Decision Tree Construction Algorithm Based on Association Rules

Yong Peng
College of Computer Science
Zhejiang University
Hangzhou,China
zjupy@zju.edu.cn

Yanming Ye, Jianwei Yin
College of Computer Science
Zhejiang University
Hangzhou,China
yeym@zju.edu.cn

***Abstract*—With the rapid development of Internet, there are more and more products information exist on the Internet. In order to facilitate the retrieval, how to classify these information automatically is becoming much more essential. So far there are many text-based classification models such as Decision Tree Model, Rule Induction Model, Bayes Model, and SVM Model and so on. In this paper, we present a new model which uses association rules to construct decision tree. The model theoretically has the better effect and the experimental results show the effectiveness of the model.**

Keywords-Text-based classification, Term association, Decision tree

I. INTRODUCTION

All manuscripts must be in English. These guidelines include complete descriptions of the fonts, spacing, and related information for producing your proceedings manuscripts. Please follow them and if you have any questions, direct them to the production editor in charge of your proceedings at Conference Publishing Services (CPS): Phone +1 (714) 821-8380 or Fax +1 (714) 761-1784.

With the development of B2C, the information of products online increases exponentially recently and how to retrieve the needed information effectively becomes increasely urgent. The most important is to classify them firstly. But the manual classification would be laborious and time-consuming. So there appears various algorithms and method to facilitate the classification automatically and based on the study of these algorithms we present an new classification model.

The model we proposed in this paper combine the association rules mining [1,2,3] and decision tree algorithm[4,7] to gain more result. In this model, we first use the association rules mining to get the rules. and then we use them to build the decision tree. Next, we apply the decision tree to the information classification. Finally, we test the model with a lot of data and compare its result with the traditional decision tree algorithm.

The reminder of this paper is organized as follows: Section 2 gives us a general introduction on association rule. Then we proposal a new model using association rules as attribute strengthen in Section 3. And In Section 4 give some experiments and results which can verify the model's effectivity. In last Section we give a summary of our research and discuss some future work.

II. RELATED WORK

There are many algorithms in text-based classification field, most of which are based on machine learning or probabilistic models. Among them Decision Tree Algorithm, Naive-Bayes Model [9], Rule Induction Algorithm and Support Vector Machines [8] are popular. Although those models are efficient and effective, how to get a higher accuracy and be more efficient always is a major area of research. So we try to apply the Association Rule Algorithm to the Decision Tree Algorithm and expect to gain the better result.

A. Assoication Rule Mining

Association rule is a kind of pattern contained in documents, and the association rule mining is the method to mine all the association rules in the certain document.

The problem of association rule [9] can be stated as follows: $I=\{i_1,i_2,...,i_n\}$ is a set of items and $T=\{t_1,t_2,t_3,...,t_n\}$ is a set of transactions where each transaction t_i is a set of items ($t_i \subseteq I$). An association rule is an implication of the form $X \to Y$, where $Y \subset I$ and $X \cap Y = \varnothing$.

$$support = \frac{(X \cup Y).count}{n} \quad (1)$$

$$confidence = \frac{(X \cup Y).count}{X.count} \quad (2)$$

As formula (1) shows, support is a set of items called itemset. A transaction $t_i \subseteq T$ is said to contain an itemset X if X is a subset of t_i, The support count of X (denoted by $X.count$) is the number of transaction in T that contain X. The strength of a rule is measured by its support (1) and confidence (2).

B. Decision Tree

Decision Tree learning is one of the most widely used techniques for text-based automatic classification. The learned model is represented as a tree, called decision tree. The Fig.1 is a simple example of decision tree.

Published by Atlantis Press, Paris, France.

