A Splitting Criteria Based on Similarity in Decision Tree Learning

Xinmeng Zhang
Cisco School of Informatics, Guangdong University of Foreign Studies, Guang Zhou, china
Email: javad0902@163.com

Shengyi Jiang
Cisco School of Informatics, Guangdong University of Foreign Studies, Guang Zhou, china
Email: jiangshengyi@163.com

***Abstract*—Decision trees are considered to be the most effective and widely used data mining technique for classification, their representation is intuitive and generally easy to be comprehended by humans. The most critical issue in the learning process of decision trees is the splitting criteria. In this paper, we firstly provide the definition of similarity computation that usually used in data clustering and apply it to the learning process of decision trees. Then, we propose a novel splitting criteria which chooses the split with maximum similarity and the decision tree is called mstree. At the same time, we suggest the pruning methodology. The empirical experiments conducted on benchmark datasets have verified that the algorithm has outperformed some classic algorithms such as id3, c4.5 in the classification precision, and less affected by the size of training set.**

***Index Terms*—data mining; decision tree; similarity; Classification**

I. INTRODUCTION

Data classification is an important data mining technique, which aim at predicting the class that a newly data items belongs to. In classification, a training algorithm is used for training the classifier by training set, and then a test algorithm is used for testing the classifier, for classifying the newly coming data. The learning process of decision trees usually contains two parts: a growing phase in which nodes are added to the tree based on a prediction gain, and a pruning phase in which the tree size is reduced in order to guard from overfitting and provide good generalization.

Classification is widely used in credit approval, target market positioning, medical diagnosis, fault detection areas etc.

Currently, the widely used classification methods include decision trees, neural networks, genetic algorithms, support vector machines, rough sets and Bayesian methods etc[1-4]. Decision tree is a common, intuitive, fast classification method. There are many decision tree construction algorithm, ID3 algorithm[1] proposed by Quinlan in 1986, is the first international, influential decision tree. It is considered as a very simple decision tree algorithm. ID3 uses information gain as splitting criteria. The growing stops when all instances belong to a single value of target feature or when best information gain is not greater than zero. ID3 does not apply any pruning procedure. It can't deal with numeric attributes neither missing values. Quinlan proposed C4.5 algorithm that improved algorithm ID3 [2],[5], adding the Discretization of Continuous Attributes and the processing functions of unknown attributes. Ruggieri's EC4.5 [6] algorithm improved the efficiency.

In recent years, many improved algorithms of decision tree have been proposed, Sattar proposed a novel classification algorithm, flexible decision tree (FlexDT), which extends fuzzy logic to data stream classification[7]. Gary provided some useful insights and a template for future analyses using more sophisticated cost models[8]. Mahmood presented two novel decision tree algorithms named C4.45 and C4.55, aimed at improving the AUC value over the C4.5[9].

Decision Tree node splitting is an important step, the core issue is how to choose the splitting attribute. The general approach is testing all splitting of each attribute, evaluate those results, and then select the splitting attribute with the best result. Therefore, the splitting criteria is a sensitive issue in constructing decision tree[10]. Splitting criteria is dependent on the purity of the measure, The main splitting criteria include information gain, information gain ratio, minimum description length[11], distance measuring statistics, weight of evidence, etc. In c4.5, the splitting criteria is calculating information gain of each attribute, then the attribute with the maximum information gain or information gain ratio is selected as splitting attribute.

In this paper, we present a novel splitting criteria based on similarity. In the algorithm, the training set is splitted into several subsets by the value of splitting attribute and the average similarity of all subsets is calculated, and so on, all attributes' average similarity is calculated one by one, then the attribute with the maximum average similarity will be selected as splitting node. The experimental results tested on UCI data sets showed that the algorithm is better than id3 and c4.5 algorithms in classification accuracy.

doi:10.4304/jsw.7.8.1775-1782

The rest of this paper is organized as follow. In section II, we discussed the method of similarity computation and proposed a splitting criterion based on similarity. In section III, a novel algorithm based on the splitting criteria is presented and the pruning methodology is discussed. In section IV, we described the experimental settings and results in detail. In section V, we draw the conclusions and pointed out the shortage of the algorithm which needed to be solved in the future.

