Systems Engineering — Theory & Practice
Volume 29, Issue 12, February 2009
Online English edition of the Chinese language journal

Available online at www.sciencedirect.
ScienceDirect

Cite this article as: SETP, 2009, 29(12): 94–104

RESEARCH PAPER

C5.0 Classification Algorithm and Application on Individual Credit Evaluation of Banks

PANG Su-lin , GONG Ji-zhang

Institute of Finance Engineering, School of Management, Jinan University, Guangzhou, 510632

Abstract: This article focuses on individual credit evaluation of commercial bank. The records of individual credit include both numerical and nonnumeric data. Decision tree is a good solution for this kind of issue. This year, the algorithm C4.5 of decision tree become popular, but C5.0 algorithm is still undergoing. In this article, we do some deep research on C5.0 algorithm by embedding "boosting" technology in cost matrix and cost-sensitive tree to establish a new model for individual credit evaluation of Commercial Bank. We apply our new model on evaluating the individual credit records of a German bank, and compared results of the adjusted decision tree model and the original one. The comparison shows that the adjusted decision tree model is more precise.

Key words: credit evaluationdecision tree; C5.0 algorithm; boosting technology; cost-sensitive tree

1 Introduction

A lot of researches show that multivariate discrimination analysislogistic regression model, Neural Network technology and Support Vector Machine are efficient in research of company and individual credit evaluation with high accuracy rate of discrimination based on financial numeric data. But those methods can not be applied on individual credit evaluation, for its record data are nonnumeric except the "income" index, which is in numeric type. Other individual credit record data such as "gender", "Occupation" and "duty" which are nonnumeric are very important. These valuable data can not be used as the input variant in multivariate discriminate analysislogistic regression model, Neural Network technology, and Support Vector Machine.

Thus, these methods can not be applied in individual Credit Evaluation directly. As one of the most widely used algorithms within data mining, decision tree algorithm has the advantage of high accuracy rate, simpliness, high efficiency. It can handle not only the numeric data such as "income", "loans", "value of pledge", but also the nonnumeric data such as "gender", "degree", "career", "duty". So it is very suitable for individual credit evaluation of commercial banks.

The first concept of decision tree come from Concept Learning System (CLS) was proposed by Hunt.E.B et al in 1966[1−2]. Based on it, a lot of improved algorithms have emerged. Among these, the most famous algorithm is ID3 with a choosing policy according to information gain, which was proposed by Quinlan in 1986[3]. Based on ID3 algorithm, many scholars proposed improved methods. Quinlan also invented C4.5 algorithm in 1993[4]. C4.5 takes information gain ratio as the selecting criterion in order to overcome the defect of ID3, which is apt to use the attributes that have more values. At the same time, C4.5 introduces some new functions such as pruning technology. For the purpose of handling large-scale data set, some scholars proposed more new decision tree algorithms, such as SLIQ[5], SPRINT[6], PUBLIC[7], and so on. These algorithms have their own advantages and characteristics.

C5.0 is another new decision tree algorithm developed based on C4.5 by Quinlan[8]. It includes all functionalities of C4.5 and apply a bunch of new technologies, among them the most important application is "boosting"[9−10] technology for improving the accuracy rate of identification on samples. But C5.0 with boosting algorithm is still undergoing and unavailable in practice. We develop a practical C5.0 algorithm with boosting algorithm and apply on pattern recognition. The iterative process of boosting technology in C5.0 is programmed, and numerical experiments were done. The algorithm process is modified and optimized during experiments for the model of individual credit evaluation of commercial banks. We apply the model for evaluating the individual credit data from some bank in German.

In the second section, we introduce the construction of a decision tree and the purring of a tree in C4.5, then the main ideas of C5.0. After that we provide the mathematical description of the boosting that we use in this article and introduce the cost matrix and cost-sensitive tree, which we use in section 3. In the third section, by embedding boosting technology, using matrix and cost-sensitive tree we construct the model of individual credit evaluation of commercial banks based on C5.0 and evaluate the individual created data from some bank in German using this model. We also compare the discrimination result of the modified decision tree with the original one. We provide the conclusion in the last section.

Received date: July 30, 2008

∗ **Corresponding author:** Tel: +86-136-1006-1308; E-mail: pangsulin@163.com

Foundation item: Supported by the National Nature Science Foundation(No.70871055) and 2008 Ministry of Education of the People's Republic of China New Century Scholar Foundation(No.NCET-08-0615)

2 C5.0 algorithm

As mentioned earlier, C5.0 is a new decision tree algorithm developed from C4.5. The idea of construction of a decision tree in C5.0 is similar to C4.5. Keeping all the functions of C4.5, C5.0 introduces more new technologies. One of important technologies is boosting, and another one is the construction of a cost-sensitive tree. We introduce the construction of a decision tree based on C4.5, the purring technology, boosting, cost matrix, and cost-sensitive tree as following.

