Journal of Theoretical and Applied Information Technology
20th May 2014. Vol. 63 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN ENHANCED SEMI-APRIORI ALGORITHM FOR MINING ASSOCIATION RULES

[1] SALLAM OSMAN FAGEERI
[2] ROHIZA AHMAD, [3] BAHARUM B. BAHARUDIN
[1, 2, 3] Department of Computer and Information Sciences
Universiti Teknologi PETRONAS Tronoh, Malaysia
[1] Sall_g01825@utp.edu.my, sallam_fageeri@hotmail.com

ABSTRACT

Mining association rules in large database is one of data mining and knowledge discovery research issue, although many algorithms have been designed to efficiently discover the frequent pattern and association rules, Apriori and its variations are still suffer the problem of iterative strategy to discover association rules, that's required large process. In Apriori and Apriori-like principle it's known that the algorithms cannot perform efficiently due to high and repeatedly database passes. In this paper we introduced an enhanced Semi- Apriori algorithm for mining frequent itemset as well as association rule, the algorithm work as follow, in the first step we use the first two steps used in Apriori algorithm in order to eliminate the repeatedly databases passes, in the third step we start discovering the rest of frequent pattern. In order to minimize the execution time as well as generating large number of candidate sets, we implement an enhanced Semi-Apriori algorithm using a binary based data structure, which is illustrated in details in enhanced Semi-Apriori technique section. Extensive experiments had been carried out, through comparing an enhanced Semi-Apriori with Apriori algorithm, the result show that our technique outperform Apriori in terms of execution time.

Keywords: *Data Mining, Frequent Items, Association Rules, Support, Confidence.*

1. INTRODUCTION

Data mining is an accumulation of techniques used to efficiently automate discovery of previously unknown, novel, valid, valuable and understandable patterns in large-scale databases [1, 2]. Data mining and knowledge discovery process aim to efficiently and probably discover the most relevant and interesting patterns and trends from the specified database [3, 4]. It becomes an important mechanism utilized for modern business to transfer data into intelligence prediction giving an informational advantage.

Association rule mining required two tasks, that's to find out all the frequent itemset, which their support is greater than or equal the specified minimum support threshold, then to find to generate the desired rule from the given frequent itemset, that their confident is greater than or equal to specified minimum confident, the first part is computationally expensive and intensive, while the second part can be generated in straightforward manner once all large items are obtained.

2. PROBLEM STATEMENTS

Association rule can be written in expression as an implication of the form $X \Rightarrow Y$, where X and Y are items of itemset I. where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \emptyset$. The expression means that if a transactions T contains the items in X, it also tends to contain the items in Y. An illustration of such a rule might be that 60% out of the total transactions that contain milk also contains sugar; 40% of all transactions contain the two items together". Here 60% will be known as confidence of the rule, and 40% will be known as support of the rule. Mining association rules from a set of items idea originates from the data analysis of market-basket, where will be the interest in mining association rules for describing customer's interest in buying product items process. The association rules problem normally can be classified into two sub problems. The first problem is to discover the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

itemsets that their occurrences go beyond a predefined minimum threshold [5]; if the itemsets pass this condition, then they will be known as large or frequent items. The second problem is to use those large frequent items to generate association rules with the constraints of minimal confidence. The two parameters used in finding the frequent itemset and association rules are support and confidence.

2.1 Measure of Association Rules

As stated earlier, to measure the association rule, there are formally two important criteria. Which are used for selecting the interesting rules: support and confidence [6]. The minimum support threshold and confident were normally determined and assigned by the user. Based on the set threshold, the itemsets with at least minimum support will be considered; while the others will be discarded, the justification of using these criteria is to discard those rules which are not interesting and useful. In data mining, association rule can be defined as strong association rule if the rule satisfies both minimum support and minimum confident threshold.

2.1.1 Support

In [7], the support has been defined as one of the measure parameters used to find the occurrence of an item or set of items among the total number of transactions. In the other words, support can calculate how many times an item or set of items appears in a set of transactions. An item or set of items can be known as frequent or large item if it has greater support. Using probability concept, we can formulate support as:

$$\text{Support} = P(A \cap B) = \frac{\text{number of transactions containing both A an B}}{\text{Total number of transactions}} \quad (1)$$

A and B represents the itemsets in a database D.

