Improving Association Rules Mining by Hashing Algorithm

Nang Khine Zar Lwin, Nay Min Tun
University of Computer Studies, KyineTong
nangkhinezarlwin@gmail.com ; naymin.300777@gmail.com ;

Abstract

Association rule mining, is one of the most important and well researched techniques of data mining. It is the process of discovering large itemsets appeared in a sufficient number of transactions. Large itemsets from a huge number of candidate large itemsets are dominating factor for the overall data mining performance. This paper presents mining association rules among items in a large database of sales transactions. To address this problem, it present an effective hash-based algorithm for the candidate set generation. This system applies an algorithm DHP (Direct Hashing and Pruning) on application cosmetic sales data to generate frequent association patterns. Generation of smaller candidate sets enables to effectively trim the transaction database size at a much earlier stage of the iterations, thereby reducing the computational cost of later iterations significantly. The experimental results of our system are also discussed in this paper.

1. Introduction

Database mining has attracted a growing amount of attention in database communities due to its wide applicability in retail industry to improving marketing strategy. The progress in barcode technology has made it possible for retail organizations to collect and store massive amounts of sales data. A record in sales data typically consists of transaction date, items bought in that transaction and sometimes customer id. Analysis of transaction data can provide very valuable information on customer buying behavior, and thus improve the quality of business decisions.

Association rule mining finds interesting association among a large set of data items. Many industries are becoming interested in mining association rules from their databases. The discovery of interesting association relationships among huge amounts of sales data improves business decision making process. Since the amount of sales data is huge, it is important to implement efficient algorithms to conduct mining on these data.

Mining association rules can be decomposed into two sub-problems. First, it is needed to identify all sets of items (itemsets) that are contained in a sufficient number of transactions above the minimum support (requirement). These itemsets are referred to as large itemsets. Once all large itemsets are obtained, the desired association rules can be generated in a straightforward manner. Second, mining association takes time to generate candidate itemsets in a large database.

This paper presents an algorithm DHP (Direct Hashing and Pruning) for efficient large itemset generation. It has two major features: one is efficient generation for large itemsets and the other is effective reduction on transaction database size. By using a hash technique, DHP is very efficient for the generation of candidate large itemsets. In addition,

DHP employs effective pruning techniques to reduce the transaction database size. Generation of smaller candidate sets by DHP enables to effectively trim the transaction database at a much earlier stage of the iterations, thereby reducing the computational cost for later iterations significantly.

This paper is organized as follows. Section 1 is the introduction, section 2 is related work. Process of association rule mining is presented in section 3. In section 4, improving Apriori of Association Rule Mining by DHP algorithm is described. Section 5 is the proposed system design and section 6 is the system implementation and sample case study for DHP algorithm. Section 7 is the conclusion and future work of the system.

2. Related Work

There are various algorithms to discover the large itemsets. Most algorithms first construct a candidate set of large itemsets based on some heuristics and then discover the subset that indeed contains large itemsets. This process can be done iteratively the sense that the large itemsetes discovered in one iteration will be used as the basis to generate the candidate set of the next iteration. For the iteration, all large itemsets example, in [7], at the kth iteration, all large itemsets containing k items, referred to as large k-itemsets, are generated. In the next iteration, to construct a candidate set of large (k + 1)-itemsets, a heuristic is used to expand some large k-itemsets into a (h + 1)-itemset, if certain constraints are satisfied. The main problem in these algorithms is the performance in the candidate generation process.

Han et al. developed the FP-growth algorithm [2] that is based on frequent pattern tree. Comparing with Apriori algorithm, this algorithm has following features. (1) It uses FP-tree to store the main information of the database. The algorithm scans the database only twice, avoids multiple database scans and reduces I/O time. (2) It does not need to generate candidates, reduces the large amount of time that is consumed in candidates generation and test. (3) It uses a divide-and-conquer approach in the mining process, so the searching space is significantly decreased. The efficiency of the FP-growth algorithm is better than Apriori algorithm. However, FP-trees consume much more memory and more time.

Another performance related issue is on the amount of data that has to be scanned during large itemset discovery. Directed Hashing and Pruning has two major features: one is efficient generation for large itemsets and the other is effective reduction on transaction database size. It is very efficient for the generation of candidate large itemsets, in particular for the large 2-itmsets, where the number of candidate large itemsets generated by DHP is smaller than that by previous methods, thus greatly improving the performance bottleneck of the whole process. In addition, DHP employs effective pruning techniques to progressively reduce the transaction database size. Hash partitioning is the most effective for applications requiring Association rule mining.

