1999 Third International Conference on Knowledge-Based Intelligent Information Engineering Systems, 31st Aug-1st Sept 1999, Adelaide, Australia

A Data Structure for Dynamic Data Mining

Akihiro KITADA*, Tetsuya MURAI, and Yoshiharu SATO
Division of Systems and Information Engineering
Graduate School of Engineering, Hokkaido University
Kita 13, Nishi 8, Kita-ku, Sapporo 060-8628, JAPAN
{quitada, murahiko, ysato}@main.eng.hokudai.ad.jp

Keywords : Data mining, Association rules, Online mining, Binary trees

Abstract

A new data structure for online data mining using binary tree is proposed. Two experiments using synthetic data show efffectiveness of the method of data mining based on the proposed data structure.

1 Introduction

The recent progress of computer technology provides a method of analyzing a massive number of transaction data in database systems. Such direction has provoked various ways of KDD (Knowledge Discovery and Data mining) and, among them, a mining of the so-called *association rules*[1] obtains the wide-spread recognition as one of most active themes of data mining and several algorithms like AIS[1] and the Apriori[2].

However, these algorithms do not serve for the recent requirement of *online* mining of association rules where a set of transactions is frequently updated.For the purpose, Aggarwal et al.[3] proposed a method based on the adjacency lattice.

The purpose of this paper is to show that further improvement of the adjacency lattice to binary tree structure enables us to perform more effective online mining of rules with two experiments using synthetic data.

*Mr. Kitada's current address: Solutions Development COE Systems Headquarters, NTT DATA CORPORATION, Kayabacho Tower Bidg., 21-2, Shinkawa 1-chome, Chuo-ku, Tokyo 104-0033, JAPAN, quitada@mua.biglobe.ne.jp

0-7803-5578-4/99/$10.00©1999 IEEE

2 Association Rules

Let $\mathcal{I}$ be a finite set of items. A subset in $\mathcal{I}$ is called an *itemset*. Any itemset can be a (possible) *transaction*. A *database* $\mathcal{D}$ is defined as a set of transactions. For an itemset $X(\subseteq \mathcal{I})$, its *degree of support* $s(X)$ is defined by

$$s(X) \stackrel{\text{def}}{=} \frac{|\{T \in D | X \subseteq T\}|}{|D|},$$

where $|\cdot|$ is a size of a set.

Definition 1 (Agrawal et al.[2])

(1) *An* association rule *is an implication of the form* $X \Rightarrow Y$, *where* X *and* Y *are itemsets with* $X \cap Y = \emptyset$.

(2) *An association rule* $X \Rightarrow Y$ *holds in* $\mathcal{D}$ *with* confidence c $(0 \leq c \leq 1)$ *iff* $c = \frac{s(X \cup Y)}{s(X)}$.

(3) *An association rule* $X \Rightarrow Y$ *has a* degree of support s $(0 \leq \mathsf{s} \leq 1)$ *in* $\mathcal{D}$ *iff* $\mathsf{s} = s(X \cup Y)$.

Mining of association rules is actually performed by generating all rules that have certain minimum support (*minsup*) and minimum confidence (*minconf*) that a user specifies.

The problem of finding all association rules that satisfy user-specified minimum support and confidence consists of the following two-steps :

Step 1: Find all itemsets X such that

$$s(X) \geq minsup.$$

Such itemsets are called *large itemsets*. Let $\mathcal{L}_{\mathcal{D}}$ be the set of large itemsets given $\mathcal{D}$.

Step 2: For each itemset $L \in \mathcal{L}_{\mathcal{D}}$, find a pair

$$(X, L \setminus X)$$

such that

$$\frac{s(X \cup Y)}{s(X)} \geq minconf$$

as a rule

$$X \Rightarrow L \setminus X$$

for any $X \subseteq L$.

Step 1 requires much time when the size of $\mathcal{D}$ is massive. So the central problem of mining association rules is to make rapid algorithms of finding all itemsets.

