Open Access

On the Application of a New Method of the Top-Down Decision Tree Incremental Pruning in Data Classification

Shao Hongbo[*,1], Zhou Jing[1] and Wu Jianhui[2]

[1]*College of Science, Agricultural University of Hebei, Baoding, China*

[2]*Hebei Province Key Laboratory of Occupational Health and Safety for Coal Industry, Division of Epidemiology and Health Statistics, School of Public Health, Hebei United University, Tangshan, China*

Abstract: Decision tree, as an important branch of machine learning, has been successfully used in several areas. The limitation of decision tree learning has led to the over-fitting of the training set, thus weakening the accuracy of decision trees. In order to overcome its defects, decision trees pruning is often adopted as a follow-up step of the decision trees learning algorithm to optimize decision trees. At present the commonly-used decision tree sample is based on statistical analysis. Due to the lack of samples, the small training set is less statistical, and it leads pruning methods to failure. Based on the previous research and study, this paper has presented a top-down decision tree incremental pruning method (TDIP), which applies the incremental learning to the comparison between the certainty and uncertainty rules so that only the former remains. In addition, to speed up the process of its pruning, a top-down search is defined to avoid the iteration of the same decision tree. The top-down decision tree incremental pruning method (TDIP) is independent of statistical characteristics of the training set. It is a robust pruning method. The experimental results show that the method maintain a good balance between accuracy and size of pruned decision trees, and is better than those traditional methods in classification problems.

Keywords: Decision trees, Incremental learning, Overfitting, Post-pruning, TDIP, Iteration.

1. INTRODUCTION

With the widespread use of computer, numerous data have been stored in the form of electronic signal. Closely connected with our life, these data will be very valuable, if they can be correctly used. And Machine learning is a general term of a set of data processing methods, which can automatically build models that can describe data set's core structure [1].

The models built by machine learning have two significant applications. First, it will predict the property of unknown data if it can precisely describe the data set's structure. Second, it could be used to further analyze this data set in more fields when it can conclude the core information of data set in the way people can easily understand [2].

The two applications are not independent. In order to analyze more efficiently, models that can made an accurate indication of the field of the data set are necessary for an efficient analysis while a good form is needed to improve the accuracy of prediction. On the other hand, some models, which were made simply for predicting, cannot be directly applied to the process of field predicting [3].

In the actual data mining, the second method cannot be fully understood or be connected with field knowledge, so with serious disadvantages, this black-box program cannot be applied to some fields that require human field experts to make decisions.

Decision tree is not only a potential predicting tool but a comprehensive structure of data set. Compared with other complicated models, it grows faster. Thanks to its advantages in prediction accuracy and domain analysis, it has been a widely-used data-mining tool. Model's prediction accuracy and domain analysis relies on machine learning to generalize the structure's complexity out of data sets. Without considering prediction accuracy, the model should be as simple as possible since its scale can directly affect the complexity of the interpretation model. Besides, it is also very important to ensure that the model does not describe false structure (which means the structure looks real, but in fact it has nothing to do with the domain)

The false structure emerges since a data set is formed by a set of finite randomization data with uncertain distribution and the random sampling process cannot guarantee that the structure that the model describes is true to the field's real condition. Unless data can be processed previously, machine learning could produce false structures, which often cause the decrease of the model's intelligibility and classification accuracy. In order to eliminate unnecessary complexity, a judgment mechanism is needed, which can determine when the accidental false structures would appear, and how to eliminate it. Finding and removing the false structure in the model is called pruning, which is to remove the unnecessary part. In the process of pruning, the scale of model would decrease with a higher prediction precision. In the real world, due to the data set's characteristics or the improper methods of building data set, data set includes noise unavoidably.

*Address correspondence to this author at the College of Science, Agricultural University of Hebei, Baoding, China; Tel: +86-13582903299; E-mail: Hongbo13582903299@qq.com

1874-4443/15 2015 Bentham Open

Therefore, pruning has gradually become an important part of machine learning algorithm [4].