2.1 Decision tree and pruning technology in C4.5 algorithm

At first, we take the given samples set as the root of decision tree. Second, we compute information gain ratio of every attribute of the training samples, and select the attribute with the highest information gain ratio as the split attribute. Then, we create a node for the split attribute and use the attribute to make an indicator for the node. Third, we create a branch for each value of the split attribute and according to this, divide the data set into several subsets. Obviously, the number of the subsets equals the number of the testing attribute's values. Then we repeat the steps above for every subset we have created until all the subsets can satisfy one of the following three conditions:

(1) All the samples in the subset belong to one class. At the same time, we create leaves for subsets.

(2) All the attributes of the subset have been transacted and there are no attributes left which can be used to divide data set. In this condition, we classify this subset as the class which most samples in it belong to and create a leave for the subset.

(3) All the testing attributes left of the samples in the subset have the same value, but the samples still belong to different classes. In this condition, we create a leave whose class is most samples in the subset belong to.

C4.5 mainly adopts EBP algorithm as its pruning method. EBP algorithm[11] is shown as follows:

Assume that T_t is a subtree whose root is an inner node t . $n(t)$ is the number of samples that fall into the node t . $e(t)$ is the number of samples that fall into node t but are misclassified by the class of this node. $r(t)$ is the misclassification rate of node t. From the viewpoint of Probability, the rate of misclassified samples can be seen as the probability that some event happens $e(t)$ times during the $n(t)$ experiments, and we can get a confidence interval $[L_{CF}, U_{CF}]$ of it. Assume that CF is confidence level of the confidence interval. We can control the degree of pruning by CF : the higher the value is the fewer the branches will be pruned; the lower the value is the more branches will be pruned. In C4.5, the default value of CF is 0.25, and we assume that the misclassification rate is accord with binomial distribution.

2.2 Boosting technology

Boosting is one of the most important improvements in C5.0 compared with C4.5. Boosting algorithm sets weight for each sample, which presents its importance. The higher the weight is, the more the sample influence on the decision tree. Initially, every sample has the same weight. In each trial, a new decision tree is constructed. The weight of each sample is adjusted, such that the learner focus on the samples which are misclassified by the decision tree constructed in the last trial, which means these samples will have higher weight. In this article we take the following boosting iteration algorithm[9−10].

Assume a given samples set S consists of n samples and a learning system that constructs decision trees from a training set of samples. Boosting constructs multiple decision trees from the samples. The number T is the number of decision trees that will be constructed, which is the number of trials will be treated. C^t is the decision tree that the learning system generates in trail t, and C^* is the final decision tree that is formed by aggregating the T decision trees from these trails. ω_i^t is the weight of sample i in trail t $(i = 1, 2, ..., N; t = 1, 2, ..., T)$. P_i^t is the normalized factor of ω_i^t and β_t is the factor that adjusts weight. We also define indicator function:

$$\theta^t(i) = \begin{cases} 1, \text{ sample } i \text{ is misclassified} \\ 0, \text{ sample } i \text{ is classified rightly} \end{cases}$$

The main steps of boosting are as follows:

Step 1 Initialize the variables: set a value to the number of T (usually is 10). Set $t = 1, \omega_i^1 = 1/n$.

Step 2 Calculate $P_i^t = \omega_i^t / \sum_{i=0}^n (\omega_i^t)$, where $\sum_{i=0}^n (P_i^t) = 1$.

Step 3 Set P_i^t to the weight of each sample and construct C^t under this distribution.

Step 4 Calculate the error rate of C^t as $\varepsilon^t = \sum_{i=0}^n (P_i^t \theta_i^t)$.

Step 5 if $\varepsilon^t < 0.5$, the trails are terminated, set $T = t + 1$; else if $\varepsilon^t = 0$, the trails are terminated, set $T = t$; else if $0 < \varepsilon^t < 0.5$, go on to step 6.

Step 6 Calculate $\beta^t = \varepsilon^t / (1 - \varepsilon^t)$.

Step 7 Adjust weight according to the error rate, that is

$$\omega_i^{t+1} = \begin{cases} \omega_i^t \beta^t, \text{ sample is misclassified} \\ \omega_i^t, \text{ sample is classified rightly} \end{cases}$$

Step 8 If $t = T$, the trails are terminated. Else, set $t = T + 1$ and go to step 2 to begin the next trail.