3. Association Rule Mining

Association rule mining is one of the most important and well researched techniques of data mining. It aims to extract interesting correlations, frequent patterns,. associations or casual structures among sets of items in the transaction databases or other data repository The discovery of association relationship among a huge database has been known to be useful in selective marketing, decision analysis, and business management popular area of applications is the market basket analysis, which studies the buying behaviors of customers by searching for sets of items that are frequently purchased together (or in sequence). [1]

Let I= $\{i_1, i_2 \ldots i_m\}$ be a set of literals, called items. Let D be a set of transactions, where each transaction, T is a set of items such that $T \subseteq I$. Each transaction is associated with an identifier, called TID. Let X be a set of items. A transaction T is said to contain X if and only if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \phi$.The rule $X \Rightarrow Y$ holds in the transaction set D with confidence c if c% of transactions in D that contain X also contain Y.

The rule $X \Rightarrow Y$ has support s in the transaction set D if s % of transactions in D contains $X \cup Y$. Mining association rules is composed of the following two steps –

- Discover the large itemsets, i.e., all sets of itemsets that have transaction support above a predetermined minimum support s.
- Use the large itemsets to generate the association rules for the database.

The overall performance of mining association rules is in fact determined by the first step.

3.1. Apriori Algorithm

In Apriori, in each iteration, it constructs a candidate set of large itemsets, counts the number of occurrences of each candidate itemset, and then determines large itemsets based on a pre-determined minimum support. In the first iteration, Apriori simply scans all the transactions to count the number of occurrences for each item. The meaning of various parameters is as follow:

Dk	Set of transactions for large k-item sets
|Dk|	No. of transactions in Dk
Ck	Set of candidate k-itemsets
Lk	Set of Large k-itemsets

Apriori algorithm consists of two steps – join and prune actions.

Join Step – It is the candidate generation process. To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} with itself. This set of candidates is denoted Ck.

Prune Step – C_k is a superset of L_k, that is, its members may or may not be frequent, but all of the frequent k-itemsets are included in Ck. A scan of DB to determine the count of each candidate in Ck would result in the determination of L_k. if the count of candidate in L_k is less than minimum supports, it cannot be frequent and so can be removed from C.

4. Hashing and Pruning

This algorithm utilizes a hash method for candidate itemset generation during the initial iterations and employs pruning techniques to progressively reduce the transaction database size.
It has two major features:

- one is efficient generation of large itemsets
- The other is effective reduction on transaction database size.

As illustrated in Apriori, in each pass the set of large item sets, L_i, is used to form the set of candidate large item sets C_{i+1} by joining L_i, with L_i on (i - 1). Then the database is scanned and it will count the support of each itemset in C_{i+1} so as to determine L_{i+1}. As a result, in general the more itemsets in C_i, the higher the processing cost of determining L_i will be. [5]

By constructing a significantly smaller C_2, DHP can also generate a much smaller D_3 to derive C_3. After this step, the size of L_i decreases rapidly as i increase. A smaller L_i leads to a smaller C_{i+1}, and thus a smaller corresponding processing cost. Algorithm DHP is designed to reduce the number of itemsets to be explored in C_i in initial iterations significantly. The corresponding processing cost to determine L_i from C_i is therefore reduced.

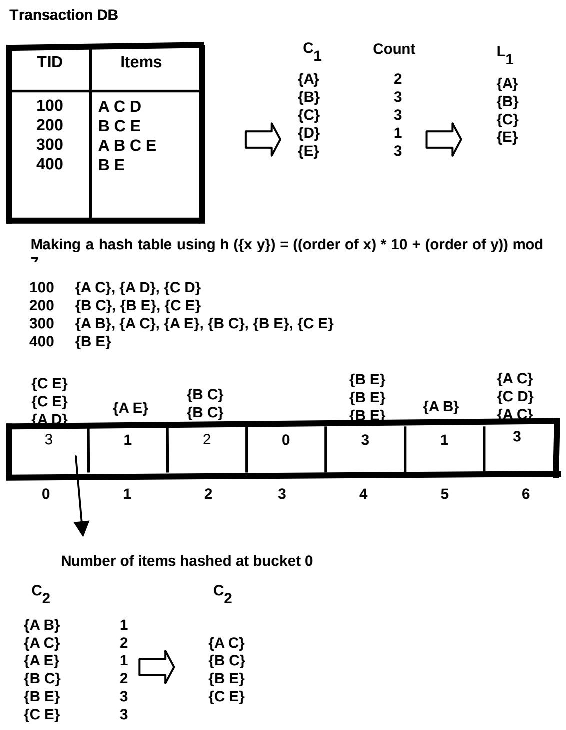


Figure 1: DHP Algorithm

DHP reduces the database size progressively by not only trimming each individual transaction size but also pruning the number of transactions in the database. Any subset of a large itemset must be a large itemset by itself. This fact suggests that a transaction be used to determine the set of large (k+1)-itemsets only if it consists of (k+1) large k-itemsetes in the previous pass. Hash table is generated according to hash algorithm h({x y}) = ((order of x) * 10 + (order of y)) mod 7, where 7 stands for the bucket count and it will be constant in this system. Bucket count can be variable based on the itemset size.

