Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2020, 7(7):1-7

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Frequent Itemset Mining Using Two Dimensional Transaction Reduction

Godspower Oraye, N. D. Nwiabu, Anireh V. I. E.

Department of Computer Science, Rivers State University, Port Harcourt, Nigeria
orasdaboss@yahoo.com, nwiabu.nuka@ust.edu.ng, anireh.ike@ust.edu.ng

Abstract Massive set of data having data sets that may be structured, semi-structured or unstructured make data mining difficult. Thus, it is problematic to discover and extract hidden patterns from the large databases. In this paper, Two Dimensional Transaction Reduction (TDTR) method has been applied as a technique for frequent itemset mining on large transactional databases. The approach applieduses row reduction and column reduction methods to mine the frequent itemsets. This method reduced the number of transactions. However, transactions that do not contain any frequent itemset were discarded and not considered for further database scan. The method counted the number of items in each transaction and removed transactions having count less than minimum support. System was implemented in python programming language. Results showed that scanning time and execution time of finding frequent items in the large transaction dataset was minimized.

Keywords Frequent Itemset, Mining, Two Dimensional Transaction Reduction

Introduction

Data Mining is the extraction of hidden information from large databases. It is also known as Knowledge Discovery in Databases (KDD) which in recent times is applied using new techniques that help industries to focus on the essential information from the data bases [1].

Data mining tools allow industries to generate knowledge-driven decisions by applying techniques that can be used to generate new software using existing resources to produce an efficient product for the society. Liu *et al* [2] proposed p-value (via Pearson's 2 test). This does a statistical selection of association rules that are significant. By this test, a correlation between a rule's antecedent and consequent (left and right terms) is said to exist if the p-value is low. In their work, they derived an exploration framework where rules are grouped by consequent, then traversed by progressively adding items to the antecedent. The derived framework provides hints to help the user to guess how each additional item would make a difference. Implementing this on large data sets raises resource issues because the whole dataset should be held in memory during the exploration by the system and each step may take a lot of time to meet the requirements of applications that are interactive.

Data mining tools can analyze the massive databases. With the current progress in science and technology, user habits, and businesses, a massive quantity of data is being produced and stored. Therefore, effective analysis of these large volumes of data generated has become more significant for both businesses and academics. Frequent Itemset Mining is an important data analysis and mining method with the task of retrieving interesting information from large databases on the basis of the frequency of occurrence of items in the data; an event or a set of events, with respect to a user-specified minimum frequency threshold [3]. The amount of data kept in computer databases is growing at an extraordinary rate. At the same time, the users are expecting more sophisticated information from them. Many techniques have been designed to extract databases depending on the frequency of events, such as FP-Growth and Apriori algorithm. Apriori algorithm is one such scheme, where frequency accumulation is done by reading the dataset iteratively for each size of the candidate itemsets [4]. Unfortunately, the extremely large memory required for managing the large number of candidate itemsets makes

the Apriori-based schemes incompetent to be applied on single machines. New techniques were developed to keep both output and run time under control. This is achieved by incrementing the minimum threshold frequency which in turn reduces the number of candidate and frequent itemsets. But according to research on recommendation systems, itemsets with lower frequencies are more desirable [5]. Although these methods work well in practice for archetypal datasets, they are not suitable for extremely large data. Therefore, applying Frequent Itemset Mining to large databases is difficult. Because very large databases are not generally stored in main memory. This research aims to find frequent itemset mining on large transactional databases using two-dimensional transaction reduction method.

2. Related Literature

Data mining is an area of research that deals with automatic discovery of patterns that are very useful and easy to understand. These patterns are usually from large data sets [6]. The four common groups of data mining tasks include the following: classification, cluster analysis, anomaly detection, and pattern mining. Classification deals with the assigning of observations from the data set to target categories, or classes. The aim is to accurately predict the target class for each observation. In cluster analysis, data is analyzed to extract groups of closely related observations in such a way that observations that belong to the same cluster are more similar to each other than observations that belong to other clusters. Anomaly detection is a data mining task which has to do with identifying observations whose characteristics are significantly different from the rest of the data.