3 Previous Works

3.1 Apriori

Agrawal et al.[2] proposed the Apriori algorithm for finding large itemsets. The first step in Apriori is to count the number of occurrences of each item and to find all large-1-itemsets whose degree of support is above *minsup*.

The kth step consists of the following two parts:

1. Generate the candidate itemsets $\mathcal{C}_k$ using the large itemsets in $\mathcal{L}_{k-1}$ found in the $(k-1)$th step.

2. Calculate $s(X)$ for each candidate itemset in $\mathcal{C}_k$.

3. The set $\mathcal{L}_k$ of large itemsets in the Kth step is defined as the set of candidate itemsets X whose degree of support $s(X)$ is at least *minsup*.

Finally the set L of large itemsets is defined by

$$\mathcal{L} \stackrel{\text{def}}{=} \bigcup_k \mathcal{L}_k.$$

Figure 1 shows an example of the Apriori algorithm.

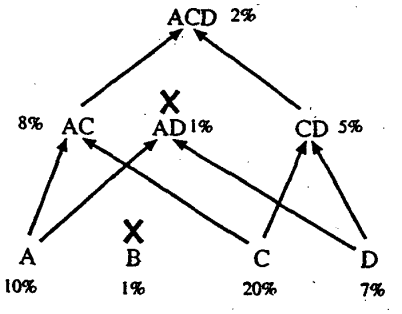

Figure 1: An example of the Apriori algorithm, where *minsup*=0.02

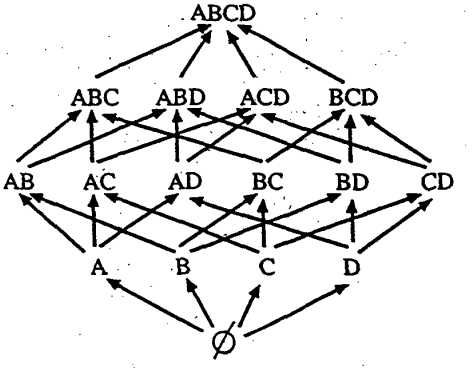

Figure 2: An example of adjacency lattice

3.2 Adjacency Lattice

We can easily understand that the Apriori algorithm actually constitutes a structure of lattice. In fact, Aggarwal et al.[3] revealed the effectiveness of using the so-called *adjacency lattice* for online-generation of association rules by referring degrees of support for each large itemsets on the lattice structure, where each node represent a candidate large itemset.

4 The Proposed Data Structure

In Aggrawal's method, in general, there are plural passes to reach a node (large itemset) when refer-

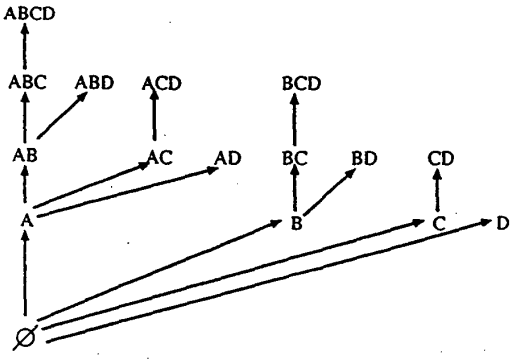

Figure 3: Reduced adjacency lattice

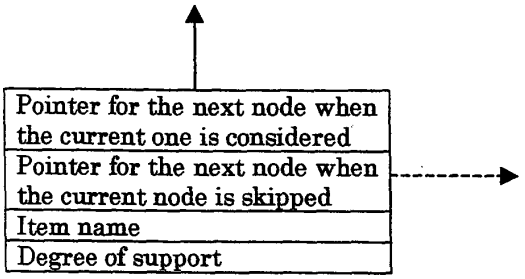

Figure 4: Data structure of nodes

ring its degree of support. So, it would be easier to deal with the structure if there is a unique pass to reach each node. For example, by introducing an order relation between nodes, the adjacency lattice in Figure 2 is reduced to the tree structure shown in Figure 3.

More precisely, when we introduce the data structure of nodes shown in Figure 4, the adjacency lattice in Figure 2 can be represented as the binary tree structure in Figure 5.

