Hash-Close: A Hash-Based Algorithm for Mining Frequent Closed Itemsets

M. Shekofteh

***Abstract*—Frequent closed itemset tend to be a condensed representation method for frequent itemsets which have smaller set but carry the power similar to the frequent itemsets, and remove redundant rules. This study employs hashing technique to improve one of the most basic algorithms used in this area. The results of the implementation of the new algorithm shows that it is more efficient as far as execution time is concerned.**

***Index Terms*—Association rule mining, frequent closed Itemsets.**

I. Introduction

Association rule mining (ARM) is one of the most important data mining techniques. ARM aims at extraction, hidden relation, and interesting associations between the existing items in a transactional database. It is highly useful in market basket analysis for stores and business centers. For example, database mining of a department store customers reveal that those who buy milk would buy butter in 60% of occasions, and such principle is observed in 80% of transactions. In this example, the above-mentioned probability is called confidence percentage, and a percentage of transactions which cover this rule is termed support percentage. To find the rules, user should set a minimum amount for support and confidence which are called minimum support (min-sup) and minimum confidence (min-conf) respectively [1].

The main step in association rule mining is the mining frequent itemsets. Frequent itemsets mining often generates a very large number of frequent itemsets and rules. A popular condensed representation method is using to frequent closed item sets. Compared with frequent item sets, the frequent closed item sets is a much more limited set but with similar power. In addition, it decreases redundant rules and increases mining efficiency. Many algorithms have been presented for mining frequent closed item sets [2], and A-close proved to be a fundamental one [3].

This study attempts to improve A-close algorithm by hashing technique arriving at a new algorithm named hash-Close.

The rest of the paper is structured as bellow. Section 2 introduces the frequent closed item sets and relevant concepts. A-Close algorithms is described in section 3 and Section 4 elaborates on hashing technique and the new algorithm, i.e. Hash-Close. Section 5 reports on the results and section 6 provides the conclusion.

Manuscript received October 14, 2012; revised November 25, 2012.

M. Shekofteh is with the Department of computer engineering, Sarvestan Branch, Islamic Azad University, Sarvestan, Iran (e-mail: Shekofteh-m@iau.sarv.ac.ir).

II. Problem Development

Let D be a transactional database. Each transactional database includes a set of transactions. Each transaction t is represented by <TID, *x*> in which x is a set of items and TID is the unique identifier of transaction. Further, let us consider I = {*i*1, *i*2, …, *in*} as the complete set of distinct items in D. Each non- empty subset y of I is termed an item set, and if includes k items, it would be called k-item set. The number of transactions existing in D including item set y, is called the support of item set y, denoted as sup(*y*) and it is usually represented in percentage. Given a minimum support, min-sup, an item set *y* is frequent item set, if sup(*y*) $\geqslant$ min-sup.

Definition1- Closed Item set: An item set y is a closed item set if there is not any superset of y like *y*' that sup(*y*) = sup(y').

The precise definition of closed item set, however, is based on Relations (1) and (2). Let us consider T, y and then respectively $T \subseteq D, y \subseteq I$ are subset of all items and transactions appeared in D. Then functions f and g are defined based on Relations (1) and (2).

$$f(T) = \{i \in I \mid \forall t \in T, i \in t\} \quad (1)$$

$$g(y) = \{t \in D \mid \forall i \in y, i \in t\} \quad (2)$$

Considering these two functions, an item set y is a closed item set, if and only if Relation (3) is held.

$$H(Y) = F(G(Y)) = FOG(Y) = Y \quad (3)$$

The combinational function h = fog is called closure operator. If a closed item set is frequent, it is called frequent closed item set.

Definition 2- Generator: An item set p is a generator of a closed item set y if p is one of the item sets (there may be more than one) that determines y using Galois closure operator: $h(p) = y$.

III. A-Close Algorithm

A-Close algorithm is based on Apriori algorithm on frequent itemsets mining [4]. As this paper aims at improving A-Close by hashing techniques, first the A-Close algorithm is described briefly. A-Close is done through two general steps, i.e. (1) making frequent generators,

and (2) closure of frequent generators in which making frequent generators stands as a main step. To make generators a level-wise approach is followed as below: in each pass, first candidate generators of that pass are made. Next, their support is calculated and three pruning steps are done upon candidate generators so that non-useful generators, i.e. generators with the support lower than the minimum support, or generators whose one of their subsets is not available in the previous pass, or generators which have similar support with one of their subsets are pruned. The result of the pruning is the frequent generators of the involved pass. To keep precision, the candidate generators of pass *i* are shown as G_i , while the frequent generators of the pass *i* are represented as G_i'. The candidate generators of each pass are achieved by joining frequent generators of the previous pass, as two generators as long as *i* in G_i' whose i-1 initial items are similar are combined together and one candidate generator G_{i+1}- is achieved. It should be noted that G_1 stands for the single items in the database, and G_1' shows those items in G_1 whose support is greater than or equal to minimum support. The function of generating candidate generators by A-close is represented in Fig 1.

```
1.  insert into G_{i+1}
2.  select p.item_1 , p.item_2 , … , p.item_i , q.item_i
3.  from G_i p , G_i q
4.  where p.item_1 = q.item_1 , … , p.item_{i-1} = q.item_{i-1} , p.item_i<q.item_i;
5.  for all candidate generators c∈G_{i+1} do begin
6.     for all i-subset s of c do begin
7.        if(s∉G_i) then delete c from G_{i+1};
8.     end
9.  end
10. G_{i+1} ← Support-Count(G_{i+1});
```