II. The Splitting Criteria based on Similarity

Data clustering is another important technique in data mining. Unlike classification, data clustering is unsupervised learning, none of data items has label. The methods of evaluating cluster performance include internal quality measures and external quality measures. In internal quality measure, cluster performance is evaluated by calculating the average similarity of cluster or the similarity of overall clusters. Cluster validity indices are adopted to evaluate cluster performance[12][13], the widely used cluster validity indices includes Davies-Bouldin index, Dunn's index, Calinski-Harabasz index, and index I[14], the bigger the internal similarity of instances in each cluster, the better the cluster performance.

In decision tree learning, the training data are partitioned into several subsets according to the values of the splitting attribute, the algorithm proceeds recursively until all instances in a subset belong to the same class. Therefore, the fundamental technique is how to choose the splitting attribute that best partitions the training data. The widely used splitting criteria include information gain, minimum description length, probability estimation methods[15] etc. We apply internal quality measures of evaluating cluster performance to decision tree learning, calculate the internal similarity of subsets partitioned by the values of splitting attribute, choose the splitting attribute with the maximum similarity.

Definition 1. Given training set T containing n instances. The similarity between any two instances r_1 and r_2 can be computed by follow equation.

$$S_{\text{im}}(r_1,r_2)=\begin{cases}1 & r_1,r_2 \text{ labeled the same class}\\ 0 & r_1,r_2 \text{ labeled the different class}\end{cases} \quad (1)$$

Definition 2. Given a splitting attribute A, it's value set $\{a_1,a_2,...,a_v\}$, v is the number of value. Training set T is splitted into v subsets according to the value of splitting attribute A, $T\{T_1,T_2,...,T_v\}$, instances in Set $T_u(1\leqslant u\leqslant v)$ that splitted by attribute value a_u belong to k classes $\{c_1,c_2,...,c_k\}$ $(1\leqslant k\leqslant m)$, the average similarity of T_u can be computed by equation (2), expressed by $S_{\text{im}}(a_u)$.

$$S_{\text{im}}(a_u)=\frac{\sum_{r_i\in T_u,r_j\in T_u} s_{im}(r_i,r_j)\times 2}{|T_u|\times(|T_u|-1)} \quad (2)$$

Where $|T_u|$ is the size of set T_u.

Definition 3. According to equation(1), the similarity of any two instances in subset T_u belonging to same class is 1, and 0 otherwise. Therefore, the total similarity of all instances belonging to same class c_j is $C^2_{|c_j|}$ $(1\leqslant j\leqslant k)$, then equation(2) can be redefined as equation (3).

$$S_{\text{im}}(a_u)=\frac{\sum_{j=1}^{k} C^2_{|c_j|}\times 2}{|T_u|\times(|T_u|-1)} \quad (|c_j|>1,\ |T_u|>1) \quad (3)$$

$|c_j|$ is the number of instances labeled class c_j in subset T_u, it will be neglected if $|c_j| = 0$. $|T_u|$ is the size of subset T_u, If $|T_u|=1$, then $S_{\text{im}}(a_u)=1$. By equation (3), if we know the number of instances of each class in subset T_u, the average similarity of subset T_u can be computed.

Definition 4. The values of splitting attribute A is $\{a_1,a_2,\ldots,a_v\}$, their probability is$\{p_1,p_2,\ldots,p_v\}$, splitting dataset into v subsets $\{T_1,T_2,\ldots,T_v\}$. The average similarity of all subsets is the sum of the similarity of each subset multiply their probability, the average similarity is taken as the average similarity of attribute A, expressed by $S_{\text{im}}(A)$, which can be defined as.

$$S_{\text{im}}(A)=\sum_{i=1}^{v} S_{im}(a_i)\times p_i \quad (1<i<v) \quad (4)$$

The attribute with the greatest similarity is selected as splitting attribute according to the splitting criteria base on max similarity.

Table I is an example of training set, including 14 instances with 5 attributes, the attribute play is the class label. Aim at establishing the classification model that decided to whether or not the competition will be.

For example, taking these instances described in table 1 as training set, we construct a decision tree depend on the splitting criteria based on similarity.

TABLE I.
EXAMPLE OF TRAINING SET

outlook	temperature	humidity	windy	play
sunny	mild	high	false	no
rain	mild	normal	false	yes
sunny	cool	normal	false	yes
rain	mild	high	false	yes
rain	cool	normal	false	yes
rain	cool	normal	true	no
overcast	cool	normal	true	yes
overcast	hot	high	false	yes
overcast	hot	normal	false	yes
overcast	mild	high	true	yes
sunny	hot	high	false	no
overcast	mild	normal	true	yes
sunny	hot	high	true	no
rain	mild	high	true	no

First step, we calculate the average similarity of each attribute.

