Adapting Association Patterns for Text Categorization: Weaknesses and Enhancements

Tieyun Qian[1], Hui Xiong[2], Yuanzhen Wang[3], Enhong Chen[4]

[1] Wuhan University, qty@whu.edu.cn
[2] Rutgers University, hxiong@andromeda.rutgers.edu
[3] Huazhong University of Science and Technology, wyzh1999@371.net
[4] University of Science and Technology of China, cheneh@ustc.edu.cn

ABSTRACT

The use of association patterns for text categorization has attracted great interest and a variety of useful methods have been developed. However, the key characteristics of pattern-based text categorization remain unclear. Indeed, there are still no concrete answers for the following two questions: First, what kind of association patterns are the best candidate for pattern-based text categorization? Second, what is the most desirable way to use patterns for text categorization? In this paper, we focus on answering the above two questions. Specifically, we show that hyperclique patterns are more desirable than frequent patterns for text categorization. Along this line, we develop an algorithm for text categorization using hyperclique patterns. The experimental results show that our method provides better performance than state-of-the-art methods in terms of both computational performance and classification accuracy.

Categories and Subject Descriptors: H.2.8 [Database Management]:Database Applications - Data Mining

General Terms: Algorithms.

Keywords: Hyperclique Patterns, Text Categorization

1. INTRODUCTION

Text categorization is a key technique for processing and organizing text documents. Text categorization techniques are often used to classify news stories and to guide a user's search on the Web. Recently, there has been considerable interest in using association patterns [1] for text categorization [4, 7, 8]. This is known as Associative Text Categorization (ATC). A key benefit of ATC is to produce semantic-aware classifiers which include understandable rules for text categorization. While several interesting algorithms have been developed, further investigation is needed to characterize ATC with respect to the following two issues:

1. What kind of association patterns are the best candidate for associative text categorization?
2. What is the most desirable way to use association patterns for text categorization?

The goal of this work is to address the above two issues. Indeed, we illustrate that the hyperclique patterns [9] is a better candidate than frequent patterns [1] for text categorization, since they can provide much better coverage of objects and have a computational advantage. In addition, we develop an algorithm for text categorization using hyperclique patterns. In this algorithm, we explore two design issues. First, we exploit a new way of removing redundant rules. Second, we integrate feature selection into the rule-extraction procedure. Finally, our experimental results show that our approach has a better performance than existing methods in terms of both computational performance and the classification accuracy.

2. ALGORITHM DESIGN ISSUES

Vertical Pruning with General-to-Specific Ordering. Given two rules R_1: $I_1 \Rightarrow c$, and R_2: $I_2 \Rightarrow c$, R_1 is said more general than R_2 and R_2 is said a specific rule of R_1, if and only if I_1 is a subset of I_2. Also, there is a general-to-specific (g-to-s) ordering between R_1 and R_2. The g-to-s pruning strategy [4, 6] compares rules having g-to-s ordering and removes more specific but less accurate one. However, it is computationally expensive to check the g-to-s relationship between rules. If association patterns are stored in a prefix tree [6, 4], rules along both the vertical direction and the horizontal direction can exhibit super-subset relationships. The vertical check only needs one pass of depth first traversing over the whole tree for all branches while the horizontal check needs multiple passes traversing over the whole tree for each rule node in the tree. So, we only eliminate ineffective specific rules along vertical direction, and defer the pruning step along horizontal direction until the classification time. If multiple rules exhibiting g-to-s ordering cover the test instance simultaneously, we only select the best matching rule for the test instance. After the next sequential covering step in the training stage, the number of rules will be greatly reduced, so the testing time will not increase too much. In this way, we make a balance between the training and testing time.

Integrating Feature Selection with Rule Pruning Traditionally, feature selection has been used as an independent data preprocessing procedure. However, this tradition way to use feature selection is not effective for ATC, since the predetermined features may not be frequent items and can be pruned before they are used for text categorization. In our method, we integrate feature selection into the rule extraction procedure. Once the measures of each rule have been computed, the feature selection metric can be derived without extra computation as an additional procedure.

CIKM'06, November 5–11, 2006, Arlington,Virginia,USA.
ACM 1-59593-433-2/06/0011.

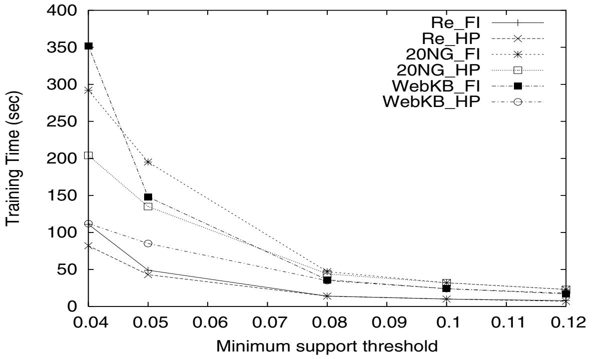

Figure 1: A Comparison of ATC using FIs and HPs

3. EXPERIMENTAL EVALUATION

We conducted experiments on three real-world data sets including *Reuters*, *20NewsGroup* (*20NG*), and *WebKB*, which are widely used in text categorization research [4, 7, 8]. All experiments were performed on a PC with 1.7GHz CPU and 1Gbyte of memory running Windows 2000.

