ATLANTIS PRESS

Advances in Computer Science Research, volume 83

8th International Conference on Social Network, Communication and Education (SNCE 2018)

An Improved Association Rule Algorithm Based on Indexed Array

Jianjing Mao[1,a] and Xianjing Zhang [2,b,*]

[1]Information Engineering Institute, Zhengzhou University of Industrial Technology, Xinzheng, 451100, China

[2]Information Engineering Institute, Zhengzhou University of Industrial Technology, Xinzheng, 451100, China

[a]20896445@qq.com,[b]1072746357@qq.com

* The Corresponding Author

Keywords: Association rule; FMAIG; Index array

Abstract. Through analyzing the existing problems in the classical Apriori algorithm and FP growth algorithm, the author proposes a frequent itemsets mining algorithm based on indexed array in the paper. On the basis of Apriori algorithm, this algorithm not only narrows down the set of candidates effectively through introducing indexed array, but also saves memory with no need of using FP storage structure. By analyzing and comparing the experiment, we can reach a conclusion that the algorithm can effectively improve the efficiency of mining frequent itemsets.

Summary

The most critical step in association rule mining is the mining of frequent item-sets. There are many mining algorithms for the mining methods of frequent item-sets, which are mainly based on Apriori algorithm [1] and FP-growth algorithm [2].The basic ideas of the two algorithms are as follows. The Apriori algorithm creates candidate sets by iteration, then by scanning the database, the non-frequent items of the candidate concentration are eliminated according to the minimum support requirement. The FP-growth algorithm, by generating frequent pattern trees, then filtering with minimal support, eventually generates frequent item-sets.

Apriori algorithm and FP-Growth algorithm have different degrees of defects. The drawback of the Apriori algorithm is that when the data width is large, the candidate assemblies increase exponentially, resulting in immeasurable I/O expense due to frequent scanning of the database [3].FP-growth algorithm, because of the adoption of FP tree structure, avoids the generation of large candidate sets and the time complexity of multiple scan databases [4].But for large databases, the deep deepening of FP- trees will result in significant memory consumption. In view of the above problems, this paper introduces the concept of index array on the basis of Apriori algorithm, and proposes a method for generating frequent item sets based on two-tuples--FMAIG(Frequent Item sets Mining Algorithm based on Ix Group).It can reduce the number of candidate sets and reduce the memory space occupation, which can effectively improve the mining efficiency of frequent item-sets.

FMAIG Algorithm

Index Array. The elements of the index array Ix[] are represented by $(\mathrm{F}_1,\ \mathrm{g}(\mathrm{C}_2-\mathrm{F}_2))$. Among them, $\mathrm{F_i}$ stands for frequent i - item sets, C_i stands for candidate i – item-sets, C_2-F_2 represents non frequent 2 item-sets. The following example illustrates the composition of an indexed array.

For example: suppose the frequent 1- item-sets are {a}, {c}, {d}, {f}, and non frequent 2-item-sets are {a, d}{a, f}{c, d}, so the two tuple of Ix[] is shown in Table 1.

Table 1 Ix[] index array

Item	Ix[]
a	d, f
c	d
d	a, c
f	a

FMAIG Algorithm Implementation. The algorithm is based on $F_{K-1} * F_1$ algorithm, and is filtered by $g(C_2 - F_2)$, which avoids the generation of redundant candidate effectively and reduces the repetition rate of candidate sets. First, F_k elements are generated by linking F_{k-1} and F_1 (according to a priori principle [5] and its inference, we know that the items in the join portfolio should be frequent). That element is not equal to the elements in the F_{k-1}, and is not included in the corresponding $g(C_2 - F_2)$. It can be seen from the generation process of Ix[] that the candidate items that do not meet the requirements of the upper class will not be in the frequent item-sets in the lower class (Such as the I frequent item-sets, its superset cannot be frequent). The number of candidate item sets can be reduced effectively by Ix[] test, thus reducing the number of database scanning.

FMAIG Algorithm Analysis

The FMAIG algorithm mainly reduces the redundant degree of the candidate item set and reduces the number of secondary connection items by using Apriori algorithm. So the number of scans of the transaction database is reduced when the calculation support is made.

Based on the priori principle and its inference, the index array is generated by Ki (frequent i-item set), non-frequent i +1- item set. The sub items in arbitrary non frequent i- items can not appear simultaneously in the frequent i+1- item sets. For $\forall I_{jk}、I_{jm} \in g_i(C_i\text{-}F_i)\{I_{j1},I_{j2},I_{j3},\ldots\ldots I_{jk}, I_{jm}, I_{ji}\}$,not exist $F_{i+1} \cup F'_{i+1=} \{I_{jk}I_{jm}\}$,therefore, before generating candidate sets, the number of candidate sets can be effectively reduced by detecting the factors of the index array.