The value of attribute outlook include overcast, rain and sunny, then training set can be split into three subsets described as $\{T_1,T_2,T_3\}$ according to the three values. The subset T_1 represents the set of instances with overcast, including five instances, all of them belong to class yes, therefore, $\text{sim}(T_1)=1$, the probability of overcase is 5/14. Subset T_2 include all instance of rain, three instances of

subset T_2 are labeled yes, and the others are labeled no. According to equation (3), $S_{\text{im}}(c)=\frac{(C_3^2+C_2^2)\times 2}{|T_2|\times(|T_2|-1)}=\frac{(3+1)\times 2}{5\times(5-1)}$=0.4, the probability of rain is 5/14. There are three instances labeled as yes and one instance labeled as no in subset T_3, $S_{im}(T_3)$=(3+0)/6=0.5, the probability of sunny is 4/14.

Therefore, the average similarity of attribute outlook can be calculated as follow.

$S_{\text{im}}(\text{outlook})=S_{\text{im}}(T_1)*p_1+S_{\text{im}}(T_2)*p_2+S_{\text{im}}(T_3)*p_3=1*5/14+0.4*5/14+0.5*4/14=0.64286$

The average similarity of other attributes can be calculated as follow.

According to the attribute temperature, the dataset is split into three subsets$\{T_1,T_2,T_3\}$. The subset T_1 represents the set of instances with cool, including three instances labeled yes and one labeled no. T_2 is the set of hot, two instances labeled yes, and the other two labeled no. T_3 includes six instances of mild, four of them are labeled yes, others labeled no.

$S_{\text{im}}(T_1)$= 3/6=0.5

$S_{\text{im}}(T_2)$= (1+1)/6=0.33333

$S_{\text{im}}(T_3)$= (6+1)/15=0.466667

$S_{\text{im}}(\text{temperature})=S_{\text{im}}(T_1)*p_1+S_{\text{im}}(T_2)*p_2+S_{\text{im}}(T_3)*p_3=0.5*4/14+0.33*4/14+0.467*6/14=0.4381$

According to the value of attribute humidity, training set is split into two subsets$\{T_1,T_2\}$. T_1 represents the set of instances with high, four of them are labeled no, and others are labeled yes. T_2 is the set of other instances, only one is labeled as no.

$S_{\text{im}}(T_1)$= (6+3)/21=0.4286

$S_{\text{im}}(T_2)$= 15/21=0.7143

$S_{\text{im}}(\text{humidity})=S_{\text{im}}(T_1)*p_1+ S_{\text{im}}(T_2)*p_2=0.4286*7/14+0.7143*7/14=0.5714$

According to the attribute windy, we can split the data set into two subsets$\{T_1,T_2\}$. T_1 represents the set of instances with false, there are eight instances in it, two of them are labeled no, and others are labeled yes. T_2 includes six instances, half of them are labeled yes.

$S_{\text{im}}(T_1)$=(15+1)/28=0.5714

$S_{\text{im}}(T_2)$=(3+3)/15=0.4

$S_{\text{im}}(\text{windy})=S_{\text{im}}(T_1)*p_1+S_{\text{im}}(T_2)*p_2=0.6071*8/14+0.4*6/14=0.4980$

sim(outlook)>sim(humidity)>sim(windy)>sim(temperature), then the attribute outlook is the best splitting attribute, each subset partitioned by the values of attribute outlook will be the node of decision tree.

Next step, in the same way , the algorithm recursively calculates the average similarity of each remaining attributes on the subsets which are partitioned by the value of splitting attribute in the last step until all instances in a subset belong to the same class.

As Fig. 1, without any pruning strategies, the decision tree generated by this algorithm is same as ID3 tree. If we set the splitting threshold as $(n\text{-}2)/n$, n is the number of instances in subsets, the tree is same as C4.5 tree showed as Fig. 1 without dotted line part.

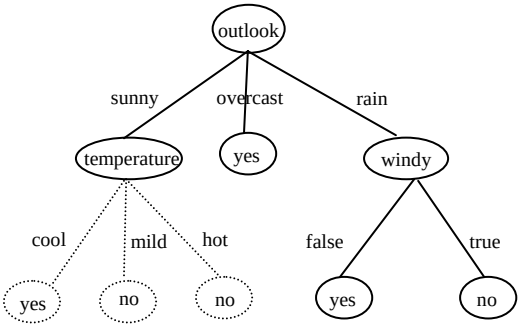


Figure 1. decision tree based on max similarity

III. General Framework of Mstree Algorithm

The tree-growing algorithm based on the maximum similarity, called mstree algorithm, is illustrated as follows.