Fig. 1. Generating candidate generators by A-Close

Once generators G_1 to G_n (n is maximum length of the generator), the closure of all these frequent generators should be calculated. The closure of all frequent generators results in the total of closed frequent itemsets. Table 1 displays a database. with minimum support 2, in G_1 there are itemsets such as {{A},{B},{C},{D},{E}}, while in G_2 there are {{AB},{AC},{AE},{BC},{BE},{CE}}, and frequent closed itemsets={{AC}, {BE}, {C}, {BCE}}.

TABLE I: A SAMPLE DATABASE

TID	ITEMS
1	ACD
2	BCE
3	ABCE
4	BE

IV. HASH-CLOSE ALGORITHM

A problem in A-Close algorithm is generating a large set of candidate generators especially in the second pass, as the number of G2 candidate generators equals ($|G_1'|*(|G_1'|-1)/2$) whereas if $|G_1'|$ is high, G2 will be much higher. As for a minimum supported, the number of occurrence of candidate generators with the length of 1 can be easily greater than or equal to the minimum support, therefore when G_1' is a large set, G2 becomes much larger. Needless to say, a large G2 implies pruning a great number of elements and thereby high cost. In other words, it is shown that the processing in initial passes especially in the second pass amounts over half of the whole algorithm cost. Accordingly, generating appropriate initial candidate generators especially for the second pass is a key to improve A-Close algorithm as well as other algorithms.

In this paper, a hashing technique is used for filtering inefficient candidate generators [5], [6], [7], as it effectively decreases the number of G_k. hashing technique arriving at a novel algorithm named Hash-Close.

In essence, Hash-Close uses the hashing technique to filter out unnecessary generator for next candidate generator generation. When the support of each generator in G_{k-1} is counted by scanning the database, new algorithm accumulates information about G_k in advance, by hashing all possible K-generator of each transaction to a hash table. Each bucket in the hash table contains an integer representing the number of generators that have already been hashed to this bucket so far. Based on the resulting hash table, a bit vector is constructed. The value of a bit is set to one, if the number in the corresponding entry of the hash table is greater than or equal to the minimum support.

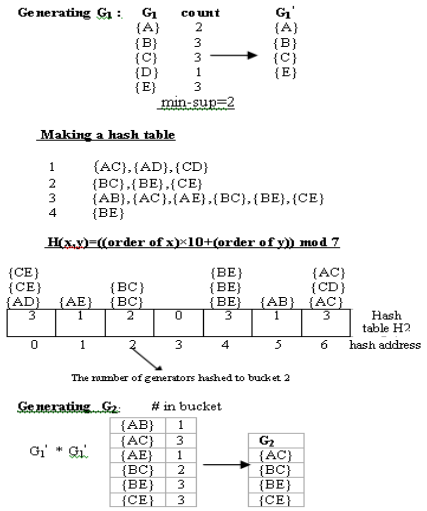

Fig. 2. Generating 2-candidate generators by hash-close

An example of generating candidate generators by Hash-Close from the database in Table I is represented in Fig 2. Based on this figure, for candidate generators with the length of 1, i.e. G_1={{A},{B},{C},{D},{E}}, all database transactions are analyzed to obtain support of these generators. For each transaction, once all the 1-subsets are counted, all the 2-subsets of that transaction are generated and hashed into a hash table H_2. As such, when a 2-subset is hashed to bucket i, 1 unit is added to the value of bucket i. In Fig. 1, hash table H_2 is shown for the assumed database. When the whole of database is analyzed, each bucket in hash table H_2 has the number 2-generators hashed to the bucket. Assuming the hash function H(x,y)=((order of x)×10+(order of y)) mod 7, and minimum support equal to 2, bit vector <1,0,1,0,1,0,1> is created. Using the bit vector, candidate set is filtered, and then instead of including all 2-generators from joining G_1' to G_1' (displaed as $G_1'* G_1'$) into G_2 ,and generating the set {{AB},{AC},{AE},{BC},{BE},{CE }}which we had in A-Close, G_2 is generated as {{AC},{BC},{BE},{CE}} which is a smaller set compared to G_2 in A-Close.

It should be noted that Hash-Close uses hash technique in initial passes where G_k is large, and in the final passes in which G_k is considerably small, the hash technique or hash

table is avoided. To set a bordering line between these passes, the algorithm follows this procedure: as long as the number of hashed bucket with greater than or equal to the minimum support is higher that a pre-set threshold named *Large*, the algorithm uses hash table, and then it takes the typical method of making candidate generators in A-Close.