Finally, we obtain the boosted tree C^* by summing the votes of the decision trees $(C^1, C^2, ..., C^T)$, where the vote for C^T is worth $log(1/\beta^t)$ units. That is $C^* = \sum_{t=1}^T (1/\beta^t) C^t$. This means when classify a testing sample using decision tree model, first, we classify this sample by $C^t (1 \le t \le T)$, and we can get T results. Then we count the final vote of each class according to the weight of $C^t (1 \le t \le T)$ and select the class, which has the highest vote as the final result.

2.3 Cost-sensitive tree

In C4.5 all kinds of misclassification errors are treated equally and the object of decision tree minimizes the number of misclassification errors. But many practical classification problems have different costs associated with different types of error. For example, in loan business of commercial banks, errors of recognizing a "bad" customer as "good" customer are usually considered more serious than the opposite type of recognizing a "good" customer as "bad".

The cost of misclassification presents the seriousness of some misclassification errors. Higher the cost is, more

Table 1. Sample set fields

Character	duration in month, credit history, purpose, credit amount, other installment plans, number of existing credits at this bank
Capacity	checking account, installment rate in percentage of disposable income, job, number of people being liable to provide maintenance for
Capital	savings account/bonds
Collateral	other debtors / guarantors, property
Condition	age in years, personal status and sex
Stability	present employment years, present residence years, housing, telephone registration, foreign worker or not

serious the misclassification is. So during the modelling, we should pay more attention on this type of errors and eliminate them as much as we can. Essentially, the cost of misclassification is the weight applied for some given result. This weight can be transformed into the factor in model which can change the result of the model in application. Different costs of misclassification can be showed in cost matrix. Cost matrix shows the different combinations of the predicted class and actual class.

In C5.0, misclassification cost can be set in costs matrix by a certain value. By minimizing expectation of the total misclassification costs, we construct a decision tree called cost-sensitive tree.

3 C5.0 application on individual credit evaluation of commercial banks

3.1 Data description

We use the individual credit data set of some German bank from internet[12], which includes 1000 records. Each record consists of 21 fields of which the first 20 are the description about the customer's credit information, which includes existing checking account, duration in month, credit history, purpose, credit amount, savings account/bonds, present employment years, installment rate in percentage of disposable income, personal status and sex, other debtors / guarantors, present residence years, property, age in years, other installment plans, housing, number of existing credits at this bank, job, number of people being liable to provide maintenance for, telephone registration, foreign worker or not. The last field is the definition of the customer in the bank, which includes 2 classes: "good customer" and bad "customer". We categorize the first 20 fields as Table 1:

These 20 fields are all important factors that affect the individual credit in individual credit evaluating. Considering that comprehensiveness of index choosing and the characteristics of decision tree algorithm, we adopt all these 20 filed in the indexes of the model to be built, which can be seen as the sample's characteristic attributes. The definition of "good" customer and "bad" customer are as following: "good" customer is whom the credit agent is willing to loan to. Credit agent believes "good" customer can repay debt and interest in time. "Bad" customer is whom credit agent is not willing to loan to, for the expectation of repaying debt and interest in time is low. We randomly select about 80%(799) from the 1000 samples as the training sample set, the left 20%(201) will be used as testing sample set to test the model. The distribution of the "good" and "bad" customers in the sample set is shown in Table 2:

Table 2. Distribution of "good" and "bad" customers in respective samples

Sample sets	Number/ good rate	Number/ bad rate	Good : bad
Total sample set	700/70%	300/30%	2.33 : 1
Training sample set	558/69.84%	241 / 30.16%	2.32 : 1
Testing sample set	142/70.65%	59/29.35%	2.41 : 1

First, we compare the distribution of the attributes value in training sample set with ones in total sample set, and find out that they are almost the same. So we can conclude that the training sample set reflects the characteristics of total sample set. Second, according to the experience of previous research, the number of bad customers should reach some amount and the number of good customers should not be smaller than the number of bad customers in the training sample set. We can see from the table above that the ratio of "good" customers to "bad" customers in training sample set is 2.32:1, which is almost the same one in total sample set. And the number of "bad" customers in training sample set reaches 80% which account for of the total number of "bad" customers. Thus, based on these two points above, we can conclude that the current training set meets the requirements of Modelling.

3.2 Data conversion and storage

We label each field name in the form of "Letter + number" and save them in computer. The final information of the sample data set for modelling is shown table in appendix.

3.3 Modelling of decision tree

According to C5.0 algorithm, we create a new node which is the root for the original sample set. Then we calculate the gain ratio of each characteristic attribute (C1-C21). First, we introduce the following definitions.