A. Main Framework of Algorithm

The decision tree generation algorithm based on maximum similarity is same as ID3 in general framework, described as follows.

Generate_decision_tree(T,A,Y)
- Input: a training set T, condition attribute set A, target attribute set Y
- Output: decision tree.
- Create a node N
- Compute the similarity of target attribute Y for training set
- If $S_{\text{im}}(Y)$>=r_1 or |T|<=r_2 then
- Return N as a leaf node, label it with the value of attribute Y
- if A is null then
- Return N as a leaf node, and label it with the highest frequency value of target attribute A.
- test_attribute=Attribute_selection(T,A,Y).
- Label node N as test_attribute;
- If test_attribute is continuous type
  - CalculateSplittingValue(T,A)
- For each a_i in values of test_attribute do
- Generate a branch from node N according to condition test_attribute=a_i
- Set s_i as the instances set according to condition test_attribute=a_i
- If test_attribute is discrete type
- A= A - test_attribute
- Label Generate_decision_tree(s_i,A,Y) as node

B. Select the Attribute with Maximum Similarity

The function of selecting the splitting attribute with maximum similarity namely attribute_selection(T,attribute_list), described as follows:

Attribute_selection(T,attribute_list)
- Input: subset T, attribute set attribute_list
- Output: the attribute with maximum similarity
- For each A_j in attribute_list do
  - For each r in T do
    - Split T into subset T_{ji} according to the value of attribute A_j

End for
Sim(A_j)=0
For each a_i in values of A_j do
Sim(a_i)=Value_sim(T_{ji},a_i,A_j)
Sim(A_j)= Sim(A_j)+Sim(a_i)*p(a_i)
End for
End for
Get the attribute A_{max} with maximum sim(A_j)
Return A_{max}

C. Calculate the Similarity of Attribute Value

Value_sim(T,V,A)
Input: subset T, an attribute A, V is a value of attribute A
Output: the similarity of subset T partitioned by the value V of attribute A
For each instance r in T do
Split T into subset T_c={T_{c1},T_{c2},…,T_{ck}} according to the class which r belongs to
T_{ck} represents the set of instances labeled class ck
End for
Sim(V) = 0
For each T_{ci} in T_c do
Calculate the similarity of T_{ci}
Add the similarity to Sim(V)
End for
Return Sim(V)

D. Calculate Continuous Splitting Value

CalculateSplittingValue(T,A)
Input: subset T, a continuous attribute A
Output: the splitting value
Sort the dataset T by attribute A_i
The sequence of values is V={v_1,v_2,…,v_m}, m=|V_A|
|V_A| is the size of subset T
max_v=0
For each v_i in V do
Calculate the average similarity of two subsets that divided by v_i, expressed as Sim(v_i)
If max_v < Sim(v_i)
max_v = Sim(v_i)
splittingValue= v_i
end if
End for
Return v_i

E. Pruning Methodology

Usually, there are noise datas in training set, which may lead to overfitting, then noise branch will be generated. We take the splitting threshold method to eliminate overfitting. Two thresholds are adopted in the pruning methodology, one is the similarity of subset named r_1, and the other threshold is the size of subset named r_2, if the similarity of subset is greater than r1 or the size of the subset is less than r_2, then stop splitting, and label the node as leaf node.

For example, if only one instance is labeled different class from the other instances in subset, the instance is likely a noise data, and it isn't right to split. To eliminate the noise data like this, r1 can be set to (n-2)/n. If subset includes n instances, n-1 of them are labeled the same class, just one instance belongs to other class, then the internal similarity of subset is ((n-1)*(n-2)/2)/(n*(n-1)/2)=(n-2)/n. Therefore, if just one instance in the subset belongs to other class, the internal similarity of subset equals (n-2)/n, then stop splitting.

If the size of subset is too small, it is unnecessary to split for avoiding overfitting, the node will be labeled as leaf node, and labeled with the class which majority instances belong to.

IV. Experimental Analysis

We conducted our experiments on 9 UCI benchmark datasets which are selected by Weka and represent a wide range of domains and data characteristics. The description of 9 data sets is shown in Table II. The data sets include: car evaluation, Nursery, Mushroom, kr-vs-kp, Tic-Tac-Toe, balance-scale, diagnosis, haberman, breast-cancer. Five data sets have only discrete attribute, one data sets have both discrete attributes and continuous attributes, and the rest have only continuous attribute.