3.1 The Choices of Association Patterns

Earlier research [5] has revealed that text categorization can benefit from larger vocabulary size. This is also true for associative text categorization. In other word, we have to increase the coverage of objects (vocabulary) by association patterns for a better performance of ATC. However, for frequent pattern mining, the increase of the coverage is at the cost of losing efficiency. Indeed, the cost of frequent pattern mining becomes extremely expensive if support thresholds are very low. To this end, we use hyperclique patterns (HPs) instead of frequent itemsets (FIs) as candidates for ATC. We choose HPs for two reasons: 1) HPs include objects which are strongly related to each other [9]. 2) HPs can be identified at very low levels of support thresholds, so HPs can provide better coverage of objects than frequent patterns.

Figure 1 shows that the training time of ATC-HPs is significantly less than that of ATC-FIs on all observed data sets. The computation savings become more dramatic when *minsup* decreases. This indicates that, with the help of hypercliques, we can get better results with a larger vocabulary size by setting lower support thresholds, especially for the sparse data set such as 20NG.

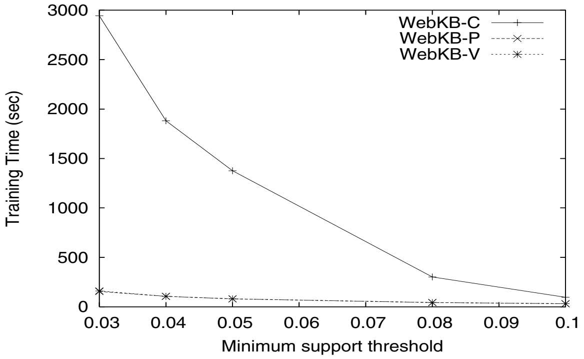

Figure 2: A Time Comparison of Pruning Methods

3.2 The Effect of Rule-Pruning Methods

Here, we evaluate the performance of the proposed rule-pruning method. In this experiment, the 'Pure' version does not have any *g-to-s* pruning. Also, the 'Complete' version has a complete pruning. Finally, the 'Vertical' version adopts a *g-to-s* pruning only along vertical paths in the prefix tree. We investigated the computational performance of the above three approaches. Due to space limitation, we only show the results on *WebKB* while similar results were also observed on Reuters and $20NG$. Figure 2 shows the training time of three approaches. We can see that the *complete* approach (curve C) is extremely time-consuming compared to the other two approaches. We also observe that the difference between *Pure* (curve P) and *Vertical* (curve V) is not significant.

Table 1: Best Results of ATC-HPs

Corpus	20NG	Reuters	WebKB
MicroAvg	75.8	92.6	90.2
MacroAvg	75.0	87.5	88.3
Training Time	691s	79s	105s
Testing Time	22s	10s	4s

3.3 A Performance Comparison

Table 1 shows the classification results using ATC-HPs on three data sets. In the table, we can observe that our classification performance on *Reuters* is much better than currently reported overall best results in [3]. Also, our result on $20NG$ is comparable to the value (around 76%) obtained in [10]. As for WebKB, we used C-SVC in LIBSVM [2] as the SVM tool and chose RBF as the kernel function. The best result of 89.68% by SVM is worse than that of ATC-HPs. More importantly, in addition to a training time of 328 seconds, SVM needs a feature selection procedure, which takes up to 83 seconds. In contrast, the total training time by ATC-HPs is only 105 seconds.

4. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, 1993.

[2] C.-C. Chang and C.-J. Lin. Libsvm. In *http://www.csie.ntu.edu.tw/ cjlin/libsvm/*.

[3] S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *CIKM*, 1998.

[4] J. Feng, H. Liu, and J. Zou. Moderate itemset fittest for text classification. In *WWW*, 2005.

[5] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *ICML*, 1997.

[6] W. Li, J. Han, and J. Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *ICDM*, 2001.

[7] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *KDD*, 1998.

[8] J. Wang and G. Karypis. Harmony: Efficiently mining the best rules for classification. In *SDM*, 2005.

[9] H. Xiong, P. Tan, and V. Kumar. Mining strong affinity association patterns in data sets with skewed support distribution. In *ICDM*, 2003.

[10] J. Yan, N. Liu, B. Zhang, S. Yan, and et al. Ocfs: Optimal orthogonal centroid feature selection for text categorization. In *SIGIR*, 2005.