2.1.2 Confidence

Confidence is used for the purpose of measuring the strength of relation and association between the itemsets [8]. The confidence evaluation determines the probability of an item B occurs in the same transaction that also contains A. In the other words, the confidence is used to explore the conditional probability of the used items. The definition of confidence is

$$\text{Confidence} = P(A \mid B) = \frac{P(A \cap B)}{P(A)} = \frac{\text{number of transactions containing both A an B}}{\text{number of transactions containin A}} \quad (2)$$

In this paper, we propose a new algorithm named as enhanced Semi-Apriori technique. The algorithm requires only two database scan, then from the candidate itemset we generate the association rule in a new binary-based data structure.

The rest of this paper organized as follows, Section 3 describe the literature review of related work, section 4 describe the key idea about the proposed Enhanced Semi-Apriori technique, section 5 experiment and result discussion is reported, and conclusion is presented in section 6.

3. LITERATURE REVIEW

Apriori algorithm, introduced by Agrawal 1993[7], is a first frequent pattern and association rule mining algorithm, in order to control the exponential of candidate itemset growth, the algorithm use the support-based pruning concept, to investigate the concept behind the Apriori principle; Apriori algorithm [7], used to find all frequent itemset and its association rules, the algorithm use breadth-first (level-wise) search method which is known as iterative approach, the key feature of Apriori is to make multiple database passes. Apriori algorithm start generate candidate 1-itemset L1, the algorithm read the database transaction and start counting the occurrence of each database item individually, the candidate 1-itemset will be generated based on the minimum support threshold specified by the user. Then in the next step the algorithm determine the 2-itesmet L2 through joint L1*L1, Any transaction in the frequent 2-itemset that is greater than or equal to the minimum support threshold will be taken to the second level of large two itemsets, to generate the 3-itemsets L3, the algorithm combine the L2 itemset L2*L2, the algorithm use this concept till reach the prune step, the algorithm in this step removes all candidates

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

that their subsets are infrequent, thus generate the final candidate itemset that's pass the minimum support threshold. Various studies and extensions on the improvements of Apriori had been introduced, e.g., partition technique [9], hashing technique [10], sampling approach [11], incremental mining [12], dynamic itemset counting [13], Binary-Based technique [14], parallel and distributed mining [15-18], and integrating mining with relational database systems [19]. The tight upper bound of the number of candidate patterns derived the association rules, which can be generated in the level-wise mining approach [20]. This outcome will be effective and efficient at reducing the number of database passes. DIC, proposed by Brin et al. [13] can append candidate itemsets dynamically in different courses of scanning database. The above algorithms are all have an advantage over Apriori, but they still spend a great deal of time scanning database., DHP, proposed by Pork et al. [21] improves the efficiency of finding frequent 2-itemsets by adopting a Hash technology, and this method also improves the process of creating candidate itemsets.

Apriori and its several variation algorithms, uses the main strategy to discover the frequent itemsets, the strategy known as candidate generation and testing, Apriori algorithm [7] suffer from the cost of high I/O, another problem is also the time cost of discarding the candidate items which is classified as infrequent items in each level, the Apriori algorithm efficiency based on the maximal frequent items, because the database will access as many times as the database size.

TID	Combinations
100	A, C, D, AC, AD, CD, ACD
200	B, C, E , BC, BE, CE, BCE
300	A,B,C,E,AB, AC, AE, BC, BE, CE, ABCE
400	B, E, BE

4. ENHANCED SEMI-APRIORI TECHNIQUE

In this part, we present an enhanced Semi-Apriori algorithm for mining frequent items and association rules which we believe outperform Apriori algorithm. For the purpose of simplifying our discussion on the technique, in this paper we will highlight a database transaction that composed of items and products that are purchased together by customers who visited a supermarket. Tabe1 (1) shows sample content of database transactions that is used in this study (Note: TID stands for transaction id and items used for transaction itemsets). The transaction database contains only four transactions and five items; each transaction in the given table shows the purchase of one customer. Table (1) shows this concept.