TABLE II.
Description Data Sets Used in The Experiments

Datasets	Discrete attributes number	Continuous attributes number	Instances number	Class number
car evaluation	6		1728	4
Nursery	8		12960	5
Mushroom	22		8124	2
kr-vs-kp	36		3196	2
Tic-Tac-Toe	9		958	2
diagnosis	5	1	120	4
balance-scale		4	625	3
haberman		3	306	2
breast-cancer		10	699	2

The more detailed description of the nine data sets as follow.

- car evaluation: The Car Evaluation Database contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes: buying, maint, doors, persons, lug_boot, safety, four class values: unacc (70.023 %) , acc (22.222 %) , good (3.993 %) ,v-good (3.762 %).
- Nursery: The database was derived from a hierarchical decision model originally developed to rank applications for nursery schools. It was used during several years in 1980's when there was excessive enrollment to these schools in Ljubljana, Slovenia, and the rejected applications frequently needed an objective explanation. The final decision depended on three subproblems: occupation of parents and child's nursery, family structure and financial standing, and social and health picture of the family. It includes eight input

attributes: parents, has_nurs, form, children, housing, finance, social, health, Five classes not_recom (33.333 %), recommend (0.015 %), very_recom (2.531 %), priority (32.917 %), spec_prior 4044 (31.204 %).

- Mushroom: This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like “leaflets three, let it be” for Poisonous Oak and Ivy.
- kr-vs-kp: The format for instances in this database is a sequence of 37 attribute values. Each instance is a board-description for this chess endgame. The first 36 attributes describe the board. The last (37th) attribute is the classification: "win" or "nowin".
- Tic-Tac-Toe: This database encodes the complete set of possible board configurations at the end of tic-tac-toe games, where "x" is assumed to have played first. The target concept is "win for x".
- Diagnosis: The data was created by a medical expert as a data set to test the expert system, which will perform the presumptive diagnosis of two diseases of urinary system. Each instance represents a potential patient. It will be the example of diagnosing of the acute inflammations of urinary bladder and acute nephritises.
- balance-scale: This data set was generated to model psychological experimental results. Each example is classified as having the balance scale tip to the right, tip to the left, or be balanced. The attributes are the left weight, the left distance, the right weight, and the right distance. The correct way to find the class is the greater of (left-distance * left-weight) and (right-distance * right-weight). If they are equal, it is balanced.
- haberman: The dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer.
- breast-cancer: Samples arrive periodically as Dr. Wolberg reports his clinical cases. 10 input attributes: Sample code number, Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Mitoses. Six classes: 2 for benign, 4 for malignant.

The mstree algorithm is programmed by java, and the experiments were performed under the windowsXp. The 10% to 90% of instances were randomly chosen as training set, and the rest instances as testing set, the average of 100 times experiment results were taken as the final result. By the same training set selection method, the experimental results of id3,c4.5 and cart are gained through weka 3.7 platform.

A. Classification Accuracy on Discrete Attributes

For discrete attribute, we tested the mstree, id3 and c4.5 algorithm on following five data sets: car evaluation, Nursery, Mushroom, kr-vs-kp, Tic-Tac-Toe. Experimental results charts are shown in Fig. 2-Fig. 6, x-axis represents the proportion of training instances, y-axis represents classification accuracy. It can be seen from these charts that curve of mstree is More smooth than others, and the charts show that the mstree algorithm is less affected by training set size. In particular, classification accuracy is significantly higher than that of id3 and c4.5 algorithms on the small training set.

Fig. 2, fig.3 and fig. 4 show that the accuracy of mstree has significantly outperformed id3 and c4.5 whatever the training set scale on car evaluation, nersery and tic-tac-toe.