In frequent pattern mining, a database in which an association rule is to be found is viewed as a set of tuples or records, where each tuple contains a set of items and each item represents an item purchased at a moment.

Ahmed and Karypis [7] proposed algorithms for mining the evolution of conserved relational states in dynamic networks. That is the extraction of maximal non-redundant evolution paths among Induced Relational States (IRS). An induced relational set is a time-conserved set of relations among a fixed set of vertices. More formally, it is a tuple Si = (Vi; si :ei) where Vi is the set of vertices of the induced subgraph that persists from snapshot Gsi to Gei; where vertices are the same and edges are conserved on each timestamp. Then an evolving induced relational state identifies a sequence of time-persistent relational patterns. For instance, in a social network, a pattern could describe the evolution of a group of friends, however, instead of the previous methods, Evolving Induced Relational States (EIRSs) identify stable groups of friends over a set of time and detect evolutions between these stable states.

Parthasarathy *et al* [8] proposed a progressively sampling method for association rules. It identifies the classes that represent the similarity values from the original database. This method reduces the overhead of progressive sampling.

Robardet [9] proposed constraint-based pattern mining in dynamic graphs. "To extract evolving patterns over a time series of graphs; dense subgraphs are extracted separately on each timestamp and associated to study their evolution. For instance, in a dynamic social network, patterns could describe the evolution of groups of friends as the apparition of new persons in the groups. Subgraphs are considered as interesting if they are pseudo-clique; they have a high density of edges".

Lakshmanan *et al* [10] did a work on the integration of constraints on itemsets in mining, considering conjunctions of conditions on itemsets. They presented "CAP" algorithm, this is a technique that allows itemsets that do not satisfy the query to be generated and tested. The algorithms implement a rule-query by separately mining for possible heads and for possible bodies. A further study was carried out by Pei *et al* [11] and employed within the FP-growth algorithm.

Jaroszewicz and Simovici [12] proposed a technique for pruning itemsets and it is based on background knowledge represented by a Bayesian network. The interestingness of an itemset is defined as the absolute difference between its supports estimated from data and from the Bayesian network.

3. Two Dimensional Transaction Reduction (TDTR)

This method deletes infrequent items and reduces the number of transactions for further database scans; It reduces the scanning time and execution time, in finding frequent item sets. The main idea of frequent pattern mining algorithm is that any subset of a frequent item set is a frequent item set and any superset of an infrequent

item set is an infrequent item set. For example, transaction database consists of table id and items as shown in Table 1.

Table 1: Transaction Table

TRANSACTION	ITEM SET
T_1	GSN
T_2	B
T_3	F
T_4	GN
T_5	AG
T_6	AEG

"An item set is said to be frequent if its support is greater than a given threshold σ, which is called minimum support" [13].

Minimum support is 50%. Support can be obtained as follows:

$$Support = \frac{minimum \quad spport}{100} * number\ of\ transaction \tag{1}$$

$$Support = \frac{50}{100} * 4 = 2$$

Thus, minimum support count = 2

Count is used to find the number of items in each transaction record as shown in Table 2.

Table 2: Candidate D_1

Transaction	Itemset	Count 1
T_1	GSN	3
T_2	B	1
T_3	F	1
T_4	GN	2
T_5	AG	2
T_6	AEG	3

T_1 consists of three items, T_2 has one item, T_3 has one item, T_4 has two items, T_5 has two items and T_6 has three items.

Remove the transaction having count1 ≤ minsup from the Database D and call it as D1 which is a reduced database as shown in Table 3.

Table 3: Transaction Reduction after Removing the Transactions

Transaction	Itemset	Count 2
T_1	GSN	3
T_4	GN	2
T_5	AG	2
T_6	AEG	3

T_2 and T_3 are removed from database since the count is less than minimum support. The items with minimum support are represented in Table 4.

Table 4: D_1 with the Count of Items

Transaction	A	E	G	N	S
T_1	0	0	1	1	1
T_4	0	0	1	1	0
T_5	1	0	1	0	0
T_6	1	1	1	0	0
Count 1	2	1	4	2	1

From D1, count the number of transactions for each item as count2.If the count of each item count2 is less than minimum support, then remove that item from the Database (D1). The transaction is reduced to items that meet the minimum support count as shown in Table 5.