The quality of pruning would affect the scale and prediction precision of the final model. Ideally, pruning would only remove the unnecessary part of the model that is produced by the noise without changing the structure that can actually reflect results. But this ideal model needs to be built manually by field experts. Therefore, the core purpose of pruning is to build a method that can "determine whether each part of the model would reflect the database's real condition by analyzing the given database".

Machine learning is a method that is based on the data derived classifier, which mainly concerns the classifier's prediction precision on unknown test samples. However, in many actual applications, the data prediction structure of classifier is expected to be comprehensible, which means its prediction can be explained. In the fields of machine learning and statistics, decision trees derivation as a classification method has been widely studied. Many simplification algorithms can produce more simplified or smaller trees. In this situation, the trees that are more simplified and smaller are assumed to be easily understood.

Pruning is a method that is widely applied in the field of decision tree optimization. When the classification accuracy remains the same, pruning can optimize the structure of decision tree, reduce the scale of decision tree, and improve the comprehensibility of decision tree, by replacing the node that is the root node of the subtree with most common label mark. This article has proposed a Top-down Decision Tree Incremental Pruning Method (TDIP), which can detect the rules' stability of decision tree and remove the unstable rules of it by adding unknown test samples to the non-leaf nodes. And finally, it can complete decision tree pruning operation and achieve the purpose of pruning. According to the experimental results of actual sample library in the UCI database, the optimized decision tree has efficiently reduced accidental rules that are caused by overfitting, and decision tree can have better classification with smaller scale.

2. DECISION TREE AND ITS LEARNING ALGORITHM

Over many years' studying, machine learning has developed into many different classifier construction methods, among which the most common is induction learning algorithm. Induction learning algorithm's core thought is to derive the knowledge out of the known data information. Decision tree study algorithm is a branch of induction learning algorithm and the classification models that are built by decision tree study can be defined as a set of rules' collection. At the end of 20th century, with the wide application of computer technology, the needs of processing and managing data increased continually. In order to fully exploit data resource, data mining technology has been introduced to assist data analysis. Decision tree, as a basic data mining technology, has provided solutions for building decision-making system and solving decision-making problems.

The decision tree in this article is built on the Decision Tree Algorithm ID3 proposed by J. R. Quinlan in 1986. Different from other classifiers, such as sample-based learning, neural network, and Bayesian network, the indication of what is learnt in decision making is easily comprehensible. In a standard decision tree indication, a simple operation comparison can predict samples' classification and reach an intuitive determination, which does not even require numerical value arithmetic. Simply put, the whole prediction process of decision tree can be explained.

A standard decision tree algorithm, such as ID3, C4.5, and CART, is completed by recursively partitioning test samples collection. The partition process starts with the beginning of all test samples' root nodes. In order to study the node algorithm, we need to choose a partition standard at first, and then divide the node into several child nodes. The partition process is recursive to each child node, until child node is pure, or its purity cannot be improved by further partition. The node's purity means the samples that are contained by the node with the same classification. It is hoped that decision tree algorithm can achieve this purpose with least partition operation, so that the decision tree can have better generalization ability with smaller scale. From this, it can be seen that the decision of partition standard is a key to decision tree algorithm. So far researchers have proposed many methods as to how to choose partition standard, including information gain (Quinlan), Gini index (Breiman, etc.), and gain ratio (Quinlan), by which a node's purity is measured and accordingly the best node partition standard is decided.

By exploiting decision tree, the classification process of unknown samples starts with the root node. For every decision-making node, decision tree tests its value of corresponding property in unknown samples. Based on the property's value, classification process would be continued on its decision-making node's sub-tree. It's a top-down process until it comes to the leaf node of decision tree, whose corresponding classification label is the classification result of unknown samples based on decision tree algorithm.