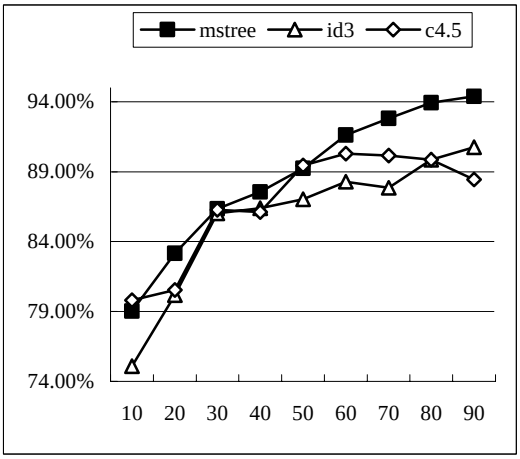


Figure 2. Comparison of mstree,id3 and c4.5 on car evaluation

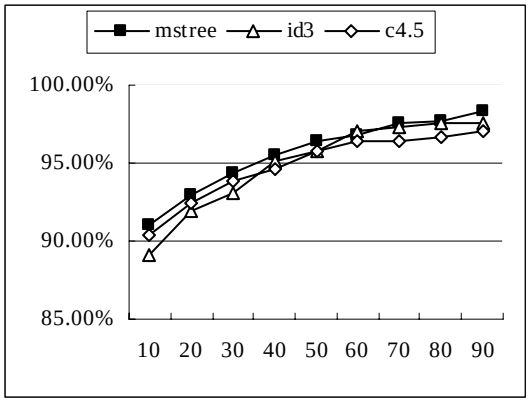


Figure 3. Comparison of mstree,id3 and c4.5 on nersery

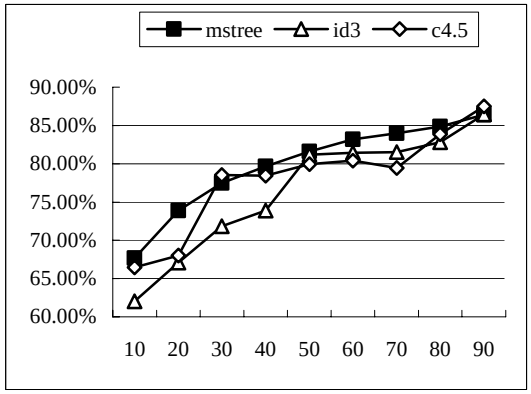


Figure 4. Comparison of mstree,id3 and c4.5 on tic-tac-toe

Fig. 5 show the experiment results tested on mushroom for three algorithms. It is obvious that the mstree algorithm has an advantage over the other algorithms on small training set scale, and what is very important. Recently, imbalance data ming become a new research hot, sometimes the training set scale is far less than testing set scale. Therefore, mstree algorithm is more suitable to imbalance data mining than id3 and c4.5.

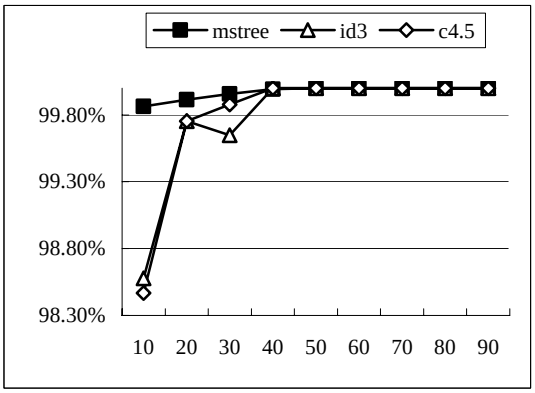


Figure 5. Comparison of mstree, id3 and c4.5 on mushroom

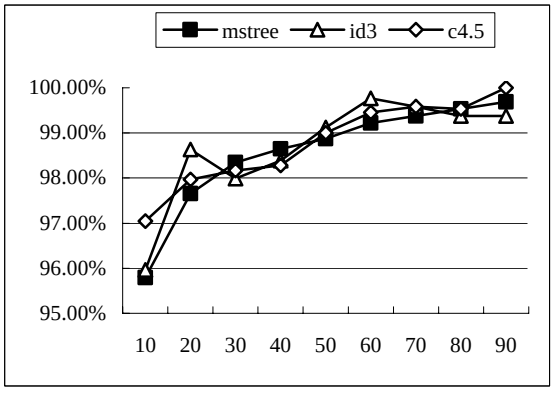


Figure 6. Comparison of mstree, id3 and c4.5 on kr-vs-kp

B. Classification Accuracy on Continuous Attributes

For continuous attribute, Fig. 7 to fig. 9 show a detailed view of experimental results tested on haberman, breast-cancer and balance-scale with only continuous attributes. Similar results with the results tested on discrete attribute data sets, the accuracy of mstree is relatively more stable on different ratio training set scale than that of c4.5 and cart.