Definition 1 (Information Entropy): Assume a given set S consists of n samples. The class attribute C has m different values $C_i(i = 1, 2, ...m)$. Each sample belongs to one of the m classes. Let n_i be the number of class C_i samples. Then the information entropy $E(S)$ needed for classifying S is defined as follows:

$$E(S) = -\sum_{i=1}^{m} p_i log_2^{p_i} \tag{1}$$

Where p_i denotes the proportion of the number of class C_i samples to the number of all samples in the training set. We can usually calculate it by $p_i = n_i/|S|$. Where $|S|$ denotes the number of samples in set S, $|S| = n$.

Table 4. Classification result of training sample set by primary decision tree model

Sample sets	Classified as good	Classified as bad	Accurate rate	Error rates
Good	528	30	94.62%	5.38%
Bad	87	154	63.90%	36.10%
Total	615	184	85.36%	14.64%

Table 5. Classification result of testing sample set by primary decision tree model

Sample sets	Classified as good	Classified as bad	Accurate rate	Error rates
Good	120	22	84.51%	15.49%
Bad	42	17	28.81%	71.19%
Total	162	39	68.16%	31.84%

Definition 2 (Conditional Information Entropy): Assume that attribute A has ν different values $(a_1, a_2, ..., a_\nu)$, which divide the set S into ν subsets $(S_1, S_2, ..., S_\nu)$. Let n_{ij} be the number of class C_i samples in the subset S_i . Then the conditional entropy of attribute $E(S|A)$ is defined as follows:

$$E(S|A) = -\sum_{j=1}^{\nu} p_i^{'} \sum_{i=1}^{m} p_{ij} log_2(p_{ij}) \tag{2}$$

where $p_i^{'}$ denotes the proportion of the number of samples whose values of attribute A are a_j to the number of all samples in the training set, $p_j^{'} = |S_j|/|S| = \sum_{i=1}^{m} n_{ij}/n$.

p_{ij} is the conditional probability that sample whose value of attribute A is a_j belongs to class C_i, $p_{ij} = n_{ij}/|S_j|$. $|S_j|$ is the number of samples whose values of attribute A are a_j, $|S_j| = \sum_{i=1}^{m} n_{ij}$.

Definition 3 (Information Gain Ratio): Information gain of attribute A is defined as:

$$Gain(A) = E(S) - E(S|A) \tag{3}$$

Definition 4 Information Gain Ratio Information gain Ratio can be obtained further as follows:

$$GainRatio(A) = Gain(A)/Split(A) \tag{4}$$

Where $Split(A) = -\sum_{j=1}^{\nu} p_j^{'} log_2(p_j^{'})$.

Then we calculate $Gainratio(C1)$ of the characteristic attribute C1, the process is as following.

Step 1 We calculate the Information Entropy of the training sample set S according to the Formula (1). The training sample set comprises of 799 samples, so $n = 799$; These samples can be classified into 2 types: good customer and bad customer, so $m = 2$.

The number of good customers is 558 and the number of bad customers is 241, so $n_1 = 588 n_2 = 241$, $p_1 = 588/799$, $p_2 = 241/799$. Then we gain the following result:

$E(S) = -\sum_{i=1}^{m} p_i log_2(p_i) = -588/799 log_2(588/799)$ $-241/799 log_2(241/799) = 0.883$

Step 2 Calculate the Conditional Information Entropy of C1 according to Formula (2). Attribute C1 has four discrete values A11,A12,A13,A14 in all. So $\nu = 4$. There are totally 222 samples when $C1 = A11$. Of these samples, there are 115 samples whose decision attribute $C2 = 1$, there are 107 samples whose decision attribute $C21 = 2$. (That is, 115 good customers and 107 bad customers). So $p_1^{'} = 222/799$, $p_{11} = 115/222$, $p_{21} = 107/222$. There are totally 216 samples when $C1 = A12$. Of these samples, there are 130 samples whose decision attribute $C21 = 1$ and 86 samples whose decision attribute $C21 = 2$. Thus $p_2^{'} = 216/799$, $p_{12} = 130/222$, $p_{22} = 86/222$. There are totally 49 samples when $C1 = A13$. Of these samples, there are 39 samples whose decision attribute $C21 = 1$ and 10 samples whose decision attribute $C21 = 2$. So $p_3^{'} = 49/799$, $p_{13} = 39/49$, $p_{23} = 10/49$. There are totally 312 samples when $C1 = A14$. Of these samples, there are 274 samples whose decision attribute $C21 = 1$ and 38 samples whose decision attribute $C21 = 2$. So $p_4^{'} = 312/799$, $p_{14} = 274/312$, $p_{24} = 38/312$. Then we can get the following result:

$$E(S|C1) = -\sum_{j=1}^{\nu} p_i^{'} \sum_{i=1}^{m} p_{ij} log_2(p_{ij}) = 0.79324$$