Currently, common decision tree algorithm varies, such as CART, ID3, C4.5, CHAID. In spite of the difference in implement method of building decision tree, their core concepts are basically the same. Seen from the perspective of hypothesis space search, a top-down decision space searching is used, which belong to the universal induction learning method. From the perspective of set, they share a process of gradually growing decision tree leaves through partitioning the original learning samples set.

From the aspect of partition set, decision tree is a partition tree. Although simple partition tree can have the structure information of decision tree, it cannot classify the unknown samples due to the reason that it does not have the decision-making information of sample classification. Therefore, we need to deduct a decision tree formed with decision-making information out of the partition tree. In the ID3 algorithm of decision tree, each test attribute in each partition is defined as a decision. It means each partition node in the partition tree respectively has an attribute of training samples, and each leaf node a classification of training samples.

Hence, in the ID3 algorithm of decision tree, node partition method has its significant effect on the building of the whole decision tree. As the standard of measuring node partition strategy's quality, ID3 algorithm adopts the much-used idea of metric entropy.

Assuming that $E = D_1 \times D_2 \times \cdots \times D_n$ is n dimension's vector space, in which D_i is a set of limited discrete variables. In this n dimension vector space, E stands for sample space, in which the element $e = (v_1, v_2, \cdots, v_n)$ $v_i \in D_i, (i = 1, 2, \cdots, n)$ is a sample. Through sample classification, sample space E can be divided into positive sample set P and negative sample set N. When a subset's elements in sample space E are either positive samples or negative ones, this subset would be considered as being pure. When a subset includes both positive and negative samples, the purity of this subset would be measured by the concept of entropy in informationism.

Definition: if a set t includes p positive samples and n negative samples, it's $\mathrm{Entropy}(t)$ should be:

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n} \tag{1}$$

The following graph shows the variation curve of set t's entropy function from 0 to 1 (Fig. **1**).

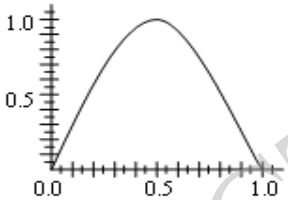

Fig. (1). $\mathrm{Entropy}(t)$'s function figure.

The partitioning node in ID3 algorithm is to improve node's purity. Therefore based on the entropy function, information gain of metric is further given to measure partition attribute's quality:

$$Gain(t,a_i) = Entropy(t) - \sum_{v \in Value(a_i)} \frac{|t_v|}{|t|} Entropy(t_v) \tag{2}$$

In this formula, $Gain(t,a_i)$ indicates the information gain that is produced by using attribute a_i to partition set t and it is the decrease of entropy's desired value which is caused by using attribute a_i to partition set t.

The information gain defines the partition attribute's classification ability. More information gain indicates that this partition attribute's ability is stronger in classifying different samples; less information gain indicates that this partition attribute's ability is weaker in classifying different samples. In each decision tree's growing process, which is also the process of partitioning sample collections, ID3 chooses to use the partition attribute with the utmost information gain as test attribute so that it can classify test samples to the maximum and obtain pure sample set as soon as possible. When the elements of one test sample subset all belong to one classification, the partition to this subset should be stopped. When all the training sample subset cannot be partitioned any further, the growing process of decision tree ends with it. In the whole partition process, algorithm maintains each test attribute decided by partitioning. It also maintains the ordinal relations among these test attributes. Because the growing process will stop when its leaf node's entropy is 1, so the corresponding classification of each leaf node is the most common classification of leaf nodes.

3. TOP-DOWN DECISION TREE INCREMENTAL PRUNING ALGORITHM

The decision tree algorithm is based on node's purity and its generalization ability would be decreased when there is noise in its training sample. This kind of algorithm would separate noise gradually as an independent branch of decision tree. Affected by these random branches, decision tree's classification accuracy of unknown samples would be decreased, as is often the case. In order to eliminate this influence, a pruning process is usually added to its decision tree algorithm to find and cut these branches and the sub-trees that are connected with them.

