Contents lists available at SciVerse ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Two scalable algorithms for associative text classification

Yongwook Yoon *, Gary G. Lee

*Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), San 31, Hyoja-Dong,
Pohang 790-784, Republic of Korea*

ARTICLE INFO

Article history:
Received 23 September 2009
Received in revised form 27 September 2012
Accepted 30 September 2012
Available online 16 November 2012

Keywords:
Association rule mining
Associative classification
Text categorization
Large-scale dataset

ABSTRACT

Associative classification methods have been recently applied to various categorization tasks due to its simplicity and high accuracy. To improve the coverage for test documents and to raise classification accuracy, some associative classifiers generate a huge number of association rules during the mining step. We present two algorithms to increase the computational efficiency of associative classification: one to store rules very efficiently, and the other to increase the speed of rule matching, using all of the generated rules. Empirical results using three large-scale text collections demonstrate that the proposed algorithms increase the feasibility of applying associative classification to large-scale problems.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Associative classification uses association rules which are mined from a transactional log database (Li, Han, & Pei, 2001; Liu, Hsu, & Ma, 1998). An association rule represents a co-occurrence relation among items in transactional logs. If an item in an association rule is a class label, the rule can be used for classification. Associative classifiers have been applied to various types of data, including text documents (Antonie & Zaïane, 2002; Li, Sugandh, Garcia, & Ram, 2007; Wang & Karypis, 2005; Yoon & Lee, 2008) and biological data such as DNA or protein sequences (She, Chen, Wang, & Ester, 2003). Whereas other types of classifiers including Naive Bayes and Support Vector Machine (SVM) consider only word features, associative classifiers can exploit *combined* features (for example, phrases) as well as words.

Words are elementary features in text classification. In a text collection, a document generally consists of hundreds of words. If a document is viewed as a sequence of word features, it exists in a very high-dimensional space. If phrase features are included as well as words, the dimensionality increases. A training database contains tens of thousands of documents, hence they become distributed sparsely in the document space. Therefore, raising the coverage for unseen test documents during the prediction phase is a big challenge. To match more test documents in associative classification, the length of word pattern in a rule needs to be reduced. Reducing the length of the pattern may degrade classification accuracy, although it can improve the coverage for test instances. Associative classifiers resolve this reduction in accuracy by choosing a smaller number of qualified rules and combining them in a predefined manner during the prediction phase (Antonie & Zaïane, 2002; Liu et al., 1998; Wang & Karypis, 2005). Promising results in text classification have been obtained by using a large number of low-order association rules, and applying a boosting technique to increase the classification accuracy (Yoon & Lee, 2008).

A large number of association rules can be mined by adjusting the mining parameters such as minimal support and minimal confidence to low values. However, too many association rules can hardly be processed in a real situation because they

* Corresponding author.
E-mail addresses: ywyoon@postech.ac.kr (Y. Yoon), gblee@postech.ac.kr (G.G. Lee).

0306-4573/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ipm.2012.09.003

require very long computation time and very large storage space. Especially, generating high-order[1] rules exacerbates this problem because the number of rules produced grows exponentially with the order. We propose two new algorithms to process such a huge number of rules very efficiently. One algorithm represents association rules in a compact format so that it can save a large amount of storage space when a larger number of rules are mined to build classifiers. The other algorithm uses a new data structure to conduct the classifier building process very quickly.

When a large-scale text collection is processed, the number of generated rules may exceed 10^6, so they must be written into a disk file due to limitations in main memory. This paper proposes a novel method of representing class association rules in a compact manner where the itemset of a rule, the antecedent part of the rule, is represented in a compact format using the information of previously generated rules. Basically, the proposed classification method requires a large number of rules while some associative classification methods produce a small number of rules from the beginning. Our classification algorithm (Yoon & Lee, 2008) can achieve a maximum performance by generating as many rules as possible, which means minimizing the loss of the given information. The saving of 1 Gbyte in the file size cannot be negligible in real situations.

To apply the mined rules to a classification task, some qualified rules should be chosen from the original generated rules. This process is called *classifier building* or *rule pruning* (Antonie & Zaïane, 2002; Liu et al., 1998; Wang & Karypis, 2005). The compact rule representation also help to perform this classifier building process efficiently. When the building process starts, the rules are loaded into memory again and are sorted in the order of confidence and support. They keep their compact form because only confidence and support information are needed in the sorting process (uncompressed antecedent information is required in the rule-matching process). Additionally, in-memory sorting requires an extra amount of memory. As the compact rules themselves occupy less memory space, the sorting or classifier building process can be performed more efficiently.

The second algorithm allows the process of rule matching against training documents to be performed quickly. To store training documents and to perform rule matching, we propose a new data structure which is specially designed for our classifier building process. One strong point of the second algorithm is the ability to delete documents from the structure efficiently as well as the matching speed. Using a naive matching algorithm takes $O(klNM)$ time, where k is the average length of a rule, M is the number of the rules, l is the average length of a test document, and N the number of the training documents. Our proposing algorithm can finish the process in one order of magnitude less time compared with the original method. Empirical results using large-scale document sets demonstrate that the proposed algorithms make associative text classification scalable to real-world problems.

This paper is structured as follows. Section 2 lists the previous studies which handled the efficiency issues related to the associative classification. Section 3 introduces associative text classification and formulates the problem. Section 4 describes a new model for representing association rules compactly, and explains the algorithm for efficiently matching rules to the training documents. Section 5 shows the experimental results using the proposed algorithms and compares them with the results obtained using previous methods. Finally, Section 6 concludes the paper.

2. Related works

Liu et al. (1998) introduced CBA, a prototype of associative classification. It adopts a two-stage induction process of classification rules. First, it generates Class Association Rules by *Apriori* method. Then, it applies each of rules to the set of training examples. The rules which classify correctly at least one example are selected and the covered examples are deleted from the database. While CBA deletes the examples when they are covered once, CMAR (Li et al., 2001) postpones the deletion until the examples are covered m times, which can improve the coverage of the selected rules. However, the postponement of the deletion means that the algorithm performs additional $m - 1$ times of rule matching against the example. In the associative classifier building process (Yoon & Lee, 2008) where our rule matching algorithm is used, there may be far more than m times of matching per training example, which requires a more efficient matching algorithm for an enormous number of association rules.

HARMONY (Wang & Karypis, 2005) keeps a rule with the highest confidence, which is assigned to each covering training instance. Without an additional pruning procedure, it can generate a small number of high-confidence rules very efficiently. However, such a one-rule-per-example principle may work adversely when handling a database with uneven distribution of class labels, because a class with many training examples may have a high prediction score on that class label, which leads to a wrong prediction.

Frequent closed itemset model (Pasquier, Bastide, Taouil, & Lakhal, 1999) was used for a compact representation for association rules. Without explicitly producing the subset rules, we can derive entire rule set from the closed form at a later time. Therefore, the rule mining process adopting the frequent closed itemset model can be finished within a less time and space than other mining methods. However, as far as we consider the classification task with the rules, it is a different story. In our classification method, we use all subset rules of a frequent closed rule because it is far more possible to match test documents if a rule has a smaller number of items. Normally, a rule in a closed itemset format is long and compact, and cannot be used in its own in our classification method; they should be expanded into its subset rules. In our rule mining method, a rule has a compact antecedent part and does not include any other subset rules in its compact form. From the beginning, we did not consider combining several rules into one because we mine rules for the purpose of classification.

[1] The order of a rule denotes the length of the word pattern.

Download English Version:

https://daneshyari.com/en/article/515528

Download Persian Version:

https://daneshyari.com/article/515528

Daneshyari.com