$$

Each time choose class c_i from super positive class which satisfies to

$$\max_{i=1,...,k} \{\Delta\text{Margin}_{\text{inner}}(C, c_i), c_i \in C_p\} \quad (18)$$

i.e. choose the class which decreases the inner-class margin most quickly. The conversion algorithm based on the inter-class and inner-class margin can be described as follows:

Procedure2.

Step1. Calculate the inter-class margin matrix M. Label all the classes positive and let C_n be null.

Step2. Choose the class c_i from the super positive class C_p which satisfies to equation (18). Check whether the values of Eq. (18) greater than zero, if yes, add class c_i to C_n; otherwise, stop.

Step3. Check whether the new subsets C_p and C_n satisfy to the inequality in Eq. (15) or not. If not, go to step 2; if yes, stop.

Compare with the enumeration method, this conversion algorithm based on the inter-class and inner-class margin of SVM just needs to calculate the inter-class margin matrix M. It needs calculate $C_k^2 = \frac{k(k-1)}{2}$ times instead of $2^{(k\text{-}1)} - 1$ times when deal with an k -class problem. When k is large, the advantage of this new algorithm is more prominent. Besides, its calculation only needs to use two class datasets each time. While the enumeration method will use equal to or greater than two class datasets each time, even the whole dataset. The quadratic programming with smaller dataset will be easier and quicker to solve, so the induction of inter-class margin and inner-class margin enhance the computational efficiency a lot. Procedure 2 uses the change of inner-class margin to split the multi-class to a two-class problem. The computational complexity of addition operation is much lower than the complexity of quadratic programming. Thus this method is more efficient.

B. Multi-stage decision tree algorithm based on inter-class and inner-class margin of SVM

The main idea of multi-stage decision tree algorithm based on inter-class and inner-class margin of SVM is to convert the multi-class classification problem into a two-class problem by *procedure 2* and re-label the initial dataset. Then for each two-class problem, it uses traditional C4.5 algorithm to generate a decision tree, which splits a dataset into two subsets for the further induction. Therefore, a decision tree is generated in each stage and the decision tree will split the dataset into two subsets--positive and negative subset. After this, we need to check whether each subset is still a multi-class classification or not. If yes, the data is treated as the initial dataset. If it becomes a two-class classification problem, C4.5 is used to generate one stage decision tree directly. This procedure will continue recursively until a set of decision trees are obtained completely.

Each individual decision tree will be converted into a set of rules, and the method of generation of rules is the same as C4.5 algorithm [2, 6]. Because the decision tree is hierarchical, the matching process of the rules is orderly. If a sample matches one stage decision tree successfully with a super class contains more than one class, it needs a further match to the next stage decision tree. An instance matches succeed if and only if it reaches a leaf node labeled just by a single class.

IV. SIMULATIONS AND DISCUSSIONS

A. An illustrative example

Take Glass database [13] for example to show how the inter-class and inner-class margin based method works, where RBF kernel function is used in the simulations and let the parameter C in Eq. (6) be 10000. Following the procedure 2, first the inter-class margin matrix is obtained:

$$M = \begin{pmatrix} 0 & 0.0057 & 0.0088 & 0.6201 & 0.4482 & 0.0242 \\ 0 & 0 & 0.0102 & 0.2355 & 0.1947 & 0.0962 \\ 0 & 0 & 0 & 0.7211 & 0.5920 & 0.0783 \\ 0 & 0 & 0 & 0 & 0.4461 & 0.4503 \\ 0 & 0 & 0 & 0 & 0 & 0.3415 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Then the initial class set is $C_p = \{c_1, c_2, c_3, c_4, c_5, c_6\}$ and C_n =null. From the inner-class margin definition, the follow equation holds true:

$$\text{Margin}_{\text{inner}}(C) = \frac{\sum m_{ij}^+ + \sum m_{ij}^-}{l^+ + l^-} = \frac{\sum m_{ij}^+ + 0}{15 + 0} = 0.28486$$

Next, follow the step 2, choose $c_i (i = 1, 2, 3, 4, 5, 6)$ to add to C_n and check $\Delta\text{Margin}_{\text{inner}}(C, c_i)$. Taking c_1 for example, add c_1 to C_n then the two class subsets become $C_p = \{c_2, c_3, c_4, c_5, c_6\}$ and $C_n = c_1$.