J. R. Quinlan used pruning algorithm that does not need extra pruning set. In the process of its whole growing and pruning, always use the same training set. The resulting error classification number is the number of mistakes in the training set classification. The error classification number completely fitted with the training sample, not on a select optimal basis after pruning the decision tree. Therefore, J.R. Quinlan used continuous correction methods for the introduction of the binomial distribution to obtain a more objective error classification number. After defining the nodes, the misclassification rate of training set of the new decision tree is:

$$r(t) = e(t)/n(t) \tag{3}$$

The misclassification rate of decision tree:

$$r(T_t) = \frac{\sum_{s \in T_t} e(s)}{\sum_{s \in T_t} n(s)} \tag{4}$$

Continuous correction method by using the binomial distribution:

$$r'(t) = \left[e(t) + \frac{1}{2} \right] / n(t) \tag{5}$$

The decision tree misclassification rate:

$$r'(T_t) = \frac{\sum_{s \in T_t} \left[e(s) + \frac{1}{2} \right]}{\sum_{s \in T_t} n(s)} \tag{6}$$

Using the wrong classification number to replace the misclassification rate:

$$e'(t) = \left[e(t) + \frac{1}{2} \right] \tag{7}$$

$$e'(T_t) = \sum_{s \in T_t} \left[e(s) + \tfrac{1}{2} \right] \qquad (8)$$

The $e'(T_t)$ represents a measure of the complexity of the decision tree, that is the complexity associated with each leaf node. Because of over fitting phenomenon, the number of pairs tree misclassification is estimated to be overly optimistic. Therefore, J. R. Quinlan weakens the conditions, only request:

$$e'(t) < e'(T_t) + SE(e'(T_t)) \qquad (9)$$

In the formula:

$$SE(e'(T_t)) = \left[e'(T_t) \times \left(n(t) - e'(T_t)/n(t) \right) \right]^{\frac{1}{2}} \qquad (10)$$

$SE(e'(T_t))$ called the standard deviation of subtree Tt. When the error classification number meets the binomial distribution, it needs to calculate the standard deviation. This algorithm is applied to the top-down. If the upper leaf node is cut off, it contains subtree which do not need to be tested again. So compared with other methods, this pruning method is faster.

But decision tree algorithm is based on the partitioning training sample collections. Such decision tree algorithm as ID has limitations. First, it is difficult to learn nonlinear programming due to its learning length, high cost and low efficiency. Second, with learning going further, there will be less learning samples and training sample collection less statistical with a higher probability of creating contingency rules.

The distribution of samples in training sample collection is a key element to decide the quality of decision tree. Samples that are provided for decision tree learning should be ideally unlimited, but in the actual application, the number of samples for decision tree learning is always limited, and some of samples would play a role as decision tree's view of learning. The more the samples in training sample collection, the broader the decision tree's view of learning. With more decision tree learning, its view of learning becomes narrow. The view of decision tree's root node is a collection of original training samples. With the constant growth of decision tree, original sample collection would be constantly partitioned and in the lower level of subtree, its view is the subset of original training samples. Although the origin of decision subtree learning does not change, its view of learning on root node becomes narrow largely. The change of learning view brings up new rules in this narrow view, which accounts for the over-fitting of decision tree. In other words, the change of the learning view increases the chance that overfitting of decision tree appears. According to the probability theory and mathematical statistics, there is a certain premise for the appearance of contingency: If the premise has changed, the rules would be broken. In classifying, a rule's stability is measured with the used collection derived by changing rules. Through the decision tree algorithm, it's seen that the stability of decision tree rules is mainly reflected on an attribute a_i's information gain in sample collection t. Therefore, a conclusion can be drawn that the rules' stability of decision tree is decided by entropy of training sample collections. By further observing the definition of entropy (Fig. **2**).

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n},$$

It is concluded that what affects the stability of rules is the number of positive and negative samples and their proportion.