Table 5: Reduced Transaction (D_2)

TRANSACTION	A	G	N
T_1	0	1	1
T_4	0	1	1
T_5	1	1	0
T_6	1	1	0
COUNT 2	2	4	2

Item E and S did not meet the minimum requirement of support and are removed from transaction records. Thus, Item {A}, {G} and {N} satisfied the minimum support, From the reduced database D1 the frequent itemsets are extracted using association rule.

3.1. Association Rule

An association rule can be represented as follows X->Y, where X and Y belong to candidate set D, and D= {d1, d2....dn} is a set of n items belonging to a retail store. In the rule X->Y, the itemset on left hand side (X) of the rule is called antecedent of the rule and itemset on righthand side (Y) of the rule is called consequent. The task in association rule is to find the relationship between the items in the rule. For example, Perfume →Gucci Guilty implies that customer bought perfume called smart.

Table 6 consists of frequent repeating item based on the minimum support count obtained in equation 1. The items that do not meet the minimum support count are pruned.

Table 6: Frequently Repeat Item using Minimum Support (1)

Items	Support
{A}	2
{G}	4
{N}	2

Table 6 shows single items and their support value. Table 7 consists of candidate with the minimum support count, item set in group of two. Item in the candidate B is grouped only with the items that come after it.

Table 7: Candidate B

Items	Support
{A, G}	2
{G,N}	2
{A,N}	0

Here, items are paired to evaluate the support value. The pairs {A, G} and {G, N} satisfied minimum support count of 2, so they are frequent. But {A, N} did not meet the minimum support, it implies not frequent. Items with the minimum support count left are found in Table 8.

Table 8: Frequently Repeat Item using Minimum Support (2)

Items	Support
{A, G}	2
{G,N}	2

The key transaction is {A, G} and {G, N}. These candidates satisfy the minimum support of 2.

Minimum confidence is 50%. Confidence is obtained as follows:

$$Confidence = \frac{Support}{Occurrence\ \ of\ item} \quad (2)$$

Table 9: Association

Association Rules	Support	Confidence	Confidence %
A → G	2	1	100%
G → A	2	0.5	50%
G → N	2	0.5	50%
N → G	2	1	100%

$$A \rightarrow G = \frac{2}{2} = 1$$

$$G \rightarrow A = \frac{2}{4} = 0.5$$

$G \rightarrow N = \frac{2}{4} = 0.5$

$N \rightarrow G = \frac{2}{2} = 1$

Rules:

A → G

G → A

G → N

N → A

4. Results and Discussion

The performance evaluation for Two Dimensional Transaction Reduction (TDTR) algorithm has been obtained via efficiency. Efficiency determines the time taken to generate the association rules for support measure. The TDTR has used 50% support measure as depicted in Figure 1.