Assigning candidates are calculated using above hash algorithm.

For example, for candidate {A B}

h ({A B}) = ((1 * 10) + 2) mod 7 = 5,

So candidate {A B} is assigned at bucket # 5.

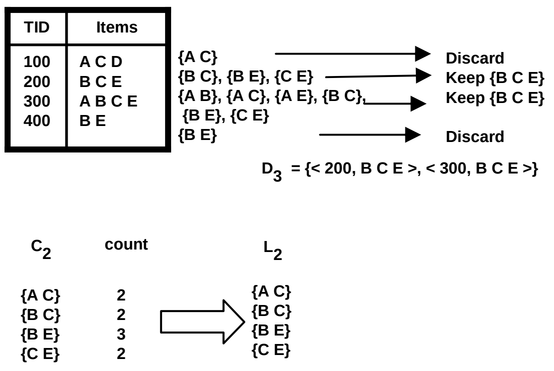


Figure 2: Sample Reducing Database

4.1. Process of DHP Algorithm

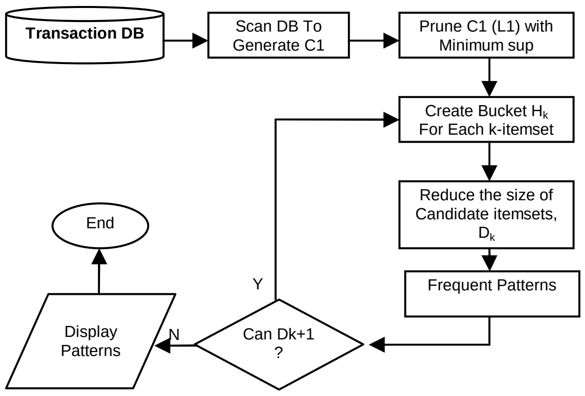
DHP is divided into 3 parts.

- Part 1 gets a set of large 1-itemsets and makes a hash table (H2) for 2-itemsets.
- Part 2 generates the set of candidate itemsets Ck based on the hash table generated in the previous pass, determines the set of large k-itemsts Lk, reduces the size of database for the next large itemsets and makes a hash table for candidate large (k+1)-itemsets.
- Part 3 is basically same as Part 2 except that it does not employ a hash table. DHP is particularly powerful to determine large itemsets in early stages, thus improving the performance bottleneck.

The size of Ck decreases significantly in later stages, thus rendering little justification its further filtering.

5. Proposed System

This system presents an algorithm for improvement of association algorithm with direct hashing and pruning algorithm. It scans the database for the first time to generate C_1 and L_1. From L_1, 2-patterns are generated. Then hash table H_2 is created using the hash algorithm. C_2 and L_2 are generated from the H_2. Reduced database D_3 is generated from L_2. Again hash table H_3 is created and C_3 and L_3 are generated. The same process is performed until no more reduced database can be created.



Transaction DB
Scan DB To Generate C1
Prune C1 (L1) with Minimum sup
Create Bucket H$_k$ For Each k-itemset
Reduce the size of Candidate itemsets, D$_k$
Frequent Patterns
Can Dk+1 ?
Y
N
Display Patterns
End

Figure 3: Process Flow of the System

The process flow of the proposed system is shown in the following Figure 3, where it reads the transaction database to generate C_1. Then C_1 is pruned to get L_1 by minimum support. Start from k = 2 (k value will be increased by 1 for every round), Hash buckets (H_k) are generated using Hash algorithm. Then by reducing the size of candidate itemsets, new and smaller size of transaction database (D_k) is generated. Then frequent patterns are extracted. This process will be repeated until it is able to build new transaction database D_{k+1}.

6. System Implementation

This paper presents an implementation of improving Association rule mining by DHP algorithm. Transaction data in this system is extracted from web-based cosmetic sales data. Sales records join with Sales Detail to create transaction data and association rule algorithm is applied to the category of sales item. Database system used in this system is shown in Figure 4, where 'Items' table stores all items in the database which is linked to 'ItemType' table and 'Brands' table. 'Sales' table stores header information of each sales and 'SalesDetail' table records detail item list of each sale. They are linked using 'SalesID', and they are transformed into transaction database later.