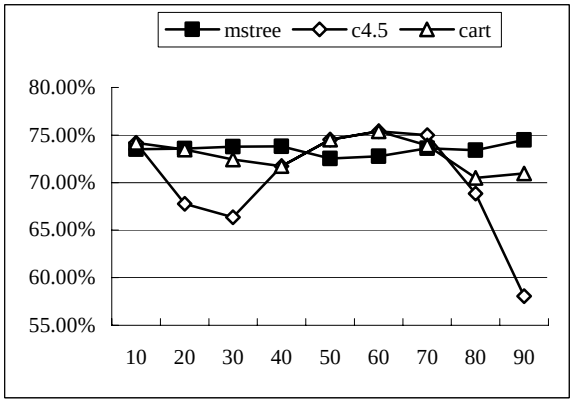


Figure 7. Comparison of mstree, cart and c4.5 on haberman.

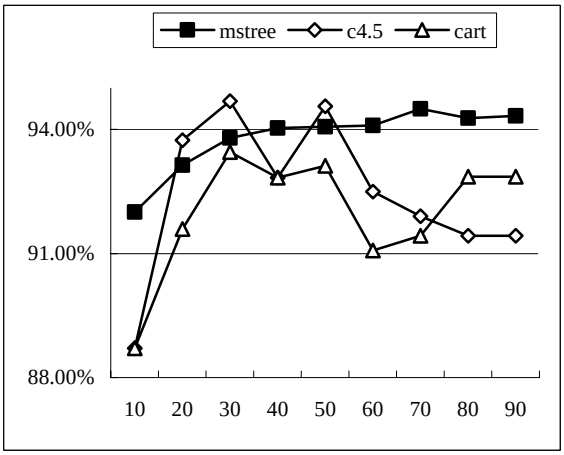


Figure 8. Comparison of mstree, cart and c4.5 on breast-cancer

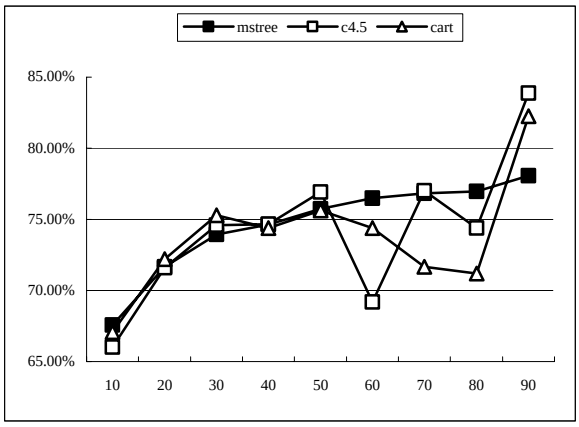


Figure 9. Comparison of mstree, cart and c4.5 on balance-scale.

C. Classification Accuracy on Mixed Attributes

Diagnosis data sets has one continuous attribute and five discrete attributes, fig.10 show that the mstree algorithm have advantage over c4.5 and cart in accuracy when training set ratio is less then 40%.

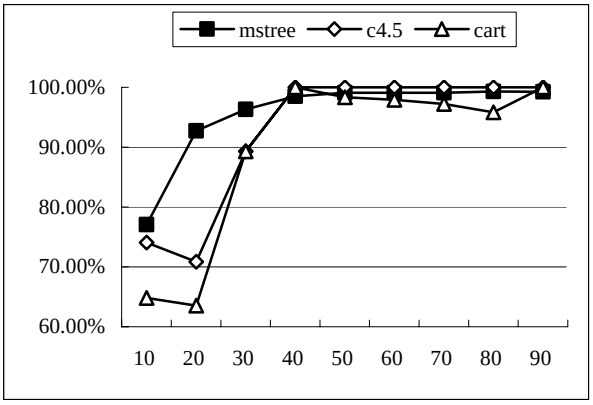


Figure 10. Comparison of mstree, cart and c4.5 on diagnosis

D. Decision Tree Size

The size of decision trees generated by id3, c4.5, cart and mstree are shown in Table III. It is seen that the size of tree generated by mstree is smaller than id3, but bigger than c4.5 and cart tree. The reason is that threshold selection is also not optimal. It is a problem how to select appropriate threshold for stopping growth.