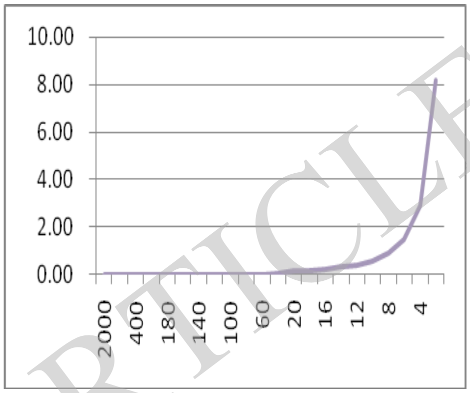

Fig. (2). The percentage of entropy function's change.

Fig. (**2**) shows that when training sample collection includes fewer samples, which means that the view of decision tree algorithm has been narrowed to a certain level, the change of entropy function becomes dramatic and the influence on the information gain becomes stronger. A stable rule should be a rule that is contained in training sample collections, where the rule's classification ability will not be affected by the number of training samples, and its information gain's change is proportional to the entropy of training sample collection. On the contrary, a contingency rule is an inaccurate rule that is indicated on training sample collections, where any change of the number of training samples can give rise to a new contingency rule. The application of original rules in classifying new training sample collections will bring less information gain. When the proportion of positive samples and negative samples in original training sample collections remains the same, some new training samples are added to original training sample collection, which broadens decision tree algorithm's current view and also increases the number of samples so that the change of entropy function would be more stable.

Prior to the addition of new training samples to original training sample collections, it should be noticed that each division of decision tree's nodes is based on certain premises. For example, in Fig. (**3**), only when the condition = is fulfilled, then using this condition to divide subtrees could happen. This process includes parallel shifting of the coordinate axis X_1 rightward with X_{v2} units, and continuing its partition in the new coordinate system.

It is therefore concluded that the essential condition that any subtree in decision tree can be established is that the upper level subtrees whose leaf nodes are their root nodes are accurate. If the overfitting exits in the upper level subtrees, then the following growing will be inaccurate in the first place, therefore the continuing partition would be meaningless. Hence, the incremental pruning process is carried out with top-down method, which can avoid the further testing on the lower level subtrees. Besides, in the process of incremental pruning, in order to make sure that the new training samples would maintain the same qualities that the original training sample collections have, the new samples and the old ones should be guaranteed to come from the same partition area.

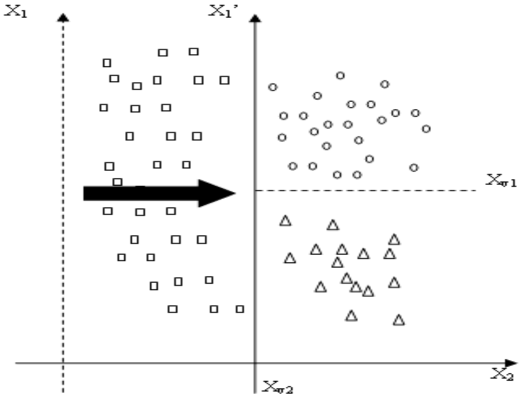

Fig. (3). The coordinate axis parallel shifting.

Now comes the following conclusion that the rules that are established by decision tree can be divided into two categories. One category is the deterministic rules, which are the true reflection of training sample collections' quality. They do not change with the changing of training sample collections' element quantity. The other category is contingency rules, which are usually created in the process of random partition with less training samples. These rules cannot accurately reflect the quality of training sample collections. With the changing of training sample structure, rules change correspondingly. It is very common in the daily learning since we can always correct the knowledge that once we believed to be correct through constant learning new samples. For example, assuming the training sample collection Train is shown in Table **1**, and Temperature and Humidity as condition attributes can equally divide the whole sample collection, which means they have the same information gain, they are believed to have the same classification ability.

Table 1. Training sample collections.

Outlook	Temperature	Humidity	Wind	Play Tennis
Sunny	Hot	High	Weak	No
Sunny	Mild	Normal	Weak	Yes
Sunny	Cool	High	Strong	No

But when the samples have been expanded as shown in Table **2**, the subsets produced by partitioning training sample collections are disorganized with Temperature being its condition attribute, whereas Humidity can obtain two organized subsets after partition.

