Text Categorization Using Association Rule Based Decision Tree

Mohammad Masud Hasan[1], Chowdhury Mofizur Rahman[2]

[1]Lecturer, Dept. of CSE, BUET, Dhaka.
[2]Professor, CSE Dept., United International University, Dhaka.
Emails: [1]hasanmm@cse.ac.bd, [2]cmr@uiu.ac.bd

ABSTRACT

Text categorization using decision tree is studied here. Instead of using words, word-relation i.e. association rules from these words, is used for building decision tree. In our experiments, we first preprocess data. We then find out association relations among these words using Rakesh Agrawal et. al.'s Apriori algorithm applying objective interestingness measures. These rules are used for training and testing the decision tree based classification system. We use the decision tree generator software of Quinlan's C4.5 system. A discussion of the result obtained is also given.

Keywords: Text categorization, decision tree, data cleaning, association rules, Apriori algorithm, confidence, support.

1. INTRODUCTION

Association rules have received much attention in the past. Rakesh Agrawal, Usama M. Fayyad, T. Imielinski, J. M. Bugajski, Ramakrishnan Srikant, H. Toivonen, H. Mannila, T. Zhang, C. Silverstein and many other scintillating researchers have worked here. There are two fundamental problems in the study of association rules: association rules and mining association rules. Recently, data mining techniques have been developed that apply concepts used in association rule mining to the problem of classification [7]. ARCS and associative classification [5] use association rules for classification. CAEP mines "emerging patterns" that consider the concept of support used in mining associations. An alternative classifier, called the *JEP-classifier*, was proposed based on jumping emerging patterns (JEPs). In this work, C4.5 is used for the analysis of text categorization system based on decision tree using association relations rather than using individual word as feature, which is a fully statistical approach.

Application domains for association rules range from decision support to telecommunications alarm diagnosis and prediction [4]. The prototypical application is in analysis of sales data. Classification has numerous applications including credit approval, medical diagnosis, performance prediction, selective marketing, indexing texts to support document retrieval [8] and extracting data from text [14].

2. MINING ASSOCIATION RULES

The following is a formal statement of the association rule [2]: Let $I = \{i_1, i_2, ..., i_m\}$ be a set of literals, called items. A set of items $X \subset I$ is called an itemset. We say that a transaction T contains an itemset X, if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \varnothing$.

Several objective measures of association rule interestingness exist based on simplicity, certainty, utility and novelty. We will use certainty measure confidence and utility function support. Association rules that satisfy both a user-specified minimum confidence threshold and user-specified minimum support threshold are strong association rules and are considered interesting. Rules below the threshold likely reflect noise, rare or exceptional cases, or minority cases and are excluded. Itemset satisfing minimum support is a frequent itemset. Association rule mining is a two-step process [7]:

1. Find all frequent itemsets.
2. Generate strong association rules from the frequent itemsets.

Additional interestingness measures can be applied, if desired. The overall performance of mining association rules is determined by the first step. Apriori [2] is an influential algorithm for mining frequent itemsets using candidate generation for Boolean association rules. In order to use the Apriori property, all nonempty Subsets of a frequent itemset must also be frequent. Once the frequent itemsets from transactions in a database have been found, we generate strong association rules from them using the following equation for confidence

$$\text{confidence } (A \Rightarrow B) = P(B \mid A) = \frac{support\_count\ (A \cup B)}{support\_count\ (A)}$$

where *support_count* ($A \cup B$) is the number of transactions containing the itemsets $A \cup B$, and *support_count* (A) is the number of transactions containing the itemset A. Frequent itemsets can be stored ahead of time in hash tables along with their counts so that they can be accessed quickly.

3. PREPARING TEXT FOR CATEGORIZATION

Text categorization is the automated assigning of natural language texts to predefined categories based on their content [5, 8, 14]. Data stored in most text databases are semistructured data in that they are neither completely unstructured nor completely structured [7]. During the first stage the full text of a document to be classified must be parsed to produce a list of potential features that could serve as a basis for categorization. Incomplete, noisy, and inconsistent data are commonplace properties of large real-world databases and data warehouses. In our experiments we do some sort of data preprocessing for quality decisions. We have the list of common words in a database file. If words of this list found in any input file, that are removed. Word stemming can either be performed by a morphological algorithm, which requires a lexicon and the morphological rules for the language [1, 6, 10], or can be approximated [3]. We do not follow the former way, hence we have to replace manually similar words by the stem word to compensate this limitation. We then find out association relations among these words. For using more linguistic knowledge one may extracts phrases from the document text [4]. Some users may like to find associations between pairs of keywords or terms from a given set of keywords or phrases, whereas others may wish to find the maximal set of terms occurring together. Therefore, based on user mining requirements, standard association mining or max-pattern mining algorithms may be evoked.

4. CLASSIFICATION USING DECISION TREE

Data classification model may be represented in various forms, such as classification (IF- THEN) rules, mathematical formulae, neural networks, or decision trees,. A decision tree is a flow-chart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions [9]. When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. Tree pruning methods address this problem of overfitting the data.