TABLE III.
COMPARISON OF TREE SIZE

Data sets	Mstree	c4.5	id3	cart
nersery	724	511	1158	381
Car evaluation	197	182	405	115
tic-tac-toe	163	142	375	61
Mushroom	27	29	37	13
kr-vs-kp	67	59	94	73
balance-scale	41	103		25
diagnosis	11	11		11
haberman	5	5		1
breast-cancer	21	29		13

V. CONCLUSIONS

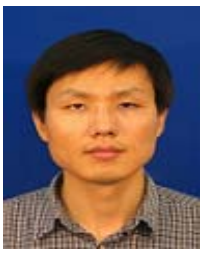
Decision tree is one of the most commonly used methods of Data classification, this paper presents a similarity-based decision tree generation algorithm and the pruning methodology. From our experiments, the mstree algorithm has two advantages over id3, c4.5 and cart algorithm: 1) It performs the better classification accuracy than the other algorithm averagely. 2) It decreases the influence by the ratio of the training set scale, and is more suitable to imbalance data mining. But, the mstree algorithm still needs to be improved. The Further work is to study how to optimize the splitting threshold, and reduce decision tree size.

ACKNOWLEDGMENT

We thank anonymous reviewers for their valuable comments and suggestions. This work is supported by Natural Science Foundation of China under grant no. 61070061.

REFERENCES

[1] J. R. Quinlan, Induction of decision trees. Machine Learning, vol.1,pp. 81-106, 1986.
[2] J. R. Quinlan, C4.5:Programs for machine learning. 1st ed. San Mateo, CA: Morgan Kaufmann, 1993.
[3] T. T. Wong, Alternative prior assumptions for improving the performance of naïve Bayesian classifiers, Data Min Knowl Disc, vol.18, pp.183–213, 2009.
[4] M. Hall, E. Frank, Combining Naive Bayes and Decision Tables. In D.L. Wilson & H. Chad (Eds), Proceedings of Twenty-First International Florida Artificial Intelligence Research Society Conference, AAAI, 2008, pp. 318-319.
[5] J. R. Quinlan, Improved use of continuous attributes in C4.5, Journal of Articial Intelligence Research, vol.4, pp.77-90, 1996.
[6] S. Ruggieri, Effcient C4.5. IEEE Transactions on Knowledge and Data Engineering,vol.14, pp.438-444, 2002.
[7] H. Sattar, Y. Ying, Flexible decision tree for data stream classification in the presence of concept change, noise and missing values. Data Min Knowl Disc, vol.19, pp.95–131, 2009.
[8] W. M. Gary, T. Ye. Maximizing classifier utility when there are data acquisition and modeling costs. Data Min Knowl Disc, vol.17,pp.253–282,2008.
[9] A. M. Mahmood, K. M. Rao, K. K. Reddi, et al. A Novel Algorithm for Scaling up the Accuracy of Decision Trees. International Journal on Computer Science and Engineering, vol.2, pp. 126-131, 2010.
[10] C. Drummond, R. Holte. Exploiting the cost (in)sensitivity of decision tree splitting criteria. Proceedings of the Seventeenth International Conference on Machine Learning .pp. 239–246,2000.
[11] S. Einoshin Suzuki. Compression-Based Measures for Mining Interesting Rulesc, Lecture Notes in Computer Sciencec,vol. 5579, pp. 741-746,2009.
[12] M. Halkidi,V. Michalisc, B. Yannis. Quality scheme assessment in the clustering process. Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, vol.1910,pp.265-276, 2000.
[13] T. J. Lim, W. Y. Loh, Y. S. Shih. A comparison of prediction accuracy, complexity,and training time of thirty-three old and new classification algorithms. Machine Learning, vol.40, pp.203–228, 2000.
[14] U. Maulik, S. Bandyopadhyay, Performance Evaluation of Some Clustering Algorithms and Validity Indices, ieee transactions on pattern analysis and machine intelligence, vol.24, pp.1650-1654,2002.
[15] L. Jiang, C. Li, An Empirical Study on Class Probability Estimates in Decision Tree Learning, journal of software,vol.6, pp.1368-1372,2011.



Xinmeng Zhang was born in Shandong, China. He received his M.Sc. in Software Engineering from the Guangdong University of Technology and B.Sc. in computer application from Shandong University, China. Currently, he is a lecturer in Cisco School of Informatics at Guangdong

University of Foreign Studies. His research interests include data mining and machine learning.

Shengyi Jiang was born in Hunan, China. He received his Ph.D. in computer application from the Huazhong University of Science and Technology, China; received his M.Sc. in applied mathematics from Central South University and B.Sc. in mathematics from the Hunan Normal University, China. He is a professor in Cisco School of Informatics at the Guangdong University of Foreign Studies, Guangzhou, China.

His research interests include data mining, natural language processing and network security, which are funded by the National Natural Science Foundation of China under grant # 61070061 and # 60673191. He has published over 50 papers in journals oracademic conferences, about 20 papers indexed by EI, 3 papers indexed by SCI.