Volume 3, Issue 9, September 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Efficient Data Mining Algorithms for Mining Frequent/Closed/Maximal Itemsets

Peddaboina Lingaraju*, K.Yellaswamy, B.Sivaiah

Department of CSE & JNTUH

India

Abstract- Data mining algorithms have been around for discovering knowledge from large real world data sets. Especially they operate on historical data in OLAP (Online Analytical Processing) applications. Frequent itemset mining is used in many applications such as query expansion, inductive databases, and association rule mining. When an itemset is repeated in specified number of times in given dataset, it is known as frequent itemset. Frequent itemset which does not appear in other frequent itemset is known as maximal itemset. If the frequent itemset is not included in other itemset then it is known as closed itemset. These itemsets have their utility in data mining applications. Recently Uno et al. proposed algorithms to discover frequent itemsets, closed itemsets and maximal itemsets. In this paper we implement these algorithms. We also built a prototype application to demonstrate the proof of concept. The experimental results revealed that the application is useful and can be used in real world applications.

Index Terms –Data mining, frequent itemset, closed itemset, maximal itemset

I. INTRODUCTION

Data mining is the domain which provides many algorithms to discover trends or patterns from the data which is meant for analysis and making well informed decisions. Expert decision making systems are made using efficient data mining algorithms. There are many algorithms that exist in data mining. They include C4.5, K-means, support vector machines, Apriori, the EM algorithm, PageRank algorithm, AdaBoost algorithm, kNN algorithm, Naïve Bayes algorithm, and CART (Classification and Regression Trees). These are the top 10 data mining algorithms that have been around for various data mining applications. For instance Apriori is use to discover frequent patterns while C4.5 is used for clustering objects. K-Means is also a famous algorithm for clustering objects.

Frequent itemset mining is one of the data mining approach in which itemset that is frequently repeated in given dataset is discovered based on the given threshold and confidence. Its applications include inductive databases, association rule mining and query expansion. Fast implementations[1], [2] offrequent itemset mining will help organizations to mine the bulk of business data to take policy decisions. The data mining is thus able to discover business intelligence from underlying datasets [2]. The business intelligence can lead to well informed expert decision making which in turn can lead to higher profits to an organizations. For this reason, of late, organizations are using data mining techniques in order to extract actionable knowledge which is not possible to do manually due to the bulk of transactions present in their databases.

Recently Uno et al. [3] presented many algorithms for discovering itemsets. The algorithms developed by them can extract frequent itemsets, closed itemsets, and maximal itemsets [4], [5] and [6]. These algorithms are efficient in discovering actionable knowledge. In this paper we implement all these algorithms using Java platform. We built a prototype application that demonstrates the efficiency of the algorithms [7]. The empirical results revealed that the proposed application is very useful and the algorithms can be used in real time applications for best possible business intelligence that lead to accurate decision making. The remainder of this paper is structured as follows. Section II provides preliminariespertaining to itemset mining. Section IIIprovides details of proposed algorithms. Section IV presents prototype information. Section V presents experimental results while section VI concludes the paper.

II. PRELIMINARIES

Set of items is known as itemset. It can be represented as I = {1, 2, 3, n}. Considering a transaction dataset T has many transactions which are represented as T= {t1, t2, t3, …, tn). For any given itemset P, if a transaction contains P is known as an occurrence of P. T (P) represents all occurrences of P in T. And the representation such as |T (P)| is known as frequency of P and denoted as frq (P). The P is frequent [8] only if it exceeds the given support and confidence. If the frequent itemset P is not included in any other frequent itemset [9], then the P is known as maximal itemset. If an itemset P is not included in any other itemset included in the same transactions as P, it is known as closed itemset.

III. ALGORITHMS

Algorithms presented by Uno et al. [10] are presented here. They are meant for extracting frequent itemsets, closed itemsets and maximal itemsets. For frequent itemset mining the algorithm presented in fig. 1 is used.

