Advances in Intelligent Systems Research, volume 147

International Conference on Network, Communication, Computer Engineering (NCCE 2018)

Research on the Factors of Frequent Itemset Mining Based on Dynamic Hashing

Hanyu Hu [a)], Yunli Chen

Beijing University of Technology, Beijing 100124, China.

[a)] Corresponding author: 305209793@qq.com

Abstract. In order to improve the efficiency of association rules mining frequent itemsets, some people proposed using hashing techniques to mine association rules, it can greatly improve the time and space efficiency of frequent item sets. the space-time efficiency of the classical Apriori algorithm and the use of hashing techniques for association rule mining are analyzed, and the influence of various factors on the use of hashing techniques is analyzed. Aiming to improve the time-space efficiency of the algorithm, a dynamic hash algorithm based on data attributes is used to improve association rules mining. Experimental results verify the correctness and effectiveness of the algorithm.

Key words: Frequent itemsets, Dynamic hashing, Number of transaction items, The average length of the transaction, Database transaction volume, Minimum support.

INTRODUCTION

With the advent of big data era, Data mining has an increasing impact on people's lives. Association rule mining [1] is an important part of data mining, it is a rule-based machine learning algorithm.The algorithm can find mutual relations in big data. Its purpose is to use some attribute criteria to identify strong rules that exist in the database. So, it is an unsupervised machine learning method. Association rule mining algorithm is not only applied to shopping analysis, but also widely applied to user recommendation association, network Safety, daily production, supermarket shopping malls and other occasions. The Apriori algorithm [2] is an original algorithm for mining frequent itemsets of association rules.By limiting candidates to generate frequent item sets, an iterative method of layer-by-layer search is used [3].Until now, Apriori has been discussed and studied as a classic algorithm for mining association rules. Many scholars have conducted extensive research on the mining of association rules.

PROBLEM ANALYSIS

Association Rules Mining

The association rule mining process mainly includes two stages: the first stage is to find all the high-frequency item sets from the collection, and the second stage is to generate the association rules from these high-frequency item sets. the high-frequency item set found in the first stage refers to the fact that the frequency of occurrence of a certain project group must reach a certain level with respect to all records, the frequency of occurrence in a project group is called support. The second stage is to generate the association rule from the high-frequency item set. At the same time, the rules that satisfy the minimum supportability threshold and the minimum confidence threshold are called strong association rules. the set of items that meet the minimum support threshold limit is called a frequent itemset. The most important part of determining the performance of association rules mining is mining frequent itemsets [4].

Apriori Algorithm

The Apriori algorithm is the most influential algorithm for mining frequent itemsets of Boolean association rules. Its core is a recursive algorithm based on the idea of two-stage frequency set. The association rule belongs to the Boolean association rule in classification. Here, all item sets with support greater than the minimum support are called frequent item sets and are called frequency sets.

The basic idea of this algorithm is to first find all the frequency sets. The frequency of occurrence of these itemsets is at least the same as the predefined minimum support. then a strong association rule is generated from the frequency set, and these rules must satisfy the minimum support and the minimum reliability. Then use the frequency set found in step 1 to generate the desired rule, producing all the rules that contain only the items of the set, where there is only one item in the right part of each rule. Here, the definition of the medium rule is used. Once these rules are generated, only those rules that are greater than the user's given minimum credibility are left. In order to generate all frequency sets, a recursive method is used.

HASH-BASED APRIORI ALGORITHM

Apriori_Hash Algorithm

In order to improve the performance of the Apriori algorithm, many scholars have conducted extensive research on the basis of hashing techniques [5] and proposed some improved algorithms [6].In order to address the pruning of the Apriori algorithm and the problem of frequently scanning datasets, The literature [7] proposes to use the hash table to store the frequent 2 itemsets and only need to execute once, then the same pruning operation of the Apriori algorithm can be completed in O(1) time. The literature [8] proposes to adopt the address based on the candidate option Lk. Greek mapping method, When the frequent K item set Lk is generated from the candidate (K-1) item set in Ck-1, Lk-1 in each transaction set of Ck-1 is sequentially connected to generate all K item sets and will have the same the candidate Lk of the address is hashed to the corresponding storage queue in the hash table. Reference [9] mentions dynamically selecting two methods for generating candidate sets. The final effect is better than using only one of them.

