ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 2, February 2019

Frequent Pattern Mining Algorithms PrePost, FIN, H-Mine – A Survey

M.Ranjanisindu[1], Dr.S.Gunasekaran[2], Ms.N.R.Deepa[3]

PG Scholar, Department of CSE, Coimbatore Institute of Engineering and Technology, Coimbatore, India[1]

Professor, Department of CSE, Coimbatore Institute of Engineering and Technology, Coimbatore, India[2]

Assistant Professor, Department of CSE, Coimbatore Institute of Engineering and Technology, Coimbatore, India[3]

ABSTRACT:Frequent pattern mining is an important data mining technique in the current research field to extract frequent itemsets from the database. These Frequent itemsets are very important in the mining tasks. Frequent pattern mining uses association rules to find the relationships among the items in the database. These techniques are very useful in the marketing fields such as cross marketing, market basket analysis and promotion assortment. There are several algorithms in the real world applications to perform frequent pattern mining. This paper presents a survey on the most popular frequent pattern mining algorithms PrePost, FIN and H-Mine. These algorithms use more efficient data structures to mine frequent itemsets from the database. PrePost uses N-list data structure to mine frequent itemsets and PPC tree to store frequent itemset information. FIN uses Nodeset data structure to mine frequent itemsets and POC tree to store frequentitemset information. H-Mine uses H-Struct data structure to mine frequent itemsets. In this paper we are using retail store transactional database that contains products as items and transactions of items.

KEYWORDS:Data mining, FIN, Frequent pattern mining, H-Mine, PrePost

I. INTRODUCTION

In the real world, tremendous amount of transaction data are created day by day. Increase in size of transactional database will lead to the mining task complex and unpredictable. Frequent pattern mining is used to extract the useful information from the transactional database. It processes the database and extracts the frequent itemsets.From the frequent itemsets, association rules are formed to perform mining tasks among the database.
Transactional database