Table (1) Sample content of database transactions

TID	I USED
100	A, C, D
200	B , C , E
300	A , B , C , E
400	B , E

Table (2) reveals all actual combinations that occur within the transactions given in table (1). For example, in transaction 100, only three items are available which are ACD. Thus, the combination of 100 will be A alone, C alone, D alone, AC, AD, CD, and ACD.

Tables (2) actual combinations

The Semi-Apriori algorithm for mining frequent items and generating association rules is divided into three stages. The first stage starts by finding the 1-itemsets L1 and pruning all items that have support less than the given minimum support threshold. This step is similar to the step used in Apriori [13] and FP-Growth algorithms [19]. The first stage demonstration and output can be shown as {A frequency 2 – B frequency 3 – C frequency 3 – E frequency 3}, the flowchart in figure (1), and the algorithm in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

figure (2) respectively explain this stage in details.

In the second stage, the algorithm figure (3) applies self-joint of L1 using L1⋈L1, also using the support measure. The items which are below the minimum support will be pruned. The second stage output can be shown as {AC, frequency 2 – BC, frequency 2 – BE, frequency 3 – CE, frequency 2}, flowchart in figure (1), and algorithm in figure (2) respectively explain this stage in details.

In the third stage, the remaining combination are generated, as illustrated in flowchart shown in figure (1), and the algorithm shown in figure (2). The algorithm start by reading all transactions from the database. For each transaction, the algorithm selects the items in the transaction that appears in 2-itemsets and add them to the local candidate set CS. The algorithm then proceeds to generates all the subsets of CS. For each generated subset, the algorithm calculates its binary map. After getting the map the algorithm looks for the map in the frequency table FT. If the map is already available in FT then the algorithm updates the frequency of this map increasing it by one. If the map doesn't exist then it's appended to FT and its frequency is set to 1. In order to avoid recalculation of the subsets, the algorithm firstly check whether the binary map of CS appears before or not. If the map appeared before then the algorithm go directly for updating the frequency of all binary maps that corresponds to the subsets of CS. Once this process is finished, the algorithm traces all the frequencies in the FT table to find the frequent itemsets. For each frequent itemsets that is having a value which is greater than threshold value of minSup the algorithm generates the reverse map of this frequency to get its constituent items back. These items represent the frequent mined items. The third stage can be shown in the flowchart figure (4) and in the algorithm figure (5). Once this steps are finished, the algorithm start extracting the association rule from the given frequent itemsets, this step can be done in straightforward manner Figure (6).

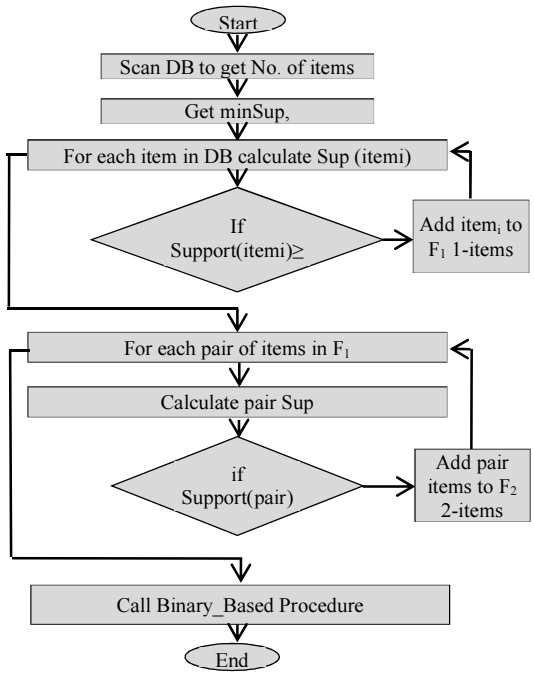

Figure (1) First And Second Frequent(1&2-Itemset) Algorithm

```
Input: Dataset D, minimum support threshold S
Output: frequent 1-items (FI1)
for all itemi in itemset
        calculate support(itemi) using eq. (1)
            if support(itemi)≥minSup
                additemi to L1
            end if
            end for
```