Table 2. Training sample collections.

Outlook	Temperature	Humidity	Wind	Play Tennis
Sunny	Hot	Normal	Weak	Yes
Sunny	Hot	Normal	Strong	Yes
Sunny	Mild	High	Strong	No
Sunny	Cool	High	Weak	No
Sunny	Cool	Normal	Strong	Yes
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes

Through this process it is determined that Humidity as a condition attribute is a deterministic rule, and Temperature a contingency rule. The change of their information gain is presented as in Fig. (**4**).

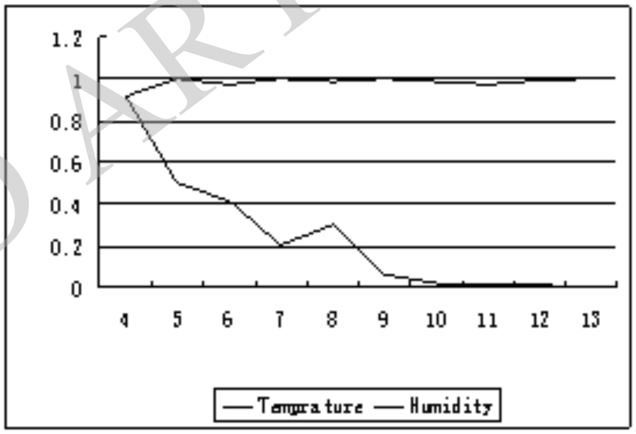

Fig. (4). The information gain change.

Based on the previous experiments, it is concluded that the essential idea of top-down decision tree incremental pruning starts with the root nodes, and it adds certain amount of new samples to each non-leaf node from the top down. If the addition does not cause any change in node's information gain, this node should be kept; otherwise, this node should be removed. The framework of this algorithm is like the following:

1) Defining pruning collection

$$PruneSet, PruneSet \in Set$$

$$PruneSet \cap TrainSet = \Phi$$

2) Add sample $d \in PruneSet$ to each non-leaf node *Node* from the root node *Root*. Calculate the information gain of *Node* before and after adding under the original test attribute *A*, which would be presented as $Gain(TrainSet, A)$ and $Gain(PruneSet \cup TrainSet, A)$. If $Gain(TrainSet, A)$ = $Gain(PruneSet \cup TrainSet, A)$, this subtree T_{Node}

Table 3. Attributes that are included in Iris dataset.

The Name of the Attribute	The Classification of Attribute Value	Possible Attribute Value
sepal length in cm	Continuous type	Nonnegative real number
sepal width in cm	Continuous type	Nonnegative real number
petal length in cm	Continuous type	Nonnegative real number
petal width in cm	Continuous type	Nonnegative real number
class	Discrete type	Iris Setosa/Iris Versicolour/Iris Virginica

should be kept; if $Gain(TrainSet, A)$ > $Gain(PruneSet \bigcup TrainSet, A)$, this subtree would be replaced with leaf node, whose classification indicator is the most common one of nodes.

3) If the subtree T_{Node} is kept, use original test attribute A to partition the current pruning collection $PruneSet$ into its corresponding subsets $PruneSet_1(A)$, $PruneSet_2(A)$,…, $PruneSet_n(A)$, and repeat the adding process to its each branch of the sub-trees $\mathrm{T}_{Node}(A)_1$, $\mathrm{T}_{Node}(A)_2$,…, $\mathrm{T}_{Node}(A)_n$.

4) This recursive process continues until it meets the leaf nodes.

4. EXPERIMENTAL TEST

The experimental data all come from the standard database UCI dataset of machine learning test. In order to provide experimental data for all kinds of classifier constructing methods in machine learning, the University of California at Irvine established a public database UCI Dataset. Currently, there are 211 datasets, which come from each aspects of human life, and each dataset describes numerous learning samples and their attributes. For example, the Iris dataset, which is frequently used by decision tree algorithm, includes 4 condition attributes and 1 classification attribute as shown in Table **3**.