DYNAMIC UTILIZATION OF HASHING TECHNIQUES TO PRODUCE CANDIDATE SET FACTORS

Number of Transaction Items

When the hashing technique generates a K-candidate set, each transaction in the database is decomposed into a K-item set and then put into a bucket. When producing a 2-item set, the efficiency is higher.

As K increases, the K-item set that needs to be placed in the bucket also gradually increases. For example, for a 20-item transaction, you get C_{20}^{2}=380 2-items sets. Candidate sets increase exponentially within a certain range. So, although a part of the bucket will be excluded, the overhead of getting the candidate set is also very large, which affects the efficiency. On the contrary, only the connection method is used to generate the candidate set. The number of transaction items will greatly affect the time and space efficiency of connection steps, so when using hashing techniques to generate candidate sets, the number of transaction items should be taken into account instead of using the hash to generate the candidate set.

The Average Length of the Transaction

In general experiments, people often use experiments with uniform average length and small average length to perform experiments. When using the hashing method to generate candidate item sets, each transaction is decomposed, resulting in the efficiency of the candidate set and the transaction used. In fact, there is a big relationship between features. When the data transactions used are usually small in length or more consistent, the use of a hashing method will indeed greatly increase the efficiency of generating candidate sets, but as the average length gradually increases and the average length becomes inconsistent How will the efficiency of generating candidate sets change? It is also necessary to conduct experiments to verify.

Database Transaction Volume

The transaction volume of the database is the most important factor affecting the running time of the algorithm and will directly affect the space and time efficiency of the algorithm. We will maintain the consistency of other variables when testing the impact of the database's transaction volume on the running time of the algorithm.

Minimum Support

Whether it is the Apriori algorithm or the hash-based method, the runtime overhead of the algorithm will decrease as the minimum support increases. The higher the minimum support, the fewer candidates will be generated and the fewer times the database will be scanned. For the specific impact, we will test data sets with different degrees of support.

EXPERIMENTAL RESULTS AND ANALYSIS

In order to further verify the influence of different factors on the mining of frequent itemsets using hashing technology, Apriori algorithm and Apriori_Hash algorithm are implemented in C++ with 2GB memory, CPU with intercore i7 3.60GHz and operating system Windows7.The experimental data refers to dataset data in machine learning.

As shown in FIGURE 1, under the same conditions of other factors, several groups of data with the increase in the number of transaction items, Apriori_Hash algorithm running time is less than the Apriori algorithm, but when the number of transaction items exceeds 15, the two algorithms run the time is greatly increased, and the running time of the Apriori_Hash algorithm even surpasses the Apriori algorithm, and the running time of the Apriori_Hash algorithm increases more significantly.

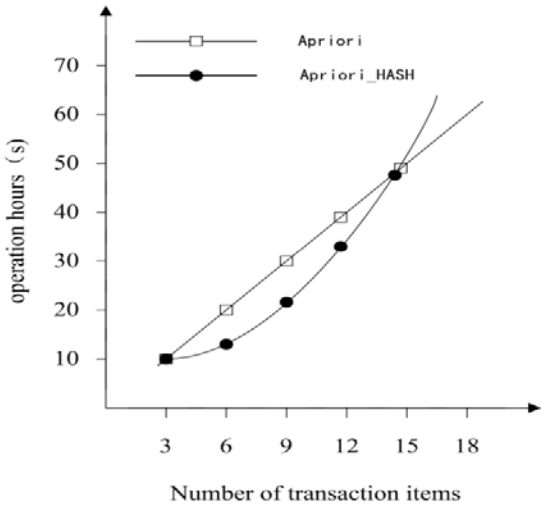

FIGURE 1. Effect of transaction item number on two algorithms

In FIGURE 2, the minimum support is 0.05. With the increase of the average transaction length, the running time of the algorithm increases with the number of data. Before the average transaction length is 8, the Apriori_Hash algorithm runs better than the Apriori algorithm. However, the gap is not large, but after the average length is greater than 8, the running time gap between the two algorithms is increasing. The improvement of the efficiency of the Apriori_Hash algorithm is even more pronounced.