Figure (2) first frequent (1-itemset) algorithm

```
Input: first frequent 1-itemset
Output: frequent 2-items (FI2)
for i=1 to length(L1)
   for j=i+1 to length(L1)
      calculatesup (itemi, itemj) using eq. (2)
      if Conf (itemi, itemj)≥minSup
            additemi to L2
      end if
      calculateConf (itemi, itemj)
      if Conf (itemi, itemj)≥minSup
          additemj to L2
      end if
    end for
 end for
```

Figure (3) Second Frequent (2-Itemset) Algorithm

Journal of Theoretical and Applied Information Technology
20th May 2014. Vol. 63 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

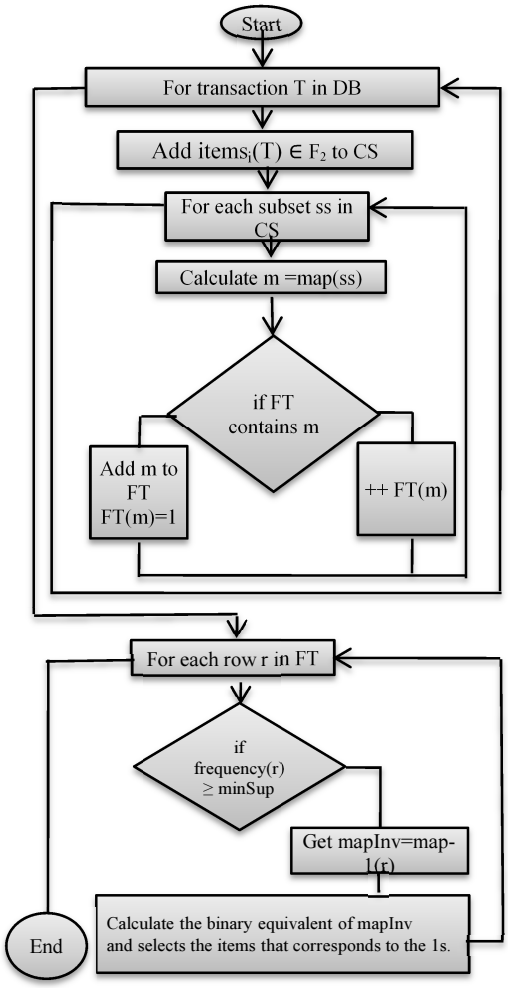

Figure (4) Generate The Candidate Itemsets Algorithm

Input: Dataset D, FI, minimum confident threshold C
Output: association rules (AR)

```
While not eof(DS)
    CS= {}
    Read T from DS
    For All items(T) do
        If itemi(T) ∈ L2
            Add itemi(T) to CS
        End if
    End for
    For each subset ss in CS
            Calculate m = map(ss)
            If frequency table FT contains m
                Increase the frequency of m by 1
            Else
                Add m to frequency table FT
                Set FT(m) = 1
        End for
  End while
  For each row r in FT
    if frequency(r) ≥minSup
            Get mapInv=map-1(r)
            Calculate the binary equivalent of mapInv and
selects the
            items that corresponds to the 1s.
        end if
end for
```