Step 3 Calculate Information Gain of C1 according to Formula (3):

$$Gain(C1) = E(S) - E(S|C1) = 0.08976$$

Step 4 Calculate Information Gain Ratio of C1 according to Formula (4)

$$GainRatio(C1) = Gain(C1)/Split(C1) = 0.0499$$

Similarly, we can calculate Information Gain Ratio of the other 19 decision attributes. We set the attribute C1 whose Information Gain Ratio is the largest and create a node with 4 branches according to the 4 possible values for C1. Then the original training sample set is separated into 4 subsets, and we create a new node for each subset whose decision attribute is A11, A12, A13, and A14, respectively. Then for each new node, we repeat the above steps until all the nodes satisfy the 3 stopping conditions that is described in section 2.1. At the end, a construction of decision tree is done. Then we prune this new tree using EBP method described in section 2.1 and get a simpler decision tree. The modelling is down.

Considering that information of reality problems is massive, we use SPSS Clementine to construct a decision tree model. During modelling the decision tree, all parameters relative to C5.0 model are set default. The following 2 tables show the classification result of the training sample set and testing sample by our built decision tree model.

From the tables above, we can see that the model is acceptable. But it can be optimized furthermore as we do in the following.

Table 6. Classification error rates of decision tree models with different values of COST(A)

COST(A) values	1	2	3	4	5
Training set					
Total error	14.64%	14.52%	15.14%	25.41%	25.66%
Type A error	36.10%	11.62%	2.10%	1.66%	1.66%
Type B error	5.37%	15.77%	20.79%	35.66%	56.30%
Test set					
Total error	31.84%	30.35%	39.80%	41.79%	43.28%
Type A error	71.20%	32.20%	40.68%	28.81%	30.50%
Type B error	15.50%	29.58%	39.44%	47.18%	48.60%

3.4 Cost matrix

We name the classifying a good customer as a bad customer as "error of type A" and the classifying a bad customer as a good customer as "error of type B". The two rate calculate respectively by:

Error rate of type A = number of type A / total number error rate of type B = number of type B / total number.

We label the cost of error in type A as COST(A) and the cost of error in type B as COST(B). In the procedure of building the model, we should eliminate the errors in type A on the premise of ensuring the high accuracy of classifying the total samples.

We get a suitable cost matrix by many experiments and comparing the results. We set 1 to COST(B) and a value which is larger than 1 to COST(A), on the condition that all the other parameters of the model are default. Then we increase the value of COST(A) continually, through several times of experiments, build up different decision tree models and select the best value of COST(A) according to the classification result of the models. Our selection criteria are as following:

Condition 1 For the training sample set and testing sample set, the total error rate should not be higher than other models with different values of COST(A). And the lower the total error rate is, the better the model is.

Condition 2 On the premise that the total error rate satisfy the criterion, the lower the error rate in type A is, the better the model is.

Condition 3 Although 2 model's error rate for the training sample set is similar, we should give the top priority to the error rate for the testing sample set. That is selecting the model, which shows better performance for the training sample set. Table 6 shows the classification error rates of the decision tree models with different values of COST(A).

Investigating the data in Table 6, we can see that along with the increase of COST(A), the total error rate of the model is increasing, both for the training sample set and the testing sample set. Whereas, the error rate in type A shows the trend of decrease.

Table 7. Cost matrix

Sample sets	Classified as good	Classified as bad
Good	0	1
Bad	2	0

In the following, we will find the best value of COST(A) according to our criteria mentioned earlier.

(1) If $COST(A) > 3$, the error rate of the model is higher than 25% for the training sample set and higher than 40% for the testing sample set, apparently a bit higher. So it does not satisfy us, and it is not reasonable to set a value bigger than 3 for COST(A).

(2) Comparing the situations when COST(A) is set to 1,2, and 3, we can see in evidence that when $COST(A) = 2$, not only the total error rate is at the lowest, but also the error rate in type A is at the lowest. So $COST(A) = 2$ is our appropriate choice. Then we can get the cost matrix as the following Table 7.

3.5 Pruning degree

In C4.5 and C5.0, pruning degree of the decision tree is controlled by the value of CF. But in the C5.0 model of SPSS Clementine, it is controlled by another index called "gravity of pruning". It can be calculated by $gravityofpruning = (1-CF)*100$, and the default value is 75 (it means the default value of CF is 0.25).