ALGORITHM BackTracking (P:current solution)
1. Output P
2. For each $e \in \mathcal{I}, e > tail(P)$ do
3. If $P \cup \{e\}$ is frequent then
call BackTracking ($P \cup \{e\}$)

Fig. 1 – Backtracking algorithm for frequent itemsets

As can be seen in fig. 1, the backtracking algorithm has a function by name BackTracking() which is recursive in nature. It calls itself until al frequent itemsets are resulted. The input is the dataset given to this while the output is a set of frequent itemsets denoted by P. For extracting closed itemsets the algorithm presented in fig. 2 is used.

```
Algorithm LCM()
1. X := I(T(∅))   /* The root ⊥ */
2. For i := 1 to |E|
3.   If X[i] satisfies (cond2) and (cond3) then
       Call LCM_Iter( X[i], T(X[i]), i ) or
       Call LCMd_Iter2( X[i], T(X[i]), i, DT )
         based on the decision criteria
4. End for
LCM_Iter( X, T(X), i(X) )   /* occurrence deliver */
1. output X
2. For each T ∈ T(X)
     For each j ∈ T, j > i(X), insert t to J[j]
4. For each j, J[j] ≠ ∅ in the decreasing order
5.   If |J[j]| ≥ α and (cond2) holds then
       LCM_Iter( T(J[j]), J[j], j )
6.   Delete J[j]
7. End for
LCM_Iter2( X, T(X), i(X), DT )   /* diffset */
1. output X
2. For each i, X[i] is frequent
3.   If X[i] satisfies (cond2) then
4.     For each j, X[i] ∪ {j} is frequent,
         DT'[j] := DT[j] \ DT[i]
5.     LCM_Iter2( T(J[j]), J[j], j, DT' )
6.   End if
7. End for
```

Fig. 2 – Linear time Closed itemset Mining Algorithm [11]

As can be seen in fig. 2, the algorithm is meant for extracting closed itemsets. The algorithm takes dataset as input and generates closed itemsets as output. For maximal itemset generation, the algorithm presented in fig. 3 is used.

ALGORITHM LCMmax (P:itemset, H:items to be added)
1. H' := the set of items e in H s.t. $P \cup \{e\}$ is frequent
2. If $H' = \emptyset$ then
3. If $P \cup \{e\}$ is infrequent for any e then
output P ; return
4. End if
5. End if
6. Choose an item $e^* \in H'$; $H' := H' \setminus \{e^*\}$
7. LCMmax ($P \cup \{e\}$, H')
8. P' := frequent itemset of the maximum size found in the recursive call in 7
9. For each item $e \in H \setminus P'$ do
10. $H' := H' \setminus \{e\}$
11. LCMmax ($P \cup \{e\}$, H')
12. End for

Fig. 3 –Algorithm for Discovering Maximal Itemsets

As can be seen in fig. 3, the algorithm is used to generate maximal itemsets. It takes an itemset P and some items to be added as input and generate maximal itemsets.

IV. PROTOTYPE IMPLEMENTATION

A prototype application is implemented to demonstrate the efficiency of the algorithms proposed. The application is built using Java platform. We built application with graphical user interface. The environment used to build the application

includes a PC with 4GB RAM, Core 2 dual processor running Windows 7 operating system. The application provides user interface to perform mining to extract frequent itemsets, maximal itemsets, and closed itemsets. Fig. 4 shows the results of algorithm for extracting frequent itemsets.

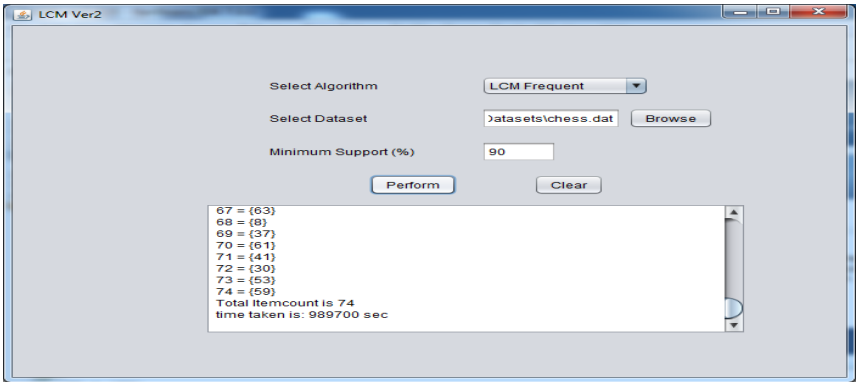

Fig. 4 – Result of algorithm for extracting frequent itemsets

As can be seen in fig. 4, user can choose a dataset and also algorithm. The result of algorithm for extracting frequent itemsets is presented in a text area. The result is based on the minimum support given by end user or a domain expert. Figure 5 shows the results of algorithm for closed itemsets.