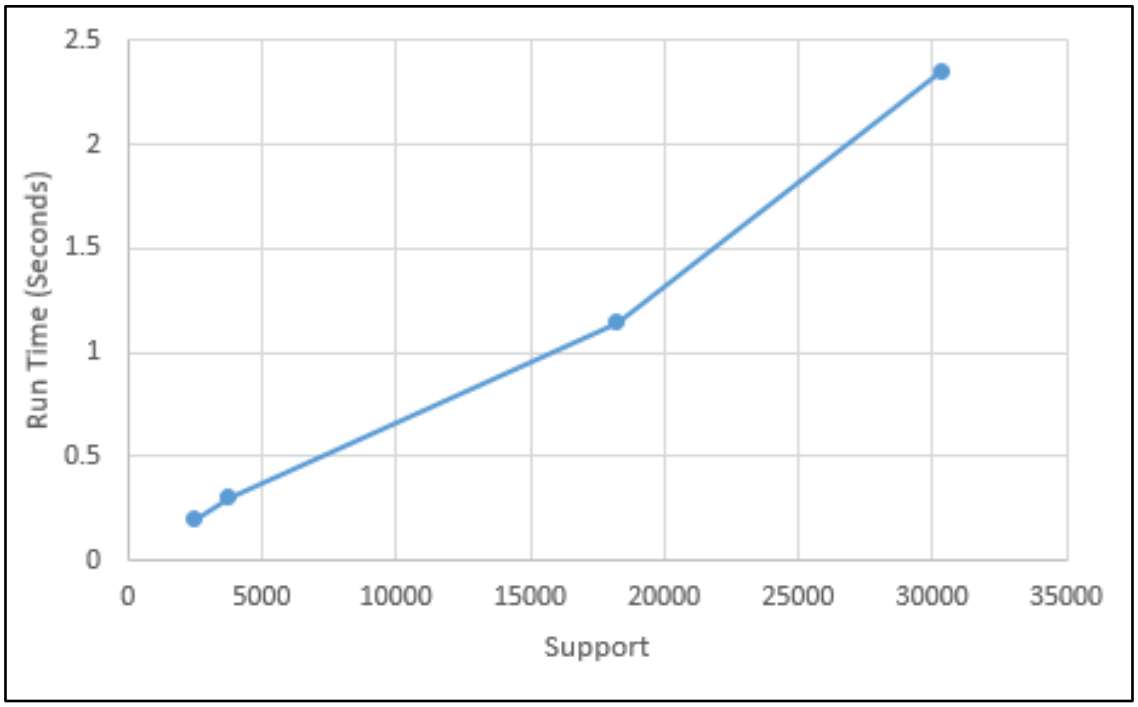

Figure 1: Performance Measure

The system has applied Two Dimensional Transaction Reduction (TDTR) method by applying row and column reduction to obtain frequent and most frequent purchases made from large retail transactions. By comparison, Apriori algorithm was slower than TDTR as shown in Table 10, we observed that Apriori takes more amount of time to process data in mining the frequent itemsets. Both algorithms were implemented and different sales transaction data sets were used to view the behavior of the algorithms on their run time.

Table 10: Comparison

Number of Transactions	Support	Models	
		TDTR (Runtime)	Apriori (Runtime)
5000	2500	0.200	0.334
7550	3775	0.305	0.492
36440	18220	1.146	2.362
60700	30350	2.354	3.806

The comparison run time of TDTR and Apriori is depicted in Figure 2. The TDTR takes less execution time when comparing with the Apriori algorithm. It takes less time to generate association rules for increasing confidence values.

