A New Approach for Decision Tree Based on Principal Component Analysis

Juanli Hu, Jiabin Deng, Mingxiang Sui
Department of Computer Engineering
Zhongshan Polytechnic
Zhongshan, China
hjlfoxes@163.com

Abstract—**Classification algorithm has always been a hot issue in data mining. Decision tree algorithm is the most active part in this area, but it is a NP problem to construct the optimization decision tree. With the development of the information collection technology, the requirements of the mass data mining have become increasingly higher. When dealing with large, continuous, even with the noise and abnormal data, the traditional decision tree algorithm seems very incompetent, encountering the efficiency of the bottleneck and classification error. In this paper, there exist the shortcomings for the decision tree algorithm to deal with multi-attribute data sources. The multivariate statistical methods is proposed to make the principal component analysis on multi-attribute data, reducing dimensionality, devoicing processing and transforming the traditional decision tree algorithm to form a new algorithm model. Comparing with the traditional decision tree algorithm, the experimental results show that this method can not only simplify the decision tree model, but also can improve prediction accuracy of the decision tree.**

Keywords-component; Data mining; Classification, Multivariate statistical; Principal components analysis; Decision tree

I. INTRODUCTION

The classification is an important part and the research application field in the data mining. The construction methods of the traditional classification models are such as statistics, neural networks, genetic algorithms, radial basis function, rough Set, fuzzy set and K-nearest neighbor classification method [1-2]. These methods in the classification knowledge discovery field have achieved satisfactory results, but the most widely used classification model is still the decision tree algorithm. Because of the clear principle, the simple application, the high efficiency and the better classification accuracy, the decision tree classification model is widely used.

The most commonly used decision tree algorithms are the ID3 in 1986 and the improved C4.5 in 1993 [3-4]. Most of the decision tree algorithms are based on the above algorithms. Such algorithms generally use the top-down greedy algorithm, choosing the best category property in the each node to continue the next step. The optional method is the impure measurement method. The criteria value of the impure measurement method can be the entropy, the information gain, the cost of complexity, the erroneous judgment instance number, the weight of evidence and so on. Different metrics have different effects. The classical decision tree algorithm uses the information gain as the attribute selection criteria [5]. In the case of the less data, the average depth of the decision tree constructed by such algorithms is small. But with the increase of data set properties, when the amount of data is very large, classification speed decreases significantly and the average depth of decision tree becomes deeper. The tree structure will become large and complex. The resultant knowledge rule set will become large and complex. However, research shows that the large and complex decision tree does not mean the more accurate rule sets [6].

In this paper, for the existence of the Quinlan's decision tree algorithm problems, facing the mass data analysis, we use the principal component analysis to filter noise data and to reduce the dimension of data sets [7]. Improving the C4.5 decision tree algorithm to form the classification model, we can choice the classification rules to meet the needs. Through the practical application detection, we achieve the further comparison to verify the correctness of the results.

II. RELATED WORKS

Considering the issues above, many scholars have put forth a variety of optimization algorithms for decision tree. At present, there are many ways to optimize the decision tree mainly including decision tree pruning, modifying the test attribute space, adapting the selection of the test properties, and using other data structures. Pruning method is one of the most popular and successful method that widely used in decision tree construction. In this section, the two principle and method of pre-pruning and post-pruning algorithm for decision tree are discussed.

A. The principle and method of Pre-pruning algorithm

As the decision tree algorithms mentioned above, all are required that the training examples in each leaf node should be belonged to the same category as the stop condition of algorithm. In such circumstance, the error rate of the training data in the decision tree is 0 (for consistency data). However, the standard is not used in the pre-pruning algorithm as the stopping criteria, but to stop the expansion before meeting such standard. When to stop the growth of this algorithm has become the main research.

978-1-4244-4507-3/09/$25.00 ©2009 IEEE

One of the simplest method is setting a threshold (often 0.25) for each sample when arriving the node. When the number of the samples is less than the threshold, the growth of decision tree will be stopped. Another common practice is to calculate the impact on each expansion of system performance, and it will not be extended if the gain is less than the threshold.

However, the branch will be ceased earlier because of the lack of sufficient forward algorithm prematurely, which is known as the "horizon effect" phenomenon [4]. Node N in the best branch of the agency does not consider the impact on the optimal level decision-making of the below node. Once stopping the branch and making N as a leaf node, it makes impossible that its follow-up nodes become the "good" branch operation. In this way, it is stopped too early and may lose some important information according to the stop condition for the overall recognition rate. Whatsoever, due to pre-pruning decision tree do not have to generate the full decision tree, it makes the algorithm has a high efficiency for large-scale problems. The typical algorithm by this strategy is PUBLIC algorithm proposed by Rastogi et al. in 2000 [8-9].