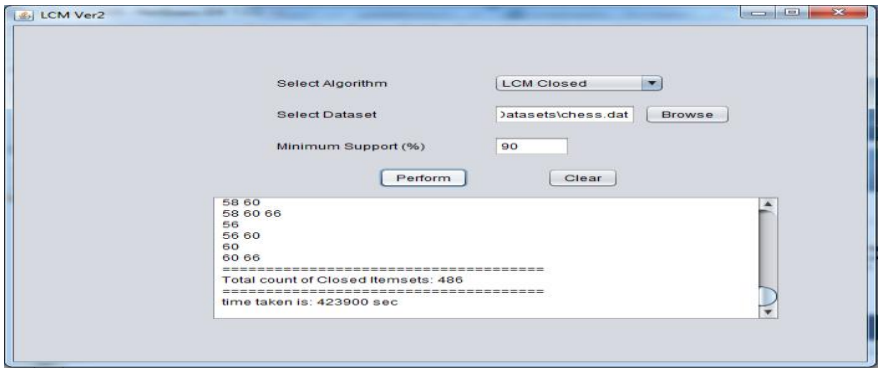

Fig. 5 – Result of algorithm for closed itemsets

As can be seen in fig. 5, user can choose a dataset and also algorithm. The result of algorithm for extracting closed itemsets is presented in a text area. The result is based on the minimum support given by end user or a domain expert. Figure 6 shows the results of algorithm for maximal itemsets. The time taken for discovering the number of closed itemsets is also presented.

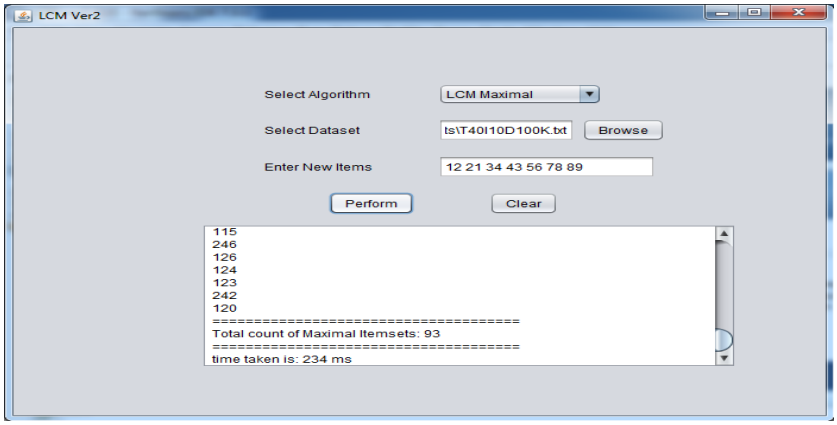

Fig. 6 – Result of algorithm for maximal itemsets

As can be seen in fig. 6, user can choose a dataset and also algorithm. The result of algorithm for extracting maximal itemsets is presented in a text area. The result is based on the new items added by end user or a domain expert. The time taken for discovering the number of closed itemsets is also presented.

V. EXPERIMENTAL RESULTS

Experiments are made using various datasets and with all algorithms proposed. We also compare the results of the algorithms implemented in this paper with previous algorithms. All the experiments are made using a minimum support value given by domain expert. A series of graphs were plotted to visualize the results of experiments.