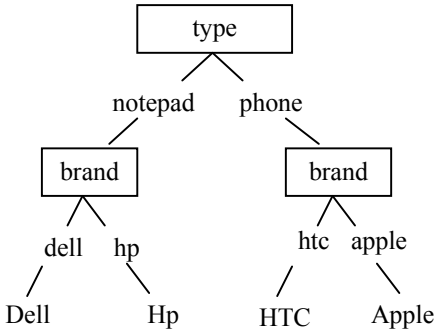

Figure 1. An example of decision tree

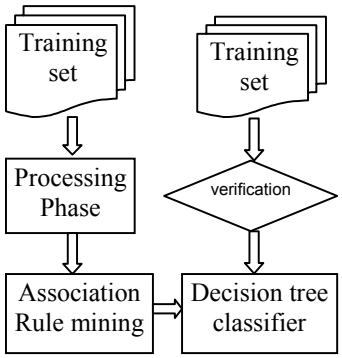

Figure 2. Graphically describe of Model

As Fig.1 shows, the root of the decision tree is type. If the type is notepad, then it turns left and so on. The leaf is the category which the document belong to. In practical application, we traverse the tree top-down according to the attributes values of the given instance until we reach a leaf node, then we can predict that the leaf is the predicted class of the instance.

III. PROPOSAL MODEL

In this section, we will introduce our classification model. In the model, we first mine association rules of the documents, then use the association rules to build the decision tree. The model is graphically described as Fig.2.

A. Term assiciation

Before mine the association rules, we first process the documents to get rid of all the phases that have no connection with the document, such as " a ", " an ", " the " and change the form of some phases.

All the category labels are predefined in the form of $C = \{c_1, c_2, ..., c_n\}$, and then we use association rule mining algorithm (ARM in short) to mine the association rules. ARM algorithm is based on Apriori[5], and described as follows.

Algorithm ARM (T)

Input A set of documents with the form $D(t_1, t_2, ..., t_n, c_i)$

A minimum support threshold and minimum confidence threshold;

Output A set of association rules with the form $R(t_1, t_2, ..., t_n, c_i)$

Method:

(1) $C_1 \leftarrow init - pass(T)$

(2) find the one frequent set that satisfies the minsup and minconf;

(3) for $(k = 2; F_{k-1} \neq \varnothing; k++)$ do

(4) find the k frequent set that satisfy the minsup and minconf;

(5) endfor

(6) return $R \leftarrow U_k R_k$

In the algorithm, C_1 basically associates each item in I with every category label, Line 2 determines whether the candidate 1-ruleitems are frequent. In Line 3, we generate k-condition Rs, then we generate all the frequent ruleitems by making multiple passes over data, then at last we determine which is actually frequent(4) and generate the final rule set R(6).

B. Decision Tree Buliding

After getting the association rules, we will build the decision tree. First, we resort to the association rules to sort the category, for example with the rule $\{t_i\} \rightarrow \{c_i\}$, if we get $\{t_i\}$ in one document then we can make t_i a decision node and make it left child labeled with c_i, and for every t_i, if we have the rule ,we name it effective rule. And then we start with the most frequent category to find the decision tree node. If we cannot find any effective rule, we compute the entropy of each attribute and find the most proper attribute as the decision node. Below give a detail description about how to make a decision tree.

Algorithm Decision Tree (D, A, R, T)

Input A set of documents with the form $D(c_i, t_1, t_2, ..., t_n)$

A set of association rule with the form $R(t_1, t_2, ..., t_n) \rightarrow C(c_i)$

Output A decision tree classifier

Method:

if D contains only training examples of the same class C_j

(2) then make T a leaf node labeled with class C_j

(3) else if $A \neq \varnothing$ then make T a leaf node labeled with C_j

(4) else if(find effective rules in R) make T a decision node on A_i

(5) else compute D's entropy and find the best attribute A_i

(6) make T a decision node on A_i

(7) endif

(8) endif

(9) endif

(10) DecisionTree $(D_j, A - \{A_g\}, T_k)$;

The learning of the tree is typically done using divide-and-conquer strategy that recursively partitions the data to produce the tree. The stopping criteria of the recursion (1) indicate that when all the training examples are of the same class. In Line 4-5 we use effective rule to construct the decision tree and when no effective rule is found we resort to the impurity function to do it (5-6).