B. The principle and method of Post-pruning algorithm

Post-pruning algorithm is not like pre-pruning, it has two major stages: fitting and simplification. First of all, the decision tree is generated with training data entirely fitting, and then to delete one or more than one sub-tree and replace by the leaf node. The category of this leaf belonging to is identified by the category that the majority of training examples are belonged to in the sub-tree. Post-pruning algorithm can be divided into two categories: one is the training data sets are divided into tree growth sets and tree pruning sets; the other is using the same training data sets in the stage of tree growth and tree pruning.

Common Post-pruning methods are as follows: CCP (cost-complexity pruning), REP (reduced error pruning), PEP (pessimistic error pruning), MEP (minimum error pruning) and so on. Post-pruning can resolve the problem that the growth is stopped too earlier. Because post-pruning is usually carried out after the decision tree created, the efficiency of generating the optimal decision tree is not improvement although it can achieve the purpose of knowledge rules simplifying.

III. DECISION TREE CONSTRUCTION BASED ON DIMENSIONALITY REDUCTION FUNCTION

Considering the problems in decision tree optimization, a novel approach of decision tree construction is presented based on principal components analysis in this paper. In order to construct the optimizing decision tree and remove the pruning, noise and abnormal data should be filtered and cleaned before generating decision tree.

A. The methods of principal components analysis

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

The principal components analysis mainly uses the idea of dimensionality reduction, which is a multivariate statistical method that transforming a number of indicators into a composite indicator under the premise of little loss of information. Usually, the composite indicator generated by transformation is known as the principal component, where each principal component is a linear combination of original variable and all the main components are isolated, making the principal component have a better performance than the original variable. In the study of complex issues, we can only consider a small number of principal components without loss of too much information so as to grasp the main contradiction more easily, simplify the problem and improve the analysis efficiency at the same time.

Principal component analysis is defined as follows:

Assume that $X=(x_1,x_2,x_3\cdots,x_n)$ as n-dimensional random variable and its pth principle component is $y_i=E_iX(E_iE_i^T=1\ \ i=1,2,3\cdots n)$, where E_i is the ith feature vector in X.

Therefore, n principal components' expressing of n variables' are n linear combinations of n variables', of which the coefficient vector of linear combination is a unit vector. The first principal component y_1 is the maximum variance in all possible linear combinations and the second principal component y_2 is the maximum variance in all linear combinations that not related to y_1. The third principal component y_3 is the maximum variance in all linear combinations that not related to y_1, y_2 and so on… The contribution rate of y_k is the proportion of the kth variance to the total variance, which is computed by $\lambda_k \Big/ \sum_{i=1}^{n}\lambda_i$. The cumulative contribution rate of the first few k principal components $(y_1,y_2,y_3\cdots y_k)$ denotes the principal components information extracted by $x_1,x_2,x_3,\cdots,x_n$, which calculated by $\sum_{j=1}^{k}\lambda_j \Big/ \sum_{i=1}^{n}\lambda_i$. If the cumulative contribution rate of the first k principle components reached the threshold (often 75% from -85%), it shows that all the basic information measurement indicators are covered. Thus, it can not only reduce the number of variables but also easy for analysis.

B. A new method for decision tree construction

In combination with principal components analysis and Decision tree with their characteristics, firstly, filter the sample data set, then extract the main attributes, and lastly construct a new decision tree by the following algorithm. The detailed is as follows:

Step 1 Convert data source into a multi-matrix, identify the main attributes by principal components analysis.

- Data matrix conversion

$$X=[x_1,x_2,x_3,\cdots,x_n]=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{p1} & x_{p2} & \cdots & x_{pn} \end{bmatrix} \quad (1)$$

where p is object's attributes and n is the attribute value.

- Calculate the standardization of each component by combining all the discrete data.

$$x_i^{'}=\frac{x_i-\overline{x_i}}{\sqrt{l_{ii}/(n-1)}} \qquad (i=1,2,3,\cdots n) \quad (2)$$

Compute the main attribute

$$\eta(p)=\sum_{j=1}^{p}\lambda_j \Big/ \sum_{i=1}^{n}\lambda_i \quad (j=1,2,3,\cdots p;\ i=1,2,3,\cdots n) \quad (3)$$

- The value of $\eta(p)$ is depended on the actual case (often $>85\%$), and the main attribute is set by the value of p.