By the binary tree structure, we do not have to retain all relationship between nodes, saving memory capacity. For example, in order to refer $ABCD$, we can take the pass

$$NULL \uparrow A \uparrow B \uparrow C \uparrow D.$$

Similarly, to refer BCD, we can take

$$NULL \uparrow A \rightarrow B \uparrow C \uparrow D.$$

The running time of referring nodes is bounded

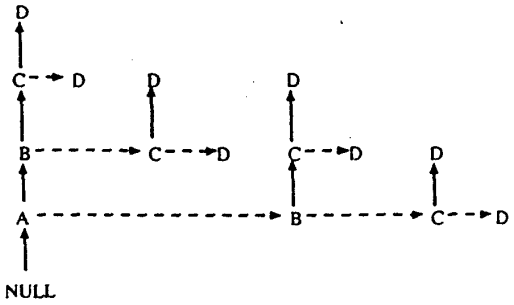

Figure 5: Binary tree structure

by $O(\log_2(n))$, where n is the size of the set of itemsets.

5 Experiments with Synthetic Data

The following two experiments were performed to compare the proposed method with the Apriori.

Static database: First we consider a static database case of constant size. As shown in Figure 6, the running time of the Airport increases inversely proportional to degrees of support, while that of the proposed method is almost constant. So we can conclude the proposed method is effective in cases of lower degrees of support.

Dynamic database: Secondly we consider a dynamic database case where a set of transactions is updated. As shown in Figure 7, the running time of the Apriori is directly proportional to the size of a set of transactions since it must recompute from the beginning whenever updating. On the other hand, the running time of the proposed method is almost constant because it has only to make a small change on a part of its binary tree structure with respect to updated transactions.

6 Concluding Remarks

In this paper, we showed that the effectiveness of binary tree structure in online association rules

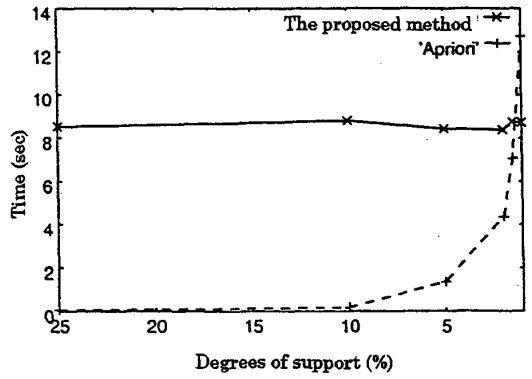

Figure 6: Static database

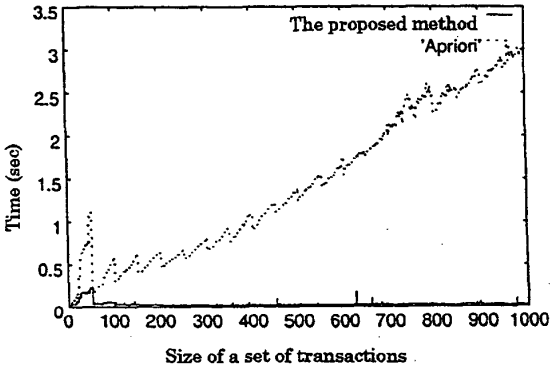

Figure 7: Dynamic database

in data mining by two experiments with synthetic data. The size of memory in the proposed method is reduced compared with the method using the adjacency lattice. Nevertheless, the proposed method requires $2^{\max(|T|)}$ nodes for the maximum size of transactions $\max(|T|)$, so further improvement should be done in future task.

References

[1] R. Agrawal, T. Imielinski, and A. Swami, Mining Association Rules between Sets of Items in Large Databases. Proceedings of ACM SIGMOD Conference on Management of Data, pp. 207-216, 1993.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, Fast Discovery of Association Rules. in U. M. Fayyad, G. Platetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.), Advances in Knowledge Discovery and Data Mining, AAAI Press / The MIT Press, pp. 307-328, 1996.

[3] C. C. Aggarwal and S. Y. Philip, Online Generation of Association Rules. Proceedings of the International Conference on Data Engineering, pp. 402-411, 1998.