Figure (5) Generating all the frequent itemsets algorithm

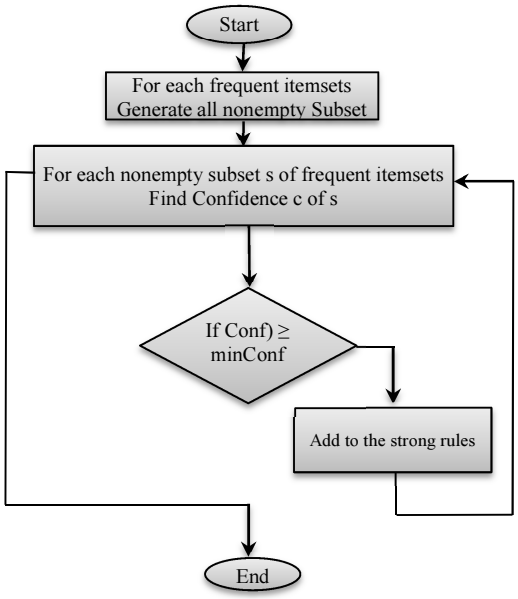

Figure (6) Generate The Association Rules

Table (3) Synthetic And Real Dataset's Properties And Features

Dataset	T40I10D100K	Mushroom
Type	Sparse	Dense
No of transactions	100,000	8,124
No of items	1000	119
AvgLength	40	23
Max items/ transaction	77	23
Size	14. 8 MB	557 KB
Support%	0.04 – 0.09	0.05 – 0.3
Frequent 1-itemset generated at Sup 0.05, and Conf 0.6	300	73
Frequent 2-itemset generated at Sup 0.05, and Conf 0.6	26	1329
Frequent 3-itemset generated at Sup 0.05, and Conf 0.6	0	10618

Journal of Theoretical and Applied Information Technology
20th May 2014. Vol. 63 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. EXPERIMENTS AND RESULTS

To compare performance of our algorithm with Apriori, experiments were conducted on Intel® corei5™ CPU, 2.4 GHz, and 02 GB of RAM computer. Graph 1 and graph 2 in figure (7) and figure (8) respectively shows the comparison of the execution time for mining association rule from synthetic dataset T40I10D100K provided by the QUEST generator from IBM's Almaden lab, and the real dataset, mushroom, that's publicly available in the FIMI dataset repository. The two datasets are having different transaction size, item size, and other behaviors. The minimum confidence was set at 60%. The graph x axis has support level 0.04 to 0.09 for the performance as compared to Apriori. For the both datasets, the execution time differs greatly between the original Apriori and semi-Apriori algorithms, when the minimum support is low. This is good since in data mining, scalability of the algorithms depends on ability to handle small minimum support.

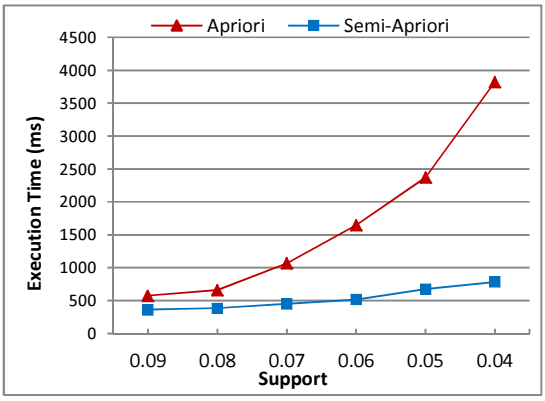

Figure (7) Execution Time Using T40I10D100K Dataset

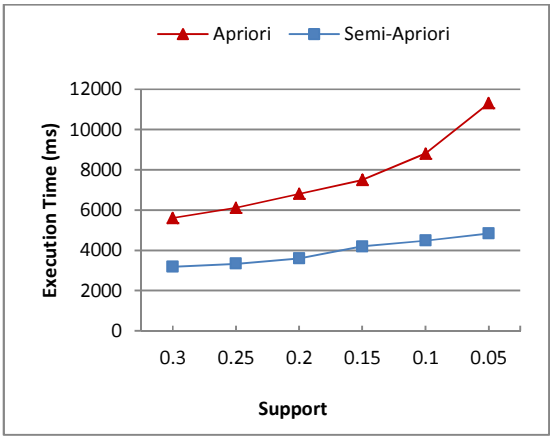

Figure (8) Execution Time Using Mushroom Dataset

Using the synthesis dataset T10I4D100K, in the support 0.01, the following output can be obtained

The top 5 Association rules with highest confidents

Rules	Support	Confident
2-itemsets		
819 ==> 70	41	0.75926
515 ==> 217	44	0.75862
515 ==> 283	44	0.75862
515 ==> 346	43	0.74138
819 ==> 765	40	0.74074
3-itemset		
33 515 ==> 346	41	0.97619
283 346 ==> 51	41	0.97619
33 283 ==> 515	40	0.97561
33 283 ==> 346	40	0.97561
33 217 ==> 515	40	0.97561

6. CONCLUSION

Mining frequent pattern and association rules in large database is one of data mining and knowledge discovery research issue. In this paper we introduced a new algorithm named as Semi-Apriori algorithm using binary-based data structure that's used to discover the frequent itemsets as well as association rules. The proposed technique avoids the cost of generating large number of candidate sets through using the binary-based data structure; hence, minimize the execution time. Extensive experiments have been carried out using the new technique, the algorithm was also compared to Apriori algorithm, the result reveal that our technique outperforms Apriori algorithm in terms of execution time to find the frequent itemsets as well as association rules.