Sepal length in cm, sepal width in cm, petal length in cm and petal width in cm are of condition attributes, whose value range is nonnegative real number. The last term *class* is the classification, whose discrete value range among Iris Setosa, Iris Versicolour, and Iris Virginica. Table **4** presents learning samples to explain the corresponding attributes.

Iris dataset includes 150 learning sample. The purpose of machine learning is to build a classifier through analyzing these learning samples, which can indicate the accurate class value based on the attributes of unknown samples.

The results of observing experiments has shown that directly using decision tree's classification accuracy as the classification accuracy of the unknown samples is not an appropriate approach. Independent test sample collections were used to make unbiased estimation of classification accuracy, some for building decision trees and others for their testing. However, the divisions of original sample collections varied, which could lead to a dramatic change in the classification accuracy of decision trees, particularly in the case of small sample collection. So a common method to achieve a stable classification accuracy is to obtain corresponding classification accuracy through multiple random divisions of original sample collections and get their mean value as the classification accuracy of the decision tree. Cross validation optimized the above-mentioned mean value method. In the first building of decision tree, firstly original samples would be divided into *k* portions, in which the first portion would be taken as the test collection and other k-1 portions as training collections. In the second building process, the second portion would be taken as the test collection and others as training collections. Repeat the whole process for k times and take the mean value of k times' results as the final estimated value. Compared with the original method, cross validation could make sure that each sample was used in test process for once, which increased the reliability of estimated value. The experimental results have shown that when the k is 10, the estimation of decision tree's classification accuracy is the most reliable.

The scale and classification accuracy of a decision tree have been much adopted to evaluate its quality.

1) Scale. Based on Occam's Razo theory, it is held that when the decision tree's scale is smaller, its classification ability would be higher, which is especially true when it comes to the statistical classifiers. A complex decision tree can well explain the training sample collections, but the test sample collections that are independent from training sample collections do not show good classification ability. Therefore in the process of decision tree pruning the decision trees scale need downsizing as much as

Table 4. The data of Iris dataset contains.

Sepal Length in cm	Sepal Width in cm	Petal Length in cm	Petal Width in cm	Class
5.1	3.5	1.4	0.2	Iris-setosa
5.8	2.7	4.1	1.0	Iris-versicolor
6.4	2.8	5.6	2.1	Iris-virginica
…	…	…	…	…

Table 5. Error rates of different pruning algorithms.

Database	REP	MEP	CVP	PEP	EBP	TDIP
Iris	5.7±0.6	6.2±0.6	5.9±0.5	5.3±0.6	5.1±0.6	5.6±0.5
Glass	38.5±1.2	38.1±1.4	36.9±1.4	35.3±1.4	35.9±1.3	35.4±1.3
Pima	25.9±0.4	27.2±0.4	30.0±0.4	28.9±0.5	28.8±0.5	31.4±0.4
Heart	23.5±0.6	23.7±0.9	24.1±0.5	22.9±0.5	29.3±0.5	23.8±0.6
Hepatitis	20.3±0.9	21.3±1.4	20.6±1.1	21.1±1.1	21.4±1.3	21.9±1.3
Blocks	3.2±0.1	3.4±0.1	3.6±0.1	3.0±0.1	3.1±0.1	3.6±0.1

Table 6. The number of leaf nodes by using different pruning algorithms.