$$\text{Margin}_{\text{inner}}(C)' = \frac{\sum m_{ij}^+ + \sum m_{ij}^-}{l^+ + l^-} = \frac{\sum m_{ij}^+ + 0}{10 + 0} = 0.31659$$

then the change of inner-class margin becomes:

$$\Delta\text{Margin}_{\text{inner}}(C,c_1)=\text{Margin}_{\text{inner}}(C)-\text{Margin}_{\text{inner}}(C)'$$
$$=0.28486-0.31659<0$$

Similarly, check the change of inner-class margin when add $c_i(i=2,3,4,5,6)$ to C_n and we get:

$$\max\{\Delta\text{Margin}_{\text{inner}}(C,c_i),c_i\in C_p\}=\Delta\text{Margin}_{\text{inner}}(C,c_4)>0.$$

Accordingly, the new subsets are $C_p=\{c_1,c_2,c_3,c_5,c_6\}$ and $C_n=\{c_4\}$, furthermore, $\text{Margin}_{\text{inter}}(C_p,C_n)=0.2355$. It is easy to check that the new subsets do not satisfy to Eq. (15), so go to step2 to continue to choose. Now the inter-class margin matrix becomes:

$$M=\begin{pmatrix}0 & 0.0057 & 0.0088 & 0.4482 & 0.0242\\ 0 & 0 & 0.0102 & 0.1947 & 0.0962\\ 0 & 0 & 0 & 0.5920 & 0.0783\\ 0 & 0 & 0 & 0 & 0.3415\\ 0 & 0 & 0 & 0 & 0\end{pmatrix}$$

Because C_n just includes one class now, $\text{Margin}_{\text{inner}}(C)$ becomes 0.17998. Similarly, calculate the change of inner-class margin when add $c_i(i=1,2,3,5,6)$ to $C_n=\{c_4\}$,at the end of this step we get:

$\max\{\Delta\text{Margin}_{\text{inner}}(C,c_i),c_i\in C_p\}=\Delta\text{Margin}_{\text{inner}}(C,c_5)>0$. Then we have $C_p=\{c_1,c_2,c_3,c_6\}$ and $C_n=\{c_4,c_5\}$. Next, check $\text{Margin}_{\text{inter}}(C_p,C_n)$ and find out it still does not satisfy to the condition (15), so continue the step2. The inter-class margin matrix becomes:

$$M=\begin{pmatrix}0 & 0.0057 & 0.0088 & 0.0242\\ 0 & 0 & 0.0102 & 0.0962\\ 0 & 0 & 0 & 0.0783\\ 0 & 0 & 0 & 0\end{pmatrix}$$

For this step, $\max\{\Delta\text{Margin}_{\text{inner}}(C,c_i),c_i\in C_p\}<0$, thus, procedure 2 stops with $C_p=\{c_1,c_2,c_3,c_6\}$ and $C_n=\{c_4,c_5\}$.Treating $C_p=\{c_1,c_2,c_3,c_6\}$ as positive class and $C_n=\{c_4,c_5\}$ as negative class, we use traditional C4.5 algorithm to get the first stage decision tree. Then check the function $N_c(U^+)$ and $N_c(U^-)$. We get $N_c(U^-)=2$, which means a two-class problem, now we use C4.5 to generate the next stage decision tree directly. The fact $N_c(U^+)=4>2$ means C_p is still a multi-class problem, consider $C_p=\{c_1,c_2,c_3,c_6\}$ as a new multi-class problem and follow procedure 2 to convert it into a two-class problem, and then generate the next stage decision tree. Finally a multi-stage decision tree is obtained by this recursive process.

B. Simulation results on UCI database

Numerical simulations are implemented on four multi-class UCI datasets [13]. Table I shows the features of the databases. The comparison results of training accuracy and testing accuracy among the C4.5, large margin learning based multi-stage decision tree (LMMDT) and IIMDT are listed in Table II, where 10-fold cross validation is used.