REFRENCES

[1] D. Braha, Data mining for design and manufacturing: methods and applications: Kluwer academic publishers, 2001.

[2] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," AI magazine, vol. 17, p. 37, 1996.

[3] K. J. Cios, Data mining: a knowledge discovery approach: Springer, 2007.

[4] M. Linares Vásquez, D. F. Hernández Losada, and F. González Osorio, "Exploiting stock data: a survey of state of the art computational techniques aimed at producing beliefs regarding

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

investment portfolios," Ingeniería e Investigación, vol. 28, pp. 105-116, 2008.

[5] K. Rameshkumar, M. Sambath, and S. Ravi, "Relevant association rule mining from medical dataset using new irrelevant rule elimination technique," in Information Communication and Embedded Systems (ICICES), 2013 International Conference on, 2013, pp. 300-304.

[6] M. N. Dehkordi, "A Novel Association Rule Hiding Approach in OLAP Data Cubes," Indian Journal of Science and Technology, vol. 6, pp. 4063-4075, 2013.

[7] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in ACM SIGMOD Record, 1993, pp. 207-216.

[8] M. M. Mazid, A. Shawkat Ali, and K. S. Tickle, "Finding a unique association rule mining algorithm based on data characteristics," in Electrical and Computer Engineering, 2008. ICECE 2008. International Conference on, 2008, pp. 902-908.

[9] A. Savasere, E. R. Omiecinski, and S. B. Navathe, "An efficient algorithm for mining association rules in large databases," 1995.

[10] J. S. Park, M.-S. Chen, and P. S. Yu, "Using a hash-based method with transaction trimming for mining association rules," Knowledge and Data Engineering, IEEE Transactions on, vol. 9, pp. 813-825, 1997.

[11] H. Toivonen, "Sampling large databases for association rules," in VLDB, 1996, pp. 134-145.

[12] H. Cheng, X. Yan, and J. Han, "IncSpan: incremental mining of sequential patterns in large database," in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 527-532.

[13] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, "Dynamic itemset counting and implication rules for market basket data," in ACM SIGMOD Record, 1997, pp. 255-264.

[14] S. Fageeri, R. Ahmad, and B. Baharum, "A Log File Analysis Technique Using Binary-Based Approach," in Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013). vol. 285, T. Herawan, et al., Eds., ed: Springer Singapore, 2014, pp. 3-11.

[15] J. S. Park, M.-S. Chen, and P. S. Yu, "Efficient parallel data mining for association rules," in Proceedings of the fourth international conference on Information and knowledge management, 1995, pp. 31-36.

[16] R. Agrawal and J. C. Shafer, "Parallel mining of association rules," Knowledge and Data Engineering, IEEE Transactions on, vol. 8, pp. 962-969, 1996.

[17] D. W. Cheung, J. Han, V. T. Ng, A. W. Fu, and Y. Fu, "A fast distributed algorithm for mining association rules," in Parallel and Distributed Information Systems, 1996., Fourth International Conference on, 1996, pp. 31-42.

[18] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, "Parallel algorithms for discovery of association rules," Data mining and knowledge discovery, vol. 1, pp. 343-373, 1997.

[19] S. Sarawagi, S. Thomas, and R. Agrawal, "Integrating association rule mining with relational database systems: alternatives and implications," Data mining and knowledge discovery, vol. 4, pp. 89-125, 2000.

[20] F. Geerts, B. Goethals, and J. Van den Bussche, "A tight upper bound on the number of candidate patterns," in Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on, 2001, pp. 155-162.

[21] J. S. Park, M.-S. Chen, and P. S. Yu, An effective hash-based algorithm for mining association rules vol. 24: ACM, 1995.