Step2 Do data cleaning for data source and generate the training sets of decision tree through converting the continuous data into discrete variables.

Step3 Compute the information (entropy) of training sample sets, the information (entropy) of each attribute, split information, information gain and information gain ratio, of which S stands for the training sample sets and A denotes the attributes.

- Compute the information (entropy) of training sample sets S

$$I(S)=-\sum_{i=1}^{m} p_i \log_2 p_i \quad (4)$$

where p_i is the probability of category C_i in S.

- Compute the information (entropy) of attribute A

$$E(S,A)=\sum_{i=1}^{m}\frac{S_i}{S}I(S,A) \quad (5)$$

- Compute information gain of A

$$Gain(S,A)=I(S)-E(S,A) \quad (6)$$

- Compute split information of A

$$Split\_Info(S,A)=-\sum_{i=1}^{m}\frac{|S_i|}{S}\log_2\frac{|S_i|}{S} \quad (7)$$

- Compute information gain ratio of A

$$Gain-Ratio(S,A)=\frac{Gain(S,A)}{I(S,A)} \quad (8)$$

For continuous attribute values, calculate information gain ratio corresponding with the segmentation points divided by $a_i(i=1,2,3,\cdots,n-1)$ and choose the maximum rate of information gain a_i as the split points of attribute classification.

Choose the choose the maximum attribute of information gain as the decision tree root.

Step4 Each possible value of root may correspond to a subset. Do step 3 recursively and generate decision tree for the sample subset until the observed data of each divided subset are the same in the classification attributes.

Step5 Extract the classification rules based on the constructed decision tree and do classification for new data sets.

IV. CASE STUDY AND COMPARATIVE ANALYSIS

A. The Improved Decision Tree Algorithm Example

Table 1 shows the customer loan database of a bank. There are seven condition attributes, some of which are continuous values. A decision-making property is the discrete value, a total of about 10,000 records.

TABLE I. THE CUSTOMER LOAN DATABASE OF A BANK

Case	*Birthday*	*Gender*	*Education*	*Annual Income*	*Loan Amount*	*The Use*	*Credibility*	*Repayment On time*
1	1965-7	M	High School	70000	300000	Purchase	1	Yes
2	1962-10	F	College	35000	80000	Decoration	1	Yes
3	1975-3	F	College	500000	2000000	Investment	3	No
.....	.....	.....	.....	.....	.....	.....	.....	.....
10012	1972-11	F	Undergraduate	45000	75000	Wedding	0	Yes
10013	1980-5	M	Master	0	35000	Study	0	Yes
10014	1956-12	F	Junior High School	47000	100000	Treatment	1	No

Making use of the type (1), we can convert data source into the multi-matrix; Making use of the type (2), we can get the correlation matrix shown in figure 1.

$$\begin{bmatrix} 1.0000 & & & & & & \\ -0.2732 & 1.0000 & & & & & \\ -0.4894 & 0.4274 & 1.0000 & & & & \\ 0.2326 & -0.0895 & -0.3427 & 1.0000 & & & \\ 0.7568 & -0.2813 & -0.6735 & 0.1672 & 1.0000 & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0.4413 & -0.1748 & -0.5145 & 0.5698 & 0.3481 & \cdots & 1.0000 \end{bmatrix}$$

Figure 1. The matrix after the component standardization

The unit eigenvector corresponding to the Eigen value is 3.2837, 3.0451, 2.7846, 2.3485, 1.1447, 0.4754 and 0.0383.

According to the formula (3), we can get the following results.

$$\eta(6)=\frac{3.2837+3.0451+2.7846+2.3485}{3.2837+3.0451+2.7846+2.3485+1.1447+0.4754+0.0383}=87.36\%$$

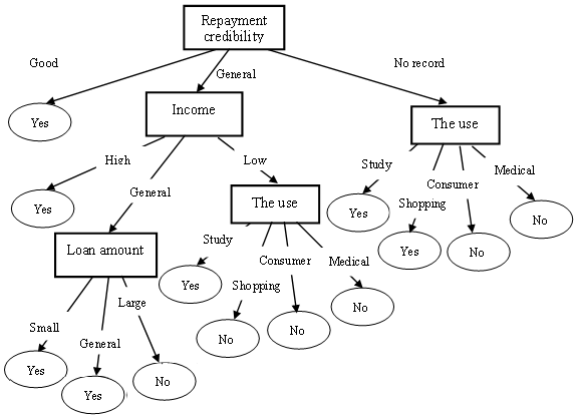

Figure 2. The decision tree model of the dimensionality reduction function

Because of ＞85%, so we only need credibility repayment, income, loan amount and loan purpose to meet the attribute achievements of the decision Tree.