Quinlan's C4.5, a later version of the ID3 algorithm [13], is a set of computer programs that construct classification models by discovering and analyzing patterns. In the software C4.5, a decision tree is generated from a set of training cases [11]. The tree is validated through a set of test (unseen) cases. Our experiment use C4.5's the decision tree generator program for the classification analysis. The fundamental file provides names for classes, attributes, and attribute values. We separate and produce training and test sets randomly. The C4.5 program produce unpruned and pruned trees, analyzes and predicts about the data. C4.5 uses the training samples to estimate the accuracy of each rule. Since this would result in an optimistic estimate of rule accuracy, C4.5 employs a pessimistic estimate to compensate for the bias [12, 13].

5. EXPERIMENTAL RESULTS

In the way to text classification process, we choose abstracts of different conference papers as a source of experimental data. We select papers from proceedings of International Conference on Computer and Information Technology of the years 1998 and 2000 (ICCIT'98 and ICCIT 2000). We take total 33 abstracts. Of them 25 texts are selected for training decision tree and the rest 8 texts are for testing that tree. The papers are of four categories: Algorithm, Artificial Intelligence, Graph Theory and Pattern Recognition. The following table shows the distribution:

Class types	Training set number	Testing data set number
Algorithm	7	2
Artificial Intelligence	5	2
Graph Theory	7	2
Pattern Recognition	6	2
Total	25	8

We then clean those 33 files using the database of common words. Output database is created by data from one input file into one line (assumes one transaction) and in sorted order. We may have to do some cleaning manually. After preprocessing, we invoke Apriori algorithm to generate frequent candidate itemsets that are used to generate association rules satisfying both minimum support (5) and minimum confidence (60%). These rules are added with corresponding class. The C4.5 software (decision tree generator) is then invoked to generate decision tree. It produces a decision tree of size 7 (i.e. the sum of intermediate nodes and leaves is 7) from these 96 attributes of 25 cases. Figure 1 shows the output. The decision tree has 3 nodes and 4 leaves. The nodes are attribute 1, attribute 22 and attribute 9. The value '*yes*' for attribute 1 indicates Artificial Intelligence class type. While training, 5 texts are recognized correctly as Artificial Intelligence class. The value '*no*' for attribute 1 and '*yes*' for attribute 22 means Graph Theory class type. While training, 7 texts are recognized correctly as Graph Theory class and 1 text file is wrongly classified. The value '*no*' for attribute 1, 'no' for attribute 22 and '*yes*' for attribute 9 denotes Algorithm class type. 6 texts are trained correctly as Algorithm class type. The value '*no*' for attribute 1, '*no*' for attribute 22 and '*no*' for attribute 9 denotes Pattern Recognition class type. 6 texts are trained correctly as this class type.

Evaluation on training data (25 items) shows that the decision tree, before pruning, has 4% (1 out of 25 training data) errors. The tree remains same after pruning and it predicts 24.3% error in categorization using this decision tree.

Evaluating on test data (8 items), we have both the unpruned and pruned tree generate 12.5% error (1 out of 8 input text files). One Pattern Recognition type text file is wrongly classified as Algorithm class type.

If we use support 5 and confidence 75%, 48 association rules are produced. Using these rules as attributes in generating decision tree by *C45.exe* we have a decision tree of size 5. The second result (39.3% error in training cases, 55.3% prediction error, 37.5% error in test cases) is worse than previous one because there are fewer attributes than previous one, i.e. later rules are less discriminating (in categorizing texts). The tree cannot categorize Graph Theory class at all. The output of *C45.exe* is shown in figure 2.

When we select the support and confidence level at 5 and 0.60 respectably, because higher confidence level produces fewer association rules to discriminate texts and lower confidence level produces too many association rules to work with. Again, lower support level produces enormous rules to be used for attributes for decision tree generator. For example, with support level 4 and confidence level 100%, we get 839 association rules. On the other hand, higher support level produces very few rules. For example, with support level 6 and confidence level 55%, we get only 8 association rules, which is definitely cannot be used for categorization. The following table lists these data:

Support	Confidence	Number of association rules generated
4	1.00	839
5	0.60	96
5	0.75	48
6	0.55	8

```
C4.5 [release 5] decision tree generator

            Wed Mar 26 20:08:50 2003
---------------------------------------
   Options:
         Trees evaluated on unseen cases

Read 25 cases (96 attributes) from DF.data

Decision Tree:

ATT1 = yes: Artificial Intelligence (5.0)
ATT1 = no:
|  ATT22 = yes: Graph Theory (8.0/1.0)
|  ATT22 = no:
|  |  ATT9 = yes: Algorithm (6.0)
|  |  ATT9 = no: Pattern Recognition (6.0)

Tree saved

Evaluation on training data (25 items):

        Before Pruning          After Pruning
       ----------------   ---------------------------
       Size     Errors   Size     Errors   Estimate

          7    1( 4.0%)     7    1( 4.0%)    (24.3%)   <<

Evaluation on test data (8 items):

        Before Pruning          After Pruning
       ----------------   ---------------------------
       Size     Errors   Size     Errors   Estimate

          7    1(12.5%)     7    1(12.5%)    (24.3%)   <<

        (a)  (b)  (c)  (d)   <-classified as
        ---- ---- ---- ----
          2                  (a): class Algorithm
               2             (b): class Graph Theory
          1         1        (c): class Pattern Recognition
                         2   (d): class Artificial Intelligence
```