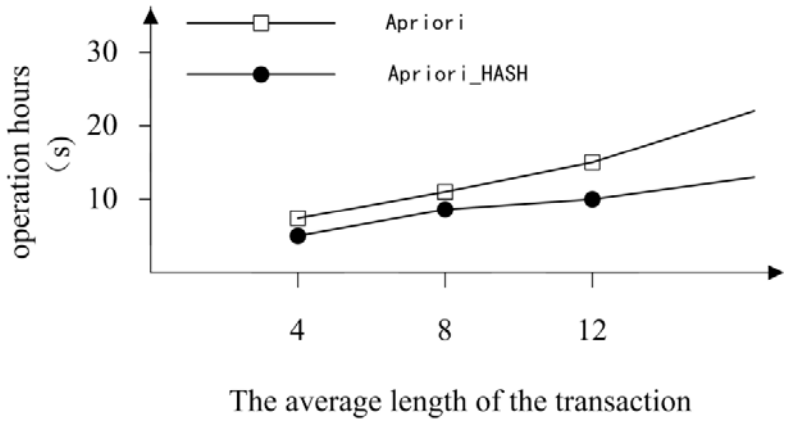

FIGURE 2. Effect of average transaction length on two algorithms

As shown in FIGURE 3, the minimum support is 0.05 and the average length of the transaction is 4. As the number of transactions increases, the running time of the Apriori_Hash algorithm is smaller than that of the Apriori algorithm.

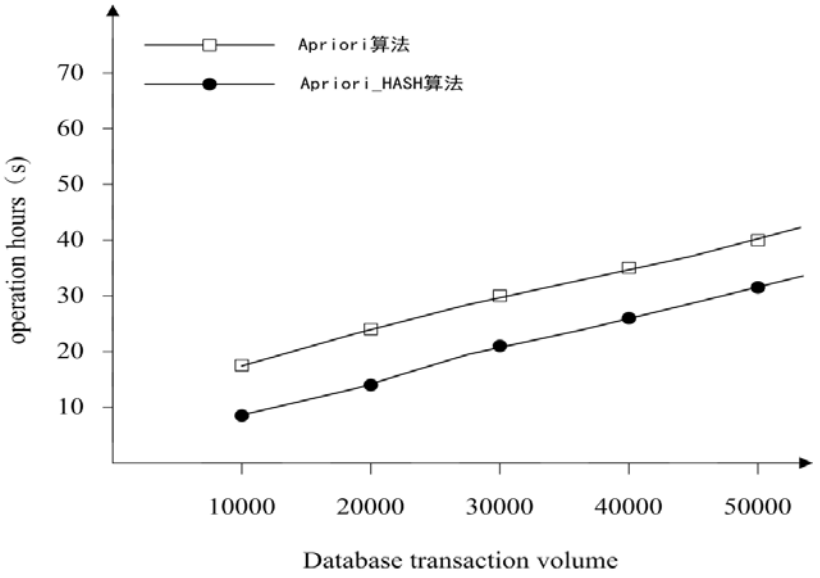

FIGURE 3. Influence of database transaction volume on two algorithms

As shown in Figure 4, under the same conditions with other factors, it can be seen that the running time of the two algorithms is decreasing with the increase of minimum support, and the Apripri_Hash algorithm is superior to the Apriori algorithm, and the minimum support is greater. At this time, the gap between the running time of the two algorithms is smaller. When the minimum support is reduced to 0.05, the running time of the algorithm increases exponentially and loses its significance.

The following conclusions can be drawn from experiments and analysis:

The Apripri_Hash algorithm greatly improves the efficiency of the Apriori algorithm generating candidate set. When the number of items is less than 15, the Apriori_Hash algorithm is significantly better than the Apriori algorithm. However, after a large-scale outbreak of candidate sets, the Apriori_Hash algorithm should not be used to generate the candidate set.