According to step2-step4 of the decision tree algorithm in the dimensionality reduction function, the returned final decision tree is shown in Figure 2.

B. The Classic Decision Tree Algorithm Example

According to the data source provided by the table 1, the decision tree constructed by the C4.5 algorithm is shown in figure 3.

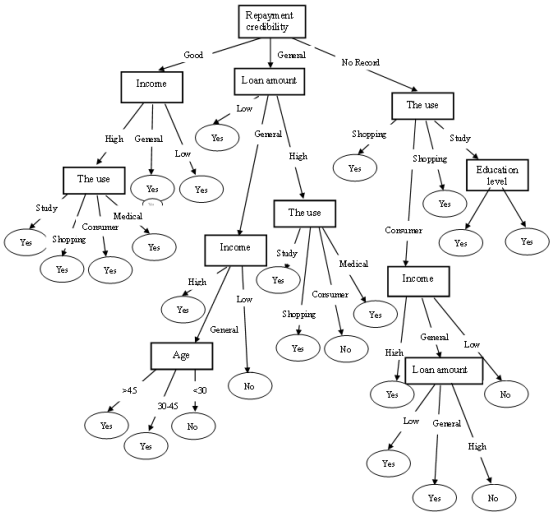

Figure 3. The C4.5 decision Tree

C. The comparison between the traditional decision tree algorithm and the improved algorithm example

From the two maps and table 2, we can draw the conclusion: The decision tree structure based on the dimensionality reduction function is better. It significantly reduces the height and the complexity of the decision tree. It is a large extent to overcome the efficiency of the bottleneck problem when the traditional decision tree algorithm deals with mass data in easy to encounter. It is clear that when dealing with the mass data of the data sets, the decision tree constructed by the dimensionality reduction function is fit for the actual situation.

TABLE II. THE COMPARATIVE TABLE OF THE EXPERIMENTAL DATA

Construction Method	Training Set	Test Set	Attribute Number	Tree Height	Leaves	Type Number
C4.5	10014	10014	7	5	25	7
Dimensionality Reduction Function	10014	10014	4	4	13	5

V. CONCLUSIONS

At present, the technologies in data mining are facing massive and multi-attribute data. In this paper, a decision tree building model of reduced-order function based on principal component analysis was presented. Firstly, the noise data were filtered by principal component analysis. After that, the attributes of the knowledge rule sets that hidden in multi-attributes were extracted. At last, the decision tree was created. The results showed that this method can not only improve the efficiency when processing with the massive data using the decision tree algorithm, but also optimize the structure of decision tree, improve the problems existing in pruning algorithms and mine the better rules without affecting the purpose of prediction accuracy. On the condition of the decision tree with complex structure, the efficiency and performance can be better and the advantages are more obvious.

Due to the amount of teaching data and different attributes, how to choose and merge the classification attributes is becoming a hard topic. In addition, the areas in continuous attributes with the simplest and the most reasonable discretization are worthy of further exploration and research.

REFERENCES

[1] Jiawei Han, Micheline Kamber Data mining: concepts and techniques: Second Edition, illustrated. Morgan Kaufmann Publishers, Inc, 2006.

[2] Mitra Sushmita1, Pal Sankar K, Mitra Pabitra IEEE Transactions on Neural Networks [J]. 13(1), 2002.

[3] Quinlan J R Indution of decision trees[J]. Machine Learning, 1986, 1(1):81 - 106.

[4] Quinlan J R. C4.5: Pragrams for machine learning [M]. California: Morgan Kaufmann Publishers, Inc,1993.

[5] Thomas G. Dietterich, An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization [J]. Machine Learning, 2000, 40, 139–157.

[6] Oates T, Jensen D. The effects of training set sizes on decision tree [A]. Proc of the 14th Int'l Conf on Machine Learning [C]. Nashville: Morgan Kaufman, 1997, 254-262.

[7] I. T. Jolliffe, Principal Component Analysis: Second Edition, Springer Publishers, 2002.

[8] M Mehta, R Agrawal. SLIQ: A fast scalable classifier for data mining [C]. Proceedings of the 5th International Conference on Extending Database Technology, 1996: 18-23.

[9] Pawlak Z., Rough Sets [J]. International Journal of Computer and Information Sciences, 1982(11): 341-356.