Figure 1: Output of C45.exe for support 5 and confidence 60%.

```
C4.5 [release 5] decision tree generator

            Wed Mar 26 20:15:04 2003
---------------------------------------
   Options:
         Trees evaluated on unseen cases

Read 28 cases (48 attributes) from DF.data

Decision Tree:

ATT2 = yes: Algorithm (9.0/2.0)
ATT2 = no:
|  ATT9 = yes: Pattern Recognition (14.0/8.0)
|  ATT9 = no: Artificial Intelligence (5.0/1.0)

Tree saved

Evaluation on training data (28 items):

        Before Pruning          After Pruning
       ----------------   ---------------------------
       Size     Errors   Size     Errors   Estimate

          5   11(39.3%)     5   11(39.3%)    (55.3%)   <<

Evaluation on test data (8 items):

        Before Pruning          After Pruning
       ----------------   ---------------------------
       Size     Errors   Size     Errors   Estimate

          5    3(37.5%)     5    3(37.5%)    (55.3%)   <<

        (a)  (b)  (c)  (d)   <-classified as
        ---- ---- ---- ----
          2                  (a): class Algorithm
                    1    1   (b): class Graph Theory
                    2        (c): class Pattern Recognition
                    1    1   (d): class Artificial Intelligence
```

Figure 2: Output of C45.exe for support 5 and confidence 75%.

That is why in our experiments we use support level 5 and confidence level 60% to 75%.

6. CONCLUSION & FUTURE WORK

There are many variations of the Apriori algorithm that focus on improving the efficiency of the original algorithm. For example: hash-based technique, transaction reduction, partitioning, sampling, dynamic itemset counting. Our experimental results can be made more impressive if we increase the number of attributes (i.e. association rules). This can be obtained by decreasing the support level and/or the confidence level. But in that case running time would increase. In fact, the more association rules we use, it does not necessarily mean that the number of input texts) will also improve the categorization. Yet there is another way to get better results. While producing input data file for C4.5 programs, we check all input files for an association rule and generate attribute value '*yes*' to the input text that contains any word that is in that association rule. Instead, we may follow more robust ways. For example, we can generate '*yes*' to the input text that contains fifty percent or more words of the association rule. This robustness will give more discriminating attributes and the decision tree built will categorize documents better. In the next attempt we hope to consider these suggestions.

REFERENCES

[1] Adriaans, Pieter and Zantige, Dolf. Data Mining, page 63, Addison Wesley Longman Singapore Pte. Ltd, 1999.

[2] Agrawal, R.; Mannila, H.; Srikant, R.; Toivonan, H.; Verkamo, A. Fast Discovery of Association Rules, Advances in Knowledge Discovery and Data Mining, 1996.

[3] Bayer, Thomas, Renz, Ingrid, Stein, Michael, & Kressel, Ulrich. 1996. Domain and Language Independent Feature Extraction for Statistical Text Categorization. Computation and Language, 7.

[4] David D. Lewis. Feature Selection and Feature Extraction for Text Categorization. Speech and Natural Language: Proceedings of a workshop held at Harriman, New York, February 23-26, 1992. Morgan Kaufmann, San Mateo, CA, pp. 212-217, 1992.

[5] Friedman, J. H. 1977. A recursive partitioning decision rule for nonparametric classification. IEEE Transactions on Computers, 404408.

[6] Gorniak, P. December 13, 1998. Sorting Email Messages by Topic.

[7] Han, Jiawei; Kamber, Micheline. Data Mining: Concepts and Techniques. Academic Press, United Kingdom, 2001.

[8] Hayes, P. and Weinstein, S. CONSTRUE/TIS: a system for content-based indexing of a database of news stories. In IAAI-90, 1990.

[9] J. Ross Quinlan. Induction of decision trees. Machine Learning, 1:81 – 106, 1986.

[10] Lewis, David D. 1992. Representation and Learning in Information Retrieval. Ph.D. thesis, University of Massachusetts.

[11] Mingers, J. 1989. An empirical comparison of selection measures for decision-tree induction. Machine Learning 3, 4, 319-342.

[12] Quinlan, J. R. 1986. Induction of decision tree, 81-106. Reprinted in J. W. Shavlik and T. G. Dietterich (eds.), Readings in Machine Learning. San Mateo) CA: Morgan Kaufmann, 1991.

[13] Quinlan, J. Ross. C4.5: programs for machine learning, Morgan Kaufmann Publishers, Sun Mateo, California,1988.

[14] Sundheim, B., ed., May 1991. Proceedings of the Third Message Understanding Evaluation and Conference, Morgan Kaufmann, Los Altos, CA.