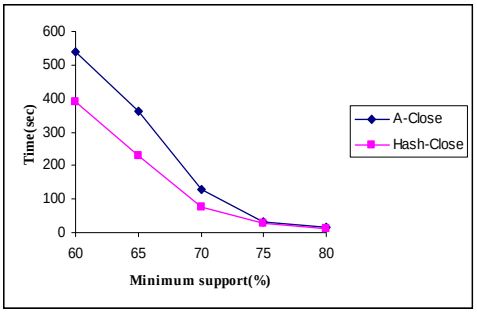

Fig. 3. Performance time comparison in pumsb dataset

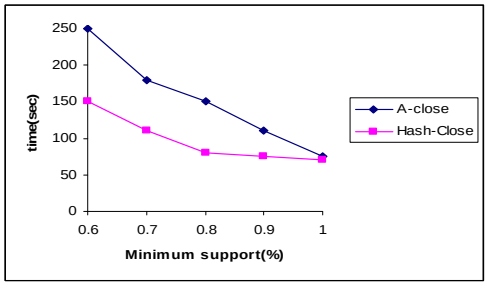

Fig. 4. Performance time comparison in T40I10D100k dataset

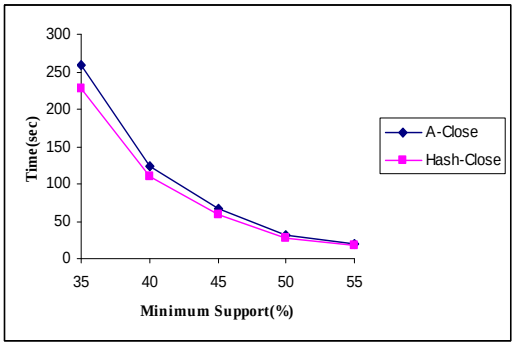

Fig. 5. Performance time comparison in connect dataset

TABLE II: DATASETS CHARACTERISTICS

Dataset	#transactions	#items
Pumsb	49046	7117
T40I10D100k	100000	1000
Connect	67557	130

V. PERFORMANCE EVALUATION

A. Implementation and Datasets; a General Investigation

In this section, A-close algorithm is compared with Hash-close algorithms. A-close and Hash-close algorithms have been also implemented with an eye to the same version. The results have been tested on three datasets whose characteristics are shown in Table II [8]. The dataset of the first and third rows belong to the real datasets, while the second dataset is synthetic. The first and third datasets are dense datasets, but the second one is sparse [9]

B. Performance Results

A number of tests have been made on various databases with varying minimum support to evaluate the efficiency of Hash-Close. Fig. 3, 4, and 5 compare A-Close with Hash-Close considering execution time in three database, i.e. Pumsb, T40I10D100k and connect.

As shown in these three figures, Hash-Close has more efficiency in the execution time over A-Close. The tests also have demonstrated that the larger the size of hash table H2, the smaller G2 and the lower the cost will be. Therefore, when the minimum support is small or the number of 2-generators is high, it is better to use high |H2| for Hash-Close.

VI. CONCLUSION

The frequent closed itemsets is an important condensed method for frequent itemsets which increases the effect of mining in the association rules. Recent years have witnessed a number of algorithms developed for mining frequent closed itemsets. A-Close is a basic algorithm in the area, however, it creates a very large set of candidate generators in initial passes especially in the second pass and wastes much time. In this study, using Hash technique, the size of candidate generators in the initial passes is decreased, and a new algorithm, i.e. Hash-Close in introduced. The tests demonstrate Hash-Close has less execution time compared with A-Close.

REFERENCES

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," presented at the. 20th int'1 conf. very large data bases, 1994.
[2] M. Shekofteh, "A survey of algorithms in FCIM," in *Proc of 2010 Int'l conf. Data storage and data engineering*, 2010, pp. 29-33.
[3] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, "Discovering frequent closed itemsets for association rules," in *Proc. of Int'l conf. Database Theory*, 1999, pp. 398-416.
[4] F. Bodon and L. S. Thieme, "The relation of closed itemset mining,complete pruning strategies and item ordering in apriori-based fim algorithms," Springer Berlin, 2005.
[5] C. C. Chang and C. Y. Lin, "perfect hashing schemes for mining association rules," *Oxford university press on behalf of the british computer society*, vol. 48, no. 2, 2005.
[6] J. S. Park, M. S. Chen, and P. S. Yu, "An effective hash based algorithm for mining association rules," *ACM SIGMOD international conference on management of Data*, vol. 24, pp. 175-186, 1995.
[7] J. C. R. Tseng, G. J. Hwang, and W. F. Tsai, "A mininmal perfect hashing scheme to mining association rules from frequently updated data," *Journal of the chines institude of engineers*, vol. 29, pp. 391-401, 2006.
[8] Frequent Itemset Mining Implementations Repository. [Online]. Available: http://fimi.cs.helsinki.fi
[9] P. Palmerini, S. Orland, and R. Perego, "Statistical Properties of Transactional Databases," presented at the ACM, 2004.

Maryam Shekofteh was born in Lar, Iran. She has got M.Sc. in computer engineering from Science and Research Branch-AHVAZ University in 2008 She is also an academic staff with Islamic Azad University, Sarvestan Branch, since 2008. her current research interests include Dataminig and she publishe some papers related to this field.