The Performance Evaluation

In the following experiments, the test data is part of the data extracted from the T10I4D100K data set in the data resource library (http://fimi.cs.helsinki.fi/data/). In Figure 1, the 800 1200 2400 3200 4800 6000 7000 record is extracted under the support of 6%, and the running time of Apriori algorithm and FMAIG algorithm is compared. In Figure 2, 9000 records are extracted from T10I4D100k data set, and the minimum support level is set to show the extraction of 8000 records (9%, 12%, 16%, 18%, 24%) . The aim is to compare the running time of Apriori algorithm and FMAIG algorithm respectively.

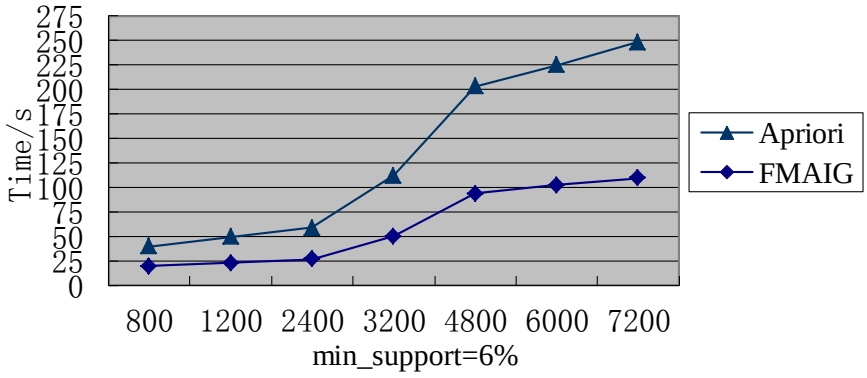

Fig.1 Comparision of running time of Apriori and FMAIG under the same support degree

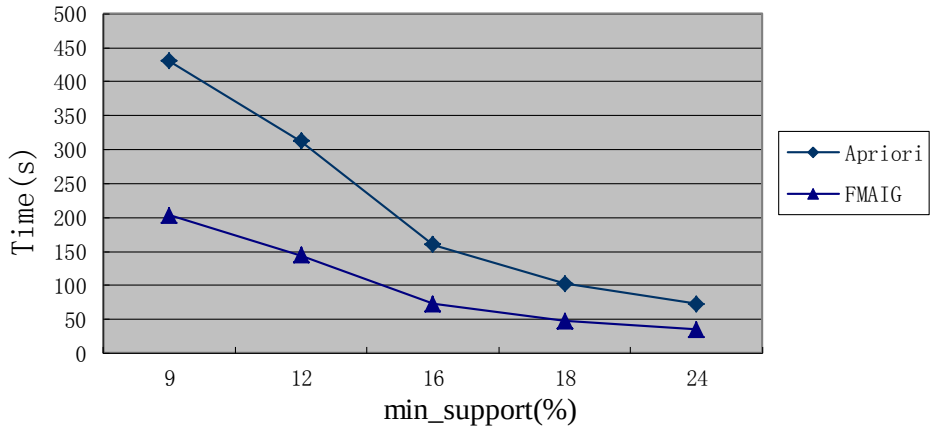

Fig.2 Comparision of running time of Apriori and FMAIG under the different support degree

The following conclusions can be drawn from the comprehensive analysis of the above charts. When the minimum support is a certain value, the running time performance of the FMAIG algorithm will be significantly improved as the number of transactions increases compared with the Apriori algorithm. However, when the mining database capacity is certain, the running time advantage of the FMAIG algorithm relative to the Apriori algorithm is shown when the minimum support setting gradually decreases. Thus, the time performance of FMAIG algorithm is higher than that of Apriori algorithm.

Reference

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules[A]. Proc. 2014 Int'l Conf. Very Large Data Bases (VLDB'94) [C].Santiago: 2014.487-499.

[2] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation [A].Proceedings of 2012 ACM-SIGMOD Int'l Conf. on Management of Data(SIGMOD'00) [C]. Dallas, TX:2012.1-12.

[3] Pang-Ning Tan Michael Steinbach Vipin Kumar, et al. Introduction to Data Mining[M]. Posts & Telecom Press 2015.209-221.

[4] QIN Liangxi, SHI Zhongzhi .New Flow Association Rules Mining Based on Ice berg Queries[J].Computer Engineering 2015,31(7),9-11.

[5] Jiawei Han Micheline Kamber. Data Mining Concepts and Techniques[M].Beijing: Machinery Industry Press, 2013:157-161.

[6] Wang Hong. Association Rules Mining in personnel management information [J]. Journal of Tianjin Normal University. Twenty-fourth volume, second issue.2014.6.