As the average transaction length increases, Apriori_Hash algorithm can greatly overcome the Apriori algorithm candidate set burst problem, and its efficiency becomes more and more obvious as the average length of the transaction increases.

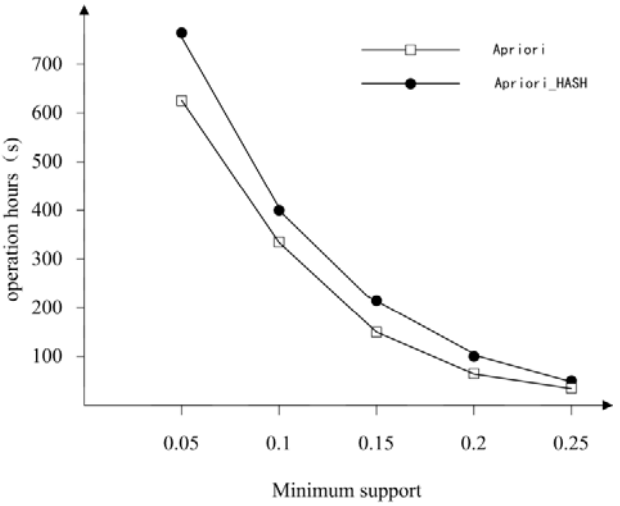

FIGURE 4. The effect of minimum support on the two algorithms

As the affairs of the database transaction volume increases, the Apriori_Hash algorithm is more efficient than the Apriori algorithm, and the gap between the two algorithms also increases as the transaction volume increases.

Apriori_Hash algorithm is obviously better than Apriori algorithm when the support degree is 0.05-0.1. When the support degree exceeds 0.1, the two algorithms have little difference in time efficiency.

CONCLUSION

The efficiency of generating candidate sets is a key issue that affects the mining of association rules.

This paper fully analyzes the factors that restrict Apriori algorithm and Apriori_Hash algorithm based on Hash technology. When the Apriori algorithm is implemented by using hashing technology, we should pay attention to the number of itemsets in the dataset, the average length of the transaction, the database transaction volume and the minimum support, and then consider which method to use to generate the candidate set.Although the Apriori_Hash algorithm is superior to the Aprori algorithm in most cases, the Apriori_Hash algorithm has drawbacks in terms of the number of item sets and the minimum support, and sometimes the efficiency is greatly reduced, and various factors of the data should be integrated. The application of hashing technology to the generation of candidate sets can be used to mine frequent itemsets more efficiently.

REFERENCES

1. Lai S Q, Zhu J P. A Survey of Association Rule Mining Algorithms in Data Mining[J]. Statistics & Information Tribune, 2005.
2. Agrawal R, Srikant R. Fast algorithms for mining association rules[M]// Readings in database systems (3rd ed.). Morgan Kaufmann Publishers Inc. 1996:2299-308.
3. Taniar D, Taniar D, Rusu L I. Strategic Advancements in Utilizing Data Mining and Warehousing Technologies: New Concepts and Developments[M]. Information Science Reference - Imprint of: IGI Publishing, 2009.
4. Leung K S, Mackinnon R K. Fast Algorithms for Frequent Itemset Mining from Uncertain Data[C]// IEEE International Conference on Data Mining. IEEE, 2015:893-898.
5. Peng Y, Xiong Y. Study on optimization of AprioriTid algorithm for mining association rules[J]. Computer Engineering, 2006.
6. Huang C. An Algorithm for Mining Association Rules Based on Hash Pruning and Redundant Transaction Compression[J]. Computer Engineering, 2003.
7. Balasubramanie P, Krihsna P V. A Hash based Mining Algorithm for Maximal Frequent Item Sets using Linear Probing[J]. Infocomp Journal of Computer Science, 2009(1).
8. Wang, Xxuehua. Hash table-based keyword mapping processing method and device, Wo 2014187040 A1[P]. 2014.

9. Huang C. An Algorithm for Mining Association Rules Based on Hash Pruning and Redundant Transaction Compression[J]. Computer Engineering, 2003.