IV. Experimental Results And Performance Study

The experimental data we used are crawled from some famous B2C website, we choose notepad as the test product and whose information $D = \{name, type, description\}$ as the main experimental data. The categories are $C = \{Dell, Lenovo, Asus, Hp, Acer, Apple, others\}$ .All the data are stored in an XML file, and every node of the file is a piece of the data.

Now we will compare the proposal model and the traditional decision model from precision, recall and F-score. We first import four variables, TP,FN,FP,TN where TP is the amount of documents classified positive and belong to positive and FN classified negative. The FP and TN is on the otherwise.

The precision p, recall r and F-score are defined as below respectively:

$$p = \frac{TP}{TP + FP} \tag{4}$$

$$r = \frac{TP}{TP + FN} \tag{5}$$

$$F - score = \frac{2pr}{p + r} \tag{6}$$

In the experiment, we use the crawled 13835 pieces of record, as the date source and of which, 8000 pieces will be the train data and the rest will be classified to test the effective of the classifier. First we get the precision of the every class that the new model produce and compare it with the precision of the class that the traditional decision tree model, then we give the total result. The precision of every class is showed in the Table1 and the total result is shown in Table2.

TABLE I. THE PRECISION OF EACH CLASS

Algorithm	Dell	Lenovo	Asus	Hp	Acer	Apple	Others
AR-DR	0.832	0796	0.321	0.923	0.316	0.768	0.864
T-DR	0.810	0.763	0.531	0.863	0.298	0.685	0.802

TABLE II. TOTAL PERFORMANCE OF TWO CLASSIFIER

Algorithm	TP	FN	FP	TN	P	R	F-score	time
AR-DR	4134	35	1637	29	0.716	0.991	0.831	42.90
T-DR	3946	223	1619	47	0.709	0.946	0.810	48.41

The Table II describe the performance of the proposal model(AR-DR) and the traditional decision tree model (T-DR), mainly about the TP , FN, FP, TN, P, R, F-score and time.

As what we can have seen in the Table I, with different categories we have different precision, and with the same category, different classifiers get different precision. But as the results show, six of seven categories perform better in AR-DR algorithm than in T-DR algorithm. In the Table2 we compare the TP, FN, FP, TN, P, R, F-score, time of two models. The results show that the AR-DR performs better than the T-DR in both accuracy and efficiency.

V. Conclusion And Future Work

In this paper, we present a new decision tree model which use the association rules to construct the decision tree, and the experimental results show that the new classifier perform better than the traditional decision tree construction algorithm. But in some condition the precision of proposal algorithm is very low, this is mainly because the association rules associate too many classes, and cannot find the most proper decision node. So how to make this algorithm more stable is the future work we need to do.

Acknowledgements

This work has been support by National Key Science and Technology Research Program of China (No.2011ZX01039-001-002, No.2009ZX01043-003-003).

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Proc.1993 ACM - SIGMOD Int. Conf. Management of Data, pp 207–216, Washington, D.C., May 1993.

[2] O. R. Za¨ıane and M.-L. Antonie. Classify- ing text documents by associating terms with text categories. In Thirteenth Australasian Da-ta base Conference (ADC'02), pages 215–222, Melbourne, Australia, January 2002.

[3] D. M. Blei, et al., "Latent Dirichlet allocation," Journal of Machine Learning Research, vol. 3, pp. 993-1022, 2003

[4] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In ACM-SIGMOD, Dallas, 2000.

[5] H. Gao, Y. Fu and J.P. Li, "Classification of Sensitive Web DOCUMENTS", "Apperceiving Computing and Intelligence Analysis, 2008. ICACIA 2008. International Conference", pp. 295-298 , Dec. 2008.

[6] FABRIZIO SEBASTIANI, "Machine Learning in Automated Text Classification", ACM Computing Surveys , F. 2009.

[7] D. Barbará, C. Domeniconi, N. Kang, "Mining Relevant Text from Unlabelled Documents", Proceedings of theThird IEEE Intern-ational Conference on Data Mining, pp. 489 – 492, 2000.

[8] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In 10th European Confere- nce on Machine Learning (ECML-98), pp. 137–142, 1998.