We select the number of the nodes as the measure of complexity of a decision tree. On the base of the decided cost matrix in last section, we give different values to the "gravity of pruning", and observer the error rate and complexity of the sequence of constructed decision trees. For preventing the model's being over-fit, simplifying the model and ensuring the accuracy, the selection criteria of "gravity of pruning" should include not only the three rules in section 2.1, but also another rule as following.

We should select the value of "gravity of pruning" which makes the constructed decision tree have less nodes (with lower complexity) when the other conditions are similar.

Investigating the data in Table 8 below, we can find that the nodes of the decision tree is decreasing and the tree become less complex with the increase of gravity of pruning. Meanwhile, the total error rate and error rate in type A is increasing in the training sample set. We analyze the result in the testing sample set as following:

(1) If gravity of pruning < 70 , the total error rate and error rate in type A both stay in a high level contrast with the very low total error rate in training sample set. It means that the decision tree model is "over-fit" because of not being pruned enough. It leads low accuracy in testing sample set contrast with high accuracy in training sample set.

(2) If gravity of pruning is between 70 and 80, the total error rate and error rate in type A in the testing sample set decline to the lowest level and the error rate in training sample set stay in a middle level.

(3) If gravity of pruning > 85, the total error rate and error rate in type A in the testing sample set begin to go

Table 8. Number of nodes and classification error rates of decision trees with different pruning

gravity of pruning	60	65	70	75	80	85	90	95
Number of nodes	211	187	145	125	118	90	64	56
Training set								
Total error rate	6.88%	8.76%	12.52%	14.52%	14.89 %	18.40%	19.65%	21.03%
Type A error rate	4.15%	4.98%	6.64%	11.62%	14.11%	14.52%	24.48%	24.07%
Test set								
Total error rate	36.82%	37.81%	36.82%	30.35%	29.85%	31.34%	30.35%	30.85%
Type A error rate	42.37%	42.37%	33.90%	32.20%	32.20%	32.20%	38.90%	39.98%

back up. And the error rate in training sample set ascends to the top. This means the decision tree is over pruned which makes it as low degree of fitting for the training sample set. The accuracy can not meet the requirement.

Based on the three points above, the value of "gravity of pruning" to 75 or 80 is suitable to our selection criteria. Because the error rates in training sample set and testing sample set of these two models are similar, we select the less complex decision tree model with 118 nodes according to our fourth rule. The "gravity of pruning" is set 80.

3.6 Model optimization by boosting

As an important technology in C5.0, Boosting increase the accuracy of the decision tree model effectively. But in practice, modelling depends on the training sample set. Any method, which can increase the accuracy of model has the risk of making the model "over-fit". To observe the effect of booting, we investigate performance of boosting in the procedure of modelling on determined model parameters above. The iteration times is set to the default value 10.

According to the description of boosting in section 2.2, the training sample set comprises of 799 samples, so $n = 799$. Here we set iteration times to 10, that is $T = 10$. The first iteration procedure is as following:

Step 1 In the first time of iteration, the original values of weight of all the samples are set as $\omega_i^1 = 1/n = 1/799$, $(i = 1, 2..., 799)$.

Step 2 Calculate the normalized weight $p_i^1 = \omega_i^t / \sum_{i=1}^{n} \omega_i^t = 1/799$, $(i = 1, 2..., 799)$. So in the first time of iteration, all the samples' weights are the same.

Table 9. Error rate of decision tree model with boosting

Train set error rate	4.63 %
Train set type A error rate	0.80%
Train set type B error rate	6.27%
Test set total error rate	31.34%
Test set type A error rate	37.29%
Test set type B error rate	28.87%

Step 3 Set the normalized weight $1/799$ to each sample, and construct the first decision tree model C^1 under this distribution.

Step 4 Calculate error rate ε^1 of C^1 . The number of the samples classified by C^1 correctly is 680 while the number of the samples misclassified by C^1 is 119. And all the samples have the same weight $1/799$ in the first time of iteration. So $\varepsilon^1 = 199/799 = 0.249$.

Step 5 Because $\varepsilon^1 < 0.5$, go on to calculate $\beta^1 = \varepsilon^1/(1-\varepsilon^1) = 0.3316$.

Step 6 Change the weight of samples for the next iteration.

$$\omega_i^2 = \begin{cases} \omega_i^1 \beta^1 = 4.15 * 10^{-4}, \text{ sample is classified rightly} \\ \omega_i^1 = 1.25 * 10^{-3}, \text{ sample is misclassified} \end{cases}$$

The first iteration ends.

After finishing the first iteration, we use the same method to calculate the normalized weight in the next iteration procedure. The whole boosting procedure include times of iteration, and constructs 10 decision trees $C^1, C^2, ...C^{10}$. While classifying the samples, we give the 10 weight $log(1/\beta^t)$ to each decision tree and then determine the final classification result through voting.