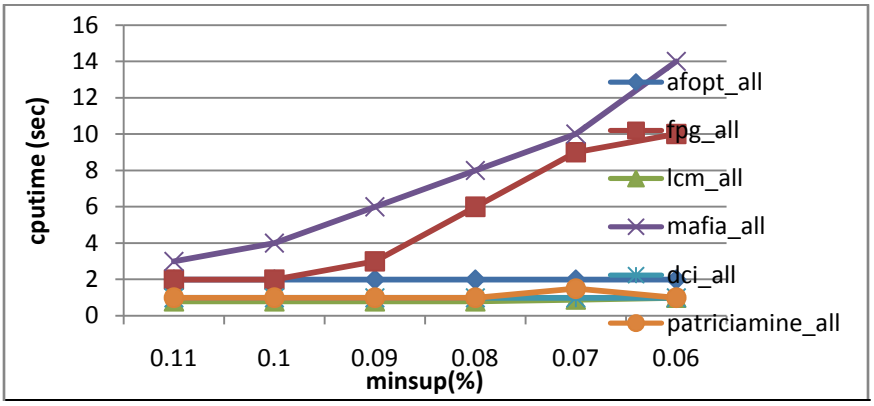

Fig 7. BMS-WebView-2-all

As shown in fig 7. Represents the horizontal axis represents minsup while vertical axis represents cputime.

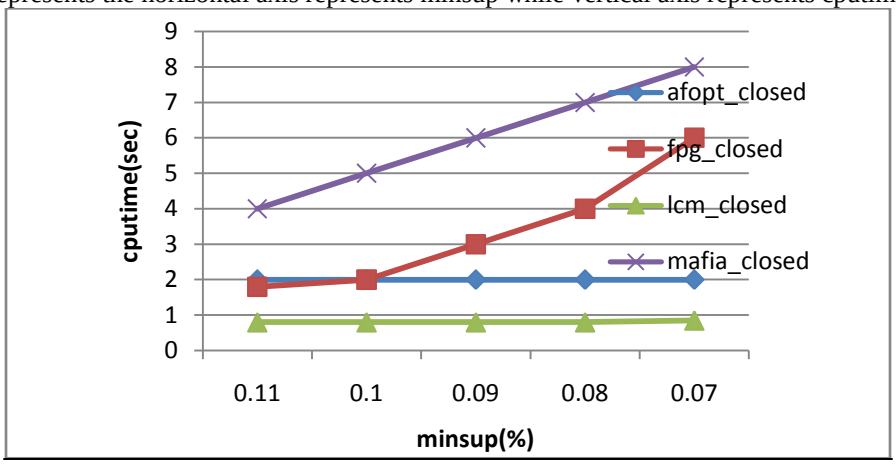

Fig 8 BMS-WebView-2-Closed

As shown in fig. 8. Represents the horizontal axis represents minsup while vertical axis represents cputime.

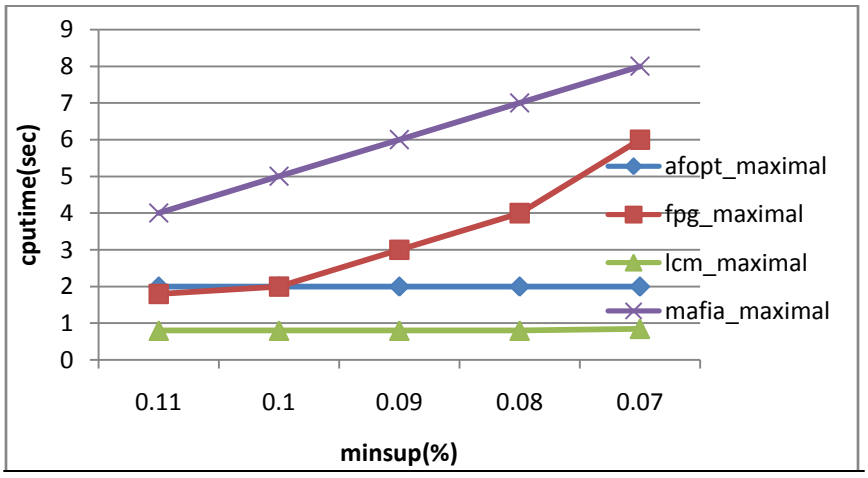

Fig 9 BMS-WebView-2-Maximal

As shown in fig. 9. Represents the horizontal axis represents minsup while vertical axis represents cputime.

