AN INCREMENTAL UPDATING ALGORITHM FOR MINING ASSOCIATION RULES*

Xu Baowen* ** Yi Tong* Wu Fangjun* Chen Zhenqiang*
*(Department of Computer Science & Engineering, Southeast University, Nanjing 210096)
**(National Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072)

Abstract In this letter, on the basis of Frequent Pattern (FP) tree, the support function to update FP-tree is introduced, then an Incremental FP (IFP) algorithm for mining association rules is proposed. IFP algorithm considers not only adding new data into the database but also reducing old data from the database. Furthermore, it can predigest five cases to three cases. The algorithm proposed in this letter can avoid generating lots of candidate items, and it is high efficient.

Key words Data mining; Association rules; Support function; Frequent pattern tree

I. Introduction

Data mining is the nontrivial extraction of implicit, previously unknown, and potentially useful information from database, which has become a very important and extensive application in database and correlative domains. At present, mining association rules has become the most important, active and mature researching direction for the sake of its successful application in commercial research.

Most present algorithms for mining association rules are based on Apriori and AprioriTid (an improved version of Apriori) algorithm[1] proposed by R. Agrawal. To improve the efficiency, J.Han has proposed the Frequent Pattern (FP) growth algorithm[2] based on FP-tree. This algorithm can generate frequent items without candidate items, moreover it only needs to scan two passes over database to mine all association rules. Therefore, the great advantage of FP-growth algorithm is that it can save much time-space cost. The most disadvantage of this algorithm is that it does not take into account the high-efficient incremental updating problem of mining association rules, which seriously restrict actual application of this algorithm. To overcome this disadvantage, we introduces the concept of support function Z, and proposes an Incremental FP-tree (IFP-tree) based on FP-tree to resolve updating problem, then based on IFP-tree, we propose an IFP-tree growth (IFP-growth) algorithm to generate frequent items. Using this IFP-growth algorithm to mine association rules is very efficient.

II. Support Function

The purpose of mining association rules is to mine all strong rules in transaction database. Usually mining association rules is divided into two problems: (1) Based on the minimum support (i.e. support threshold), to find out all frequent items in transaction database; (2) Based on frequent items and the minimum confidence (i.e. confidence thresh-
old), to generate association rules. The first problem is the key because it can scale the standard of mining algorithm of association rule.

Commonly, the support specified by users can not change during mining process. However, to find undiscovered association rules, users must adjust the support and confidence threshold step by step according to their requirements. This is a dynamic alternate process. Whether an association rule interests users is closely related to users’ knowledge. As a result, we affirm that users have more go-aheadism in this process. In order to embody the dynamic alternate process and users’ go-aheadism, we introduce a concept of support function based on the concept of support to improve this circumstance. It is well known that all item-attributes are not entirely equal in transaction database, they can be distinguished into primary and secondary, such as relationship database has key item and non-key item. For all these, combining with practical circumstances of users, we define the support function $Z$.

The definition of support function $Z$ is as follows: Given a transaction database, which has $n$ item-attributes, support function is defined as $Z = f(x_1, x_2, \cdots, x_n)$, where $x_i (1 \leq i \leq n)$ is the item-variable. Combining with practical circumstances, users can restrict $m$ conditions as follows:

\[
\begin{align*}
\phi_1(x_1, x_2, \cdots, x_n) &= 0 \\
\vdots \\
\phi_m(x_1, x_2, \cdots, x_n) &= 0
\end{align*}
\]

For the nice property of continuum function in borderline, the support function $Z$ commonly is defined as a continuum function.

Continuum functions in borderline have two well-known theorems: the maximum value & minimum value theorems and the intervenient theorem. With the former theorems, we can gain the maximum value $\xi_1$ and the minimum value $\xi_2$ of the support function $Z$. Moreover the latter theorem can gain support between the maximum value $\xi_1$ and the minimum value $\xi_2$.

Based on the restricted condition, the support function $Z$ applies Lagrange method to resolve conditional extremum. We resolve all coordinate values under all conditions, at the end we can gain the maximum value $\xi_1$ and the minimum value $\xi_2$ of the support function $Z$. $\xi_2$ can be used to construct the original FP-tree. At the process of mining association rules, users can specify the value of item variables $x_1, x_2, \cdots, x_n$ whose values are stochastic. Then we resolve the dependent variable values, i.e., the temporarily specified minimum supports. In order to state conveniently, and for universality, in this letter, we define the minimum support as the minimum support count, and define the support of items as the value of item minus the minimum support count.

Support function is better than support in consistency and interactivity. The process of mining association rules focuses on users’ interest step by step. With the help of support function, users can stochastically define the arbitrary support threshold according to their requirements. Unlike the support, support function can change support threshold during mining, which embodies users’ go-aheadism. So support function is better than support in flexibility and can embody the dynamic alternative process better.

III. Incremental Mining Association Rules

D. W. Cheung, et al. first considered the high-efficient updating problem of mining