Likely, in practical calculation, we use SPSS Clementine for iteration and modelling. The classification result of the constructed decision tree model with booting is showed in the following Table 9.

Compared with the data in the column in which "gravity of pruning" is 80 of Table 7, after boosting, the total error

Table 10. Classification of being adjusted parameter in train set

Sample sets	Classified as good	Classified as bad	Total	Accurate rate	Error rates
Good	473	85	558	84.77%	14.23%
Bad	34	207	241	85.89%	14.11%
Total	507	292	799	85.11%	14.89%

Table 11. Classification of being adjusted parameter in test set

Sample sets	Classified as good	Classified as bad	Total	Accurate rate	Error rates
good	101	41	142	71.13%	28.87%
bad	19	40	59	67.80%	32.20%
total	120	81	201	70.15%	29.85%

rate and error rate in type A of the training sample set declines clearly, from 14.89% to 4.63% and from 14.11% to 0.8%, respectively. But in the testing sample set, both the total error rate and error rate in type A increase, from 29.85% to 31.34% and from 32.2% to 37.29%, respectively. This shows that boosting technology can increase the fitting degree of the decision tree model effectively, which makes the model's accuracy in training sample set improved evidently. But the accuracy in the testing sample set is still.

3.7 Classification of being adjusted parameter

The classification results of the modified model in the training sample set and testing sample set are showed in Tables 10 and 11.

We can compare the model after parameter adjusting with the primary model constructed in section 3.3 and get the flowing Table 12.

The parameter adjusted model is showed in the following figure.

From Table 12 we can see that, in the training sample set, the total error rate of adjusted model is similar to that of the primary model. But error rate in type A of the adjusted model declines by 21.99%. In the testing sample set, compared with the primary model, the total rate of adjusted model declines by about 2%, and the error rate in type A declines by 38.99%. This result satisfies our modelling object.

Table 12. Error rate of decision tree model with boosting

	Primary model	Parameter adjusted model	Difference
Train set Total Error rate	14.64%	14.89%	-0.25%
Train set type A Error rate	36.10%	14.11%	21.99%
Train set type B Error rate	5.38%	15.23%	-9.85%
Test set Total Error rate	31.84%	29.85%	1.99%
Test set type A Error rate	71.19%	32.20%	38.99%
Test set type B Error rate	15.50%	28.87%	-13.37%

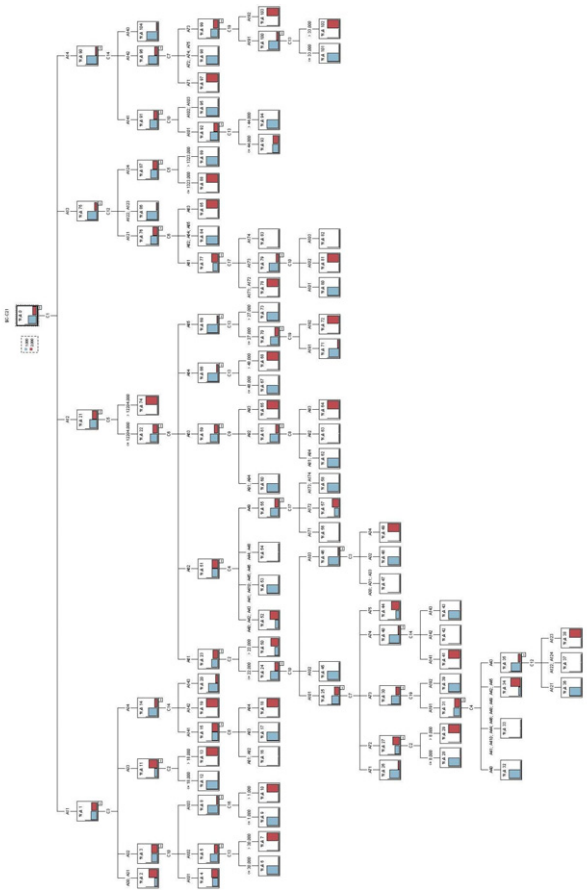

Figure 1. Parameter adjusted decision tree model

4 Conclusions

This article introduces C5.0 algorithm based on C4.5, boosting technology and the construction of decision tree, and constructs the Individual credit evaluation model of commercial banks based on C5.0 algorithm, then does empirical test using the individual credit data from a German bank. By setting misclassification cost, selecting pruning degree and analyzing the practical effect of boosting, we construct a decision tree model based on C5.0. We conclude that our model based on C5.0 algorithm is high accuracy, low cost of risk and high controllability compared with other model in practice.