TABLE I. FEATURES OF THE DATASETS

Name of Databases	Number of cases	Number of attributes	Number of classes
Car	1728	6	4
Derm	396	33	6
Glass	214	9	6
Ecoli	336	7	8

TABLE II. COMPARISON OF TRAINING AND TESTING ACCURACY BASED ON 10-FOLD CROSS VALIDATION

Database Name	C4.5		LMMDT		IIMDT	
	Train	Test	Train	Test	Train	Test
Car	99%	88.5%	99.2%	92.7%	98.7%	91.4%
Derm	97.6%	90.6%	98.0%	94.1%	97.82%	94.5%
Glass	70.9%	60.0%	78.8%	69.5%	78.5%	67.5%
Ecoli	83.2%	71.2%	87.5%	74.5%	86.0%	73.9%

Compare to C4.5, the IIMDT algorithm improves the test accuracy a lot when treating with multi-class classification problems; meanwhile, compare with LMMDT, IIMDT keeps or even exceeds the testing accuracy of LMMDT. While as analysis at the end of section III A, the IIMDT decreases the computational complexity a lot. Therefore, when the number of class k is large, IIMDT takes advantages of both accuracy and computational efficiency, which is more feasible.

C. Discussions

Here are some discussions about the generalization capability of IIMDT algorithm compare to traditional decision tree and LMMDT. Since the multi-stage decision tree generates a set of decision trees instead of only one decision tree, the obtained rules become simpler and shorter comparing to C4.5, which implies a better generalization capability according to Occam's razor theorem [6].

IIMDT first converts the multi-class problem into a two-class problem by calculating inter-class and inner-class margin of SVM. It guarantees the minimum margin within classes and the maximum margin between super classes. The measurement of cohesion degree and coupling degree becomes more reasonable by using the margin of SVM. It considers not only the margin between classes but also the margin within classes. For a clustering problem, this comprehensive consideration often leads to better results. Therefore, this conversion decreases the complexity of classification for decision tree even more and provides better heuristic information for multi-stage decision tree. It often leads to simpler rules with a higher truth level and improves generalization capability.

V. CONCLUSION

In order to improve the computational efficiency and classification accuracy of multi-stage decision tree, this paper proposes a new algorithm called multi-stage decision tree based on inter-class and inner-class margin of SVM to deal

with multi-class classification problems. After a brief review of decision tree and SVM, this paper presents a new multi-class to two-class conversion algorithm for multi-stage decision tree, which uses the margin of SVM as measurement and takes into account the clustering criterion-high cohesion degree within classes and low coupling degree between classes. Later, an example is given to show how this new algorithm works. Finally, the simulation results and theoretical discussions verify its effectiveness and feasibility. In the future, its application to text classification and pattern recognition will be investigated.

REFERENCES

[1] Quinlan J R., "Induction of decision tree," Machine Learning, vol. 1, no. 1, pp. 81-106, 1986.

[2] Jianbing Huo, Xizhao Wang, Mingzhu Lu, et al., "Induction of multi-stage decision tree," IEEE Conference on System, Man and Cybernetics, Taipei, pp. 835-839, 2006.

[3] V. N. Vapnik, Statistical learning theory, Wiley, New York, 1998.

[4] V. N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 2000.

[5] Xizhao Wang, Qiang He, Degang Chen, et al, "A genetic algorithm for solving the inverse problem of support vector machines", Science Direct, vol. 68, pp. 225-238, 2005.

[6] Tom M. Mitchell. Machine Learning, China Machine Press, Beijing, 2003.

[7] V. N. Vapnik, "An Overview of Statistical Learning Theory," IEEE Transactions on Neural Networks, vol. 10, no.5, pp. 988 – 999, 1999.

[8] Xizhao Wang, Qiang He, "Enhancing Generalization Capability of SVM-Classifiers with Feature Weight Adjustment," Lecture Notes in Computer Science, vol. 3213, pp.1037-1043, Sep. 2004.

[9] Chih-Wei Hsu and Chih-Jen Lin, "A Comparison of Methods for Multiclass Support Vector Machines", IEEE Transactions on Neural Networks, vol. 13, no. 2, pp. 415-425, Mar. 2002.

[10] Agrawal, R., Srikant, R., "Fast algorithm for mining association rules. In: Jorge, B.B, Matthias, J., Carlo, Z., eds," Proceedings of the 20th International Conference on Very Large Databases, Santiago: Morgan Kaufmann Publishers, Inc., pp. 487~499, 1994.

[11] Qinbao Song, Junyi Shen, "A Web Document Clustering Algorithm Based on Association Rule," Journal of software, vol. 13, no.3, pp.417~423, 2002.

[12] David Hand, Heikki Mannila, Padhraic Smyth, Principle of Data Mining, The MIT Press, Cambridge, Massachusetts, London, England,2001.

[13] UCI Repository of machine learning databases and domain theories. FTP address: ftp:// ftp.ics.uci.edu/pub/machine-learning-databases.