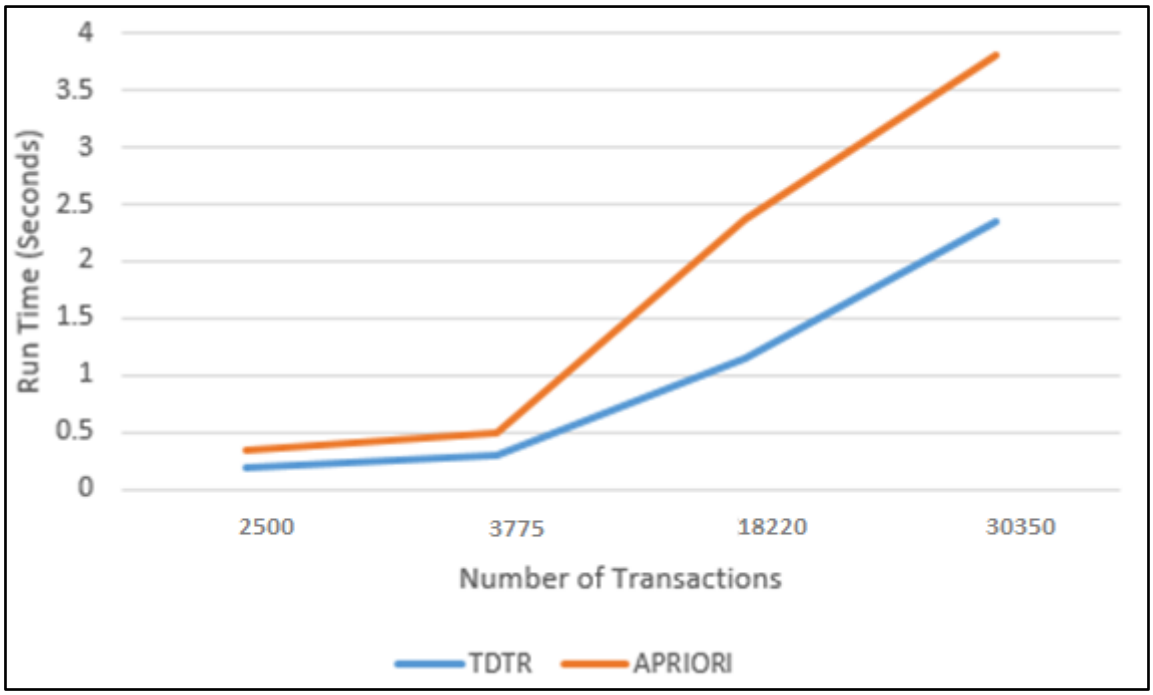

Figure 2: Comparison of Execution Time in seconds by TDTR and Apriori Algorithms with Support Count of 50%.

5. Conclusion

Two Dimensional Transaction Reduction (TDTR) has been used by applying row reduction and column reduction method. This method reduces the number of transactions by deleting transactions that do not contain any frequent itemset. Such transactions will not be required for further database scans. The method finds the count of number of items in each transaction and removes transactions having count less than minimum support. However, from the database the frequent item sets are extracted using association rules. Transaction reduction increases the performance of finding the frequent itemsets.

References

[1]. Hanirex, D. K. (2017). Association pattern mining for efficient generation of frequent item sets in dengue virus type1 for drug discovery using an integration of transaction reduction and random sampling algorithm. *Bharath University*.

[2]. Liu, G., Feng, M., Wang, Y., Wong, L., See-Kiong N., Mah, T. L. and Deoon-Lee, E. J. (2011). Towards exploratory hypothesis testing and analysis. In *ICDE*, pp. 745–756.

[3]. Agrawal, R. and Srikant, R. (1994). Fast Algorithms for Mining Association Rules. *Proceedings of the 20th VLDB Conference*, pp. 487−499.

[4]. Ibrahim, A., Jin, H., Yassin, A. A. and Zou, D. (2012).Towards privacy preserving mining over distributed cloud databases. In *Proceedings of the 2nd International Conference on Cloud and Green Computing (CGC) IEEE*, pp. 130-136.

[5]. Hegland, M. (2005). The apriori algorithm – a Tutorial, *CMA*, Australian National University, *WSPC/Lecture Notes Series*, pp. 22-27.

[6]. Han, J., Cheng, H., Xin, D., & Yan, X. (2007). Frequent pattern mining: current status and future directions. *Data mining and knowledge discovery*, 15(1), 55-86.

[7]. Ahmed, R. and Karypis, G. (2011). Algorithms for Mining the Evolution of Conserved Relational States in Dynamic Networks. In *ICDM*, pp. 1–10.

[8]. Parthasarathy, S., Zaki, M. J. and Ogihara, M. (2001). "Parallel data mining for association rules on shared-memory systems. Knowledge and Information systems", *An International Journal*,3(1),pp.1-29.
[9]. Robardet, C. (2009). Constraint-Based Pattern Mining in Dynamic Graphs. In *ICDM*, pp. 950–955.
[10]. Lakshmanan, L.V.S., Ng, R.T., Han, J. and Pang, A. (1999).Optimization of constrained frequent set queries with 2-variable constraints. *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pp. 157–168.
[11]. Pei, J., Han, J. and Lakshmanan, L.V.S. (2001). Mining frequent itemsets with convertible constraints. In *Proceedings of the 17th International Conference on Data Engineering, IEEE Computer Society*, pp. 433–442.
[12]. Jaroszewicz, S. and Simovici, D. A. (2004). Interestingness of frequent itemsets using Bayesian networks as background knowledge. *Proceedings 10th ACM SIGKDD International Conference on Knowledge discovery and data mining*, pp. 178–186.
[13]. Moens, S., Aksehirli, E. and Goethals, B. (2018). Frequent Itemset Mining for Big Data, *Universiteit Antwerpen*, pp.2.