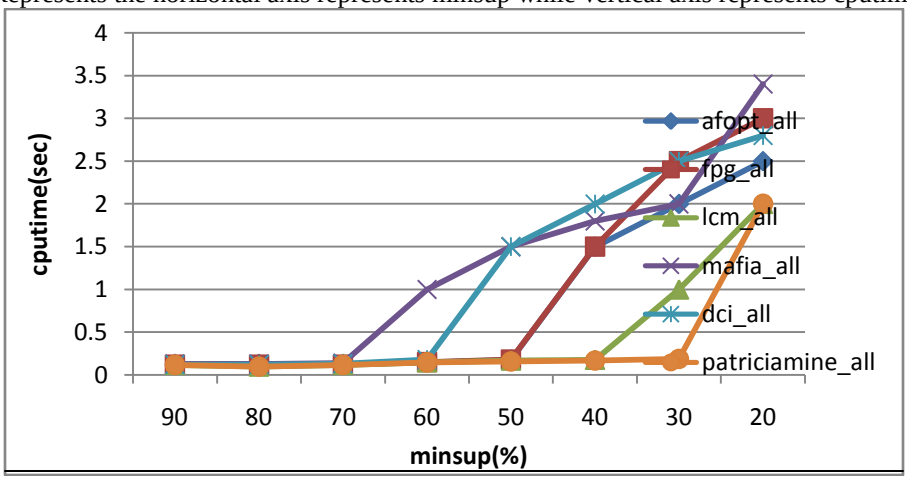

Fig 10 Chess All

As shown in fig. 10. Represents the horizontal axis represents minsup while vertical axis represents cputime.

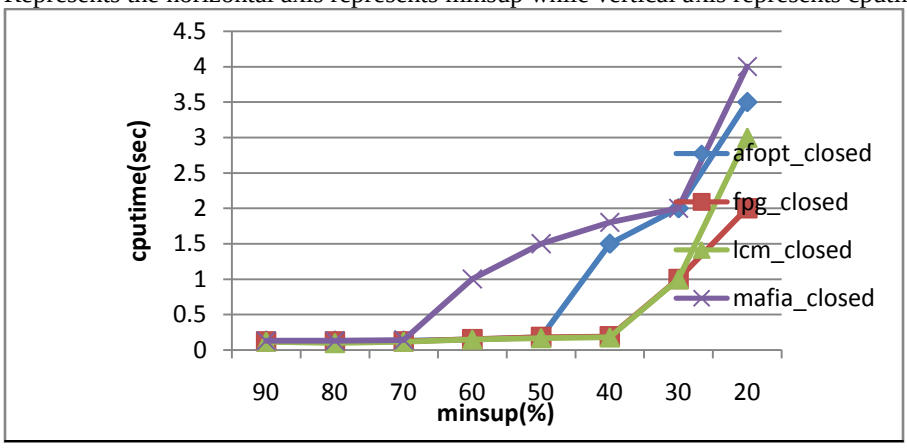

Fig 11 Chess Closed

As shown in fig.11. Represents the horizontal axis represents minsup while vertical axis represents cputime.

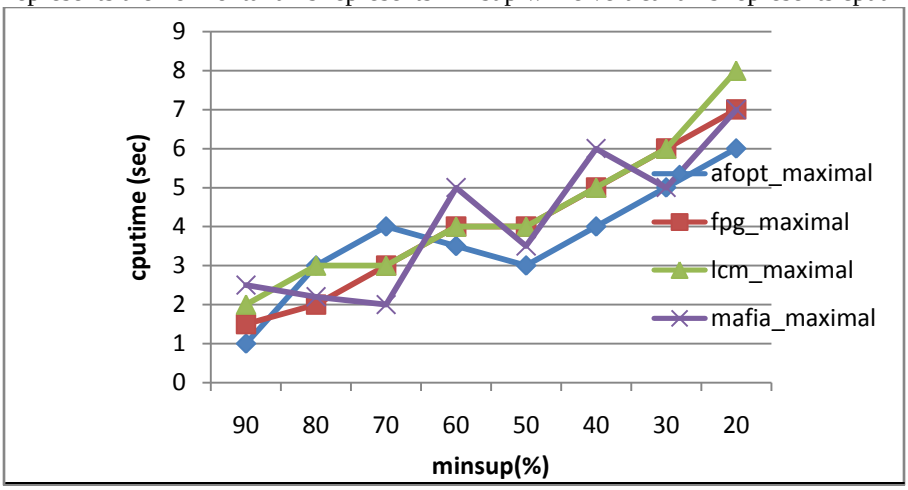

Fig 12 Chess Maximal

As shown in fig. 12. Represents the horizontal axis represents minsup while vertical axis represents cputime.