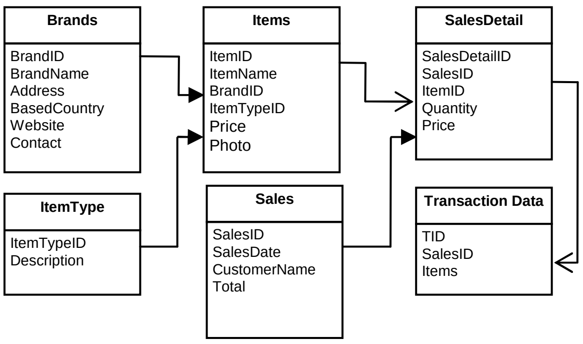


Figure 4: Database Design

6.1. Case Study

In this system, sales and salesdetail data are converted into transaction data. Example sales data and transaction data are shown in following tables.

Table 1: Data from Sales table

Sales ID	Sales Date	Customer	Total
1	05 / 05 / 2010	Mya Mya	40500
2	07 / 05 / 2010	Hla Hla	51000
3	10 / 05 / 2010	Ma Ma	62000
4	14 / 05 / 2010	Su Su	45000

Table 2: Data from Sales Detail table

SalesDetailID	SalesID	Item ID	Quantity	Price
1	1	LI01	1	11000
2	1	LO01	1	21000
3	1	EY01	1	8500
4	2	MA01	1	22500
5	2	LO01	1	21000
6	2	SM01	1	7500
7	3	LI01	1	11000
8	3	MA01	1	22500
9	3	LO01	1	21000
10	3	SM01	1	7500
11	4	MA01	1	22500
12	4	SM01	1	7500

Transaction database is generated by joining Sales table and sales detail table. SalesID field becomes the TID in transaction database.

Table 3: Transaction database

TID	SalesID	Items
1	1	LI01, LO01, EY01
2	2	MA01, LO01, SM01
3	3	LI01, MA01, LO01, SM01
4	4	MA01, SM01

6.3. Experimental Results

This system is tested with 1500 transactions for 202 items and 24 item types. In this paper, association is computed for item type. According to experimental results, DHP algorithm outperforms over the Apriori algorithm. Performance analysis is shown in Figure 5. In this system, minimum support is computed in decimal value of percentage, i.e., minimum support range is between 0 and 1.

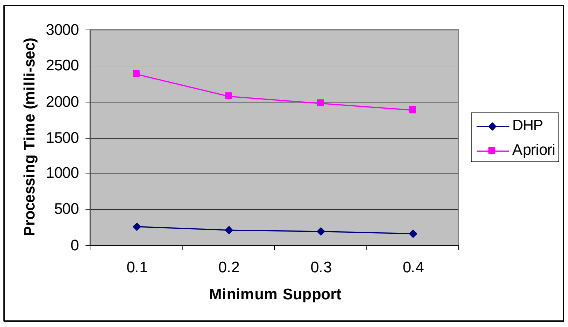


Figure 5: Performance Analysis for Processing Time (mili-seconds) of DHP and Apriori algorithm.

7. Conclusion

This paper presents the mining association rules among items in a large database of sales transactions. It will discuss an effective DHP algorithm for the initial candidate set generation. DHP is a hash-based algorithm and it is effective for the generation of candidate set of large 2-itemsets. In addition, the generation of smaller candidate sets enables us to effectively trim the transaction database at a much earlier stage of the iterations, thereby reducing the computational cost for later stages significantly.

8. References

[1] J. Han. Data Mining, Concepts and Techniques, Mining Association Rules in Large Databases.

[2] J. Han, J. Pei, Y. Yin, “Mining frequent patterns without candidate generation”. In: M. Dunham, J. Naughton, W. Chen eds. Proc. of 2000 ACM-SIGMOD Int’l Conf on Management of Data (SIGMOD’00). Dallas, TX, New York: ACM Press, 2000. pp. 1-12.

[3] L. Qin and Z. Shi, "Efficiently Mining Association Rules from Time Series", International Journal of Information Technology Vol.12 No.4 2006.

[4] M. Fan. and C.Li, “Mining frequent patterns in an FP-tree without conditional FP-tree generation”. Journal of computer research and development, 2003, 40(8). pp. 1216-1222.

[5] M. Houtsma and A. Swami, “Set-Oriented Mining of Association Rules” Technical Report RJ 9567, IBM Almaden Research Laboratory, San Jose, CA, October 1993

[6] N. Son. and E. Maria "A Further Study in the Data Partitioning Approach for Frequent Itemsets Mining", Proceeding at the 17th Australasian Database Conference (ADC 2006), Hobart, Australia.

[7] R. Agrawal and S. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. Proceedings of the 20th International Conference on Very Large Data Bases, September 1994.

[8] R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. Proceedings of ACM SIGMOD, pages207-216, May 1993

[9] R .T. Ng and J. Han. Efficient and Effective Clustering Methods for Spatial Data Mining .Proceedings on the 18th International Conference on Very Large Data Bases, pages 144-155, September 1994.