References

[1] Hunt E B, Krivanek J. The effects of pentylenetatrazole and methyl-phenoxy propane on discrimination learning. Psychopharmacologia, 1966(9): 1–16.

[2] Ji G S, Chan P L, Song H. Study the survey into the decision tree classification algorithms rule. Science Mosaic, 2007(1): 9–12.

[3] Quinlan J R. Induction of decision trees. Machine Learning, 1986(4): 81–106.

[4] Quinlan J R. C4.5: Programs for machine learning. Machine Learning, 1994(3): 235–240.

[5] Mehta M, Agrawal R, Rissanen J. SLIQ: A fast scalable classifier for data mining. In Proceedings International Conference on Extending Database Technology, 1996: 18–32.

[6] Shafer J, Agrawal R. A scalable parallel classifier for data mining. Proceedings of 1996 International Conference on Very Large Data Bases, 1996(9): 544–555.

[7] Rastogi R, Shim K. Public: A decision tree classifier that integrates building and pruning. Proceedings of 1998 International Conference on Very Large Data Bases, New York, 1998: 404–415.
[8] Quinlan J R. “C5”. http://rulequest.com, 2007.
[9] Quinlan J R. Bagging, Boosting and C4.5. Proceedings of 14th National Conference on Artificial Intelligence, 1996: 725–730.
[10] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 1997(1): 119–139.
[11] Fan J, Yang Y X. Researches on methods for postpruning decision trees. Journal of Hunan Padio and Febevision University, 2005(1): 54–56.
[12] “UCI Machine Learning”. http://www.ics.uci.edu/mlearn/MLRepository.html, 2007.

Appendix

Table 3. Information of fields in sample data set after data processing

No	Field name	Type	Values	Description
C1	checking account	discrete	A11, A12, A13, A14	A11:<0 DM ; A12:<= 200 DM; A13:>= 200 DM; / salary assignments for at least 1 year; A14 : no checking account
C2	duration in month	continuous	[4,72]	
C3	credit history	discrete	A30, A31, A32, A33, A34	A30 : no credits taken/ all credits paid back duly A31 : all credits at this bank paid back duly A32 : existing credits paid back duly till now A33 : delay in paying off in the past A34 : critical account/ other credits existing (not at this bank)
C4	purpose	discrete	A40, A41, A42, A43, A44, A45, A46, A48, A49, A410	A40 : car (new) A41 : car (used) A42 : furniture/equipment A43 : electrical appliance A44 : domestic appliances A45 : repairs A46 : education A48 : retraining A49 : business A410 : others
C5	credit amount	continuous	[250,18424]	
C6	savings account/bonds	discrete	A61, A62, A63, A64, A65	A61 : < 100 DM; A62 : >=100 DM AND < 500 DM; A63 :>=500 DM; AND < 1000 DM; A64 : >= 1000 DM; A65 : unknown/ no savings account
C7	present employment years	discrete	A71, A72, A73, A74, A75	A71 : unemployed; A72 : < 1 years; A73 : >=1 years AND ¡ 4 years; A74 : >=4 years AND < 7 years; A75 : >= 7 years
C8	ratio of installment to income	discrete	A81, A82, A83, A84	A81:<10%; A82: >=10% AND <20%; A83: >=20% AND <40%; A84: >=40%
C9	personal status and sex	discrete	A91, A92, A93, A94	A91 : male : divorced/separated; A92 : female; A93: male : single; A94 : male : married
C10	other debtors / guarantors	discrete	A101, A102, A103	A101 : none; A102 : co-applicant; A103 : guarantor
C11	present residence years	discrete	A111, A112, A113, A114	A111:>= 10 years; A112:<10 years AND >=6 years; A113:<6 years AND >=2 years; A114:<2 years
C12	property	discrete		A121 : real estate; A124 : unknown / no property; A122 : if not; A121 : building society savings agreement/ life insurance; A123 : if not A121/A122 : car or other
C13	age in years	continuous	[19,75]	
C14	other installment plans	discrete	A141, A142, A143	A141 : bank; A142 : stores; A143 : none
C15	housing	discrete		A151 : rent; A152 : own; A153 : for free
C16	present credits record	continuous	[1,4]	
C17	job	discrete	A171, A172, A173, A174	A171 : unemployed/ unskilled - non-resident; A172 : unskilled - resident; A173 : skilled employee / official; A174 : management/ self-employed/ highly qualified employee/ officer
C18	number of person to support	continuous	[1,2]	
C19	telephone registration	discrete	A191,A192	A191 : none; A192 : yes, registered under the customers name
C20	foreign worker or not	discrete	A201,A202	AA201 : NO; A202 : YES
C21	class	discrete	1,2	1: good customer 2bad customer