VI. CONCLUSION

In this paper we have implemented the algorithms proposed by Uno et al. [3] for discovering frequent itemsets, closed itemsets and maximal itemsets. The discovered itemsets can be used in many real time applications for making well informed

decisions. Thus the algorithms implemented in this paper have greater utility in solving various problems in the real world. The algorithms provide trends or patterns that can be used to derive actionable knowledge. We built a prototype application with graphical user interface to demonstrate the efficiency of these data mining algorithms. We tested the application with various kinds of datasets. The experimental results revealed that the application is very useful and the algorithms can be used in real world applications for expert decision making.

REFERENCES

[1] R. Agrawal and R. Srikant, \Fast Algorithms forvMining Association Rules in Large Databases," In Proceedings of VLDB '94, pp. 487{499, 1994.
[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonenvand A. I. Verkamo, \Fast Discovery of Association Rules," In Advances in Knowledge Discovery and Data Mining, MIT Press, pp. 307{328, 1996.
[3] Takeaki Uno, Masashi Kiyomi and Hiroki Arimura, "LCM ver. 2: E_cient Mining Algorithms for Frequent/Closed/Maximal Itemsets".
[4] E. Boros, V. Gurvich, L. Khachiyan, and K. Makino, \On the Complexity of GeneratingMaximal Frequent and Minimal Infrequent Sets," STACS 2002, pp. 133-141, 2002.
[5] D. Burdick, M. Calimlim, J. Gehrke, \MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases," In Proc. ICDE 2001, pp. 443-452, 2001.
[6] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu, \MAFIA: A PerformanceStudy of Mining Maximal Frequent Itemsets," In Proc. IEEE ICDM'03 Workshop FIMI'03,2003. (Available as CEUR Workshop Proc. series, Vol. 90, http://ceur-ws.org/vol-90)
[7] G. Grahne and J. Zhu, \E_ciently Using Pre_x-trees in Mining Frequent Itemsets," InProc. IEEE ICDM'03 Workshop FIMI'03, 2003. (Available as CEUR Workshop Proc. series, Vol. 90, http://ceur-ws.org/vol-90)
[8] J. Han, J. Pei, Y. Yin, \Mining Frequent Patterns without Candidate Generation," SIGMODConference 2000, pp. 1-12, 2000
[9] R. Kohavi, C. E. Brodley, B. Frasca, L. Mason and Z. Zheng, \KDD-Cup 2000 Organizers'Report: Peeling the Onion," SIGKDD Explorations, 2(2), pp. 86-98, 2000.
[10] R. J. Bayardo Jr., \E_ciently Mining Long Patterns from Databases", In Proc. SIGMOD'98,pp. 85{93, 1998.
[11] Takeaki Uno, Tatsuya Asai, Yuzo Uchida and Hiroki Arimura, "LCM: An Efficient Algorithm for Enumerating Frequent Closed Item Sets".

AUTHORS

Peddaboina Lingaraju is student of CMR College of Engineering and Technology, Hyderabad, Andhra Pradesh, INDIA. He has received B.Tech Degree Computer Science and Engineering and M.Tech Degree in Computer Science and Engineering. His main research interest includes Data Mining and Web Technologies..

K.Yellaswamy is working as an Assistant Professor in CMR College of Engineering and Technology, JNTUH, Hyderabad, Andhra Pradesh, India. He has completed M.Tech (C.S.E) from JNTUH. His main research interest includes Information Retrieval Systems and Web Technologies

B.Sivaiah is working as an Associate Professor in CMR College of Engineering and Technology, JNTUH, Hyderabad, Andhra Pradesh, India. He has completed M.Tech (C.S.E) from JNTUH. His main research interest includes Data Mining and Data Base Management Systems.