Database	REP	MEP	CVP	PEP	EBP	TDIP
Iris	3.4±0.1	4.0±0.2	5.4±0.2	3.8±0.2	4.9±0.3	4.0±0.2
Glass	11.1±0.9	18.5±1.1	27.9±0.7	21.1±0.5	28.7±0.6	15.1±1.0
Pima	18.5±1.5	32.4±3.7	70.8±1.3	55.7±1.2	65.8±1.5	22.8±1.7
Heart	33.5±2.2	31.9±3.6	58.2±4.8	21.9±1.3	46.1±1.5	45.9±3.0
Hepatitis	2.7±0.4	8.4±0.4	3.6±0.7	5.4±0.5	9.1±0.5	4.4±0.6
Blocks	24.7±0.9	65.1±1.3	78.5±1.3	37.3±0.7	50.9±1.3	30.4±1.1

possible, which is often measured by the number of its nodes or leaf nodes and they are correlated. For example, assume the decision tree is a binary tree (which means that each node can only have 0 or 2 subtrees), and then the number of decision tree's leaf nodes would be equal to the number of nodes plus 1, or more when the decision tree is expanded to a non-binary-tree with multiple output values. Because the number of leaf nodes is equal to that of decision tree's equivalence rules collections, in the following discussion the number of leaf nodes is used to measure decision tree's scale.

2) Classification accuracy. It refers to the classification ability of a decision tree on independently distributed test sample collections. This article exploits the classification error rate as the measurement standard of classification ability, which is the percentage of the decision tree's test samples that have been falsely classified in the whole test sample collection. This calculation method does not present different class samples' classification accuracy. When different classifications are of unequal percentage in test sample collections, the fewer the class samples are, the lower the classification accuracy will be. Therefore, this criterion can only be considered as a rough judgment of decision tree's classification accuracy.

Through the comparison between the decision tree's scale and classification accuracy that has been perfected separately by using PEP, MEP, CVP, EBP, and TDIP, which was proposed in this article, advantages and disadvantages of each algorithm have been analyzed and TDIP has been proven efficient (Tables **5** and **6**).

Through comparing the experimental data of Tables **5** and **6**, we can see that the number of leaf nods in decision tree pruned by decision tree incremental pruning method is usually smaller than those pruned by other methods, and the classification accuracy is usually higher than that by other methods. Therefore, it is concluded that DIP gives priority to the limitation of decision tree's scale.

CONCLUSION

Decision tree is a common method of machine learning. But due to the existence of the noise in data input of decision tree algorithm, its node partition cannot make an accurate expression of the real rules of data source, which results in overfitting in various decision tree algorithms. To address this phenomenon, many researchers have proposed numerous optimization methods of decision tree, among which decision tree pruning algorithm has become an important branch. Currently, many different decision tree algorithms have been applied to actual pruning process, such as PEP, MEP, CVP, and EBP, which have been well-received. But different algorithms have their pruning preferences, and the quality and quantity of different data sources can affect the results of pruning. In order to solve the problems that have arisen in each aspect of decision tree pruning algorithm, this article has proposed a new type of decision tree pruning algorithm based on the serious studying of previous achievements. It is aimed to keep the deterministic rules and remove the contingency ones by comparing their difference in the information gain differences till it finds and removes the overfitting subtrees. Besides, this method can ensure an efficient and accurate decision tree pruning since it defines the pruning order, reduces the searching times and increases the algorithm's execution speed. Through the theoretical analysis together with the experimental results of algorithm, the, top-down decision tree incremental has been proven effective.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

None declared.

REFERENCES

[1] Y. Li, and C-H. Huang, “Influence of sports self-concept and self-efficacy on sports practice of university students,” *Journal of Sports Adult Education*, vol. 27, no. 6, pp. 32-35, 2011.

[2] J. Liu, “Comparative research on difference between body shape and function and physical fitness of college students for different body-mass index level,” *Journal of Pla Institute of Physical Education*, vol. 30, no. 1, pp. 125-128, 2011.

[3] J. Ning, “Study of transformation of college students' exercise methods,” *Bulletin of Sport Science & Technology*, vol. 21, no. 4, pp. 85-87, 90, 2013.

[4] D. Shi, and G. Yang, “Study on physical characteristics and influencing factors of fat students with difference physical exercise,” *Journal of Hubei Sports Science*, vol. 32, no. 3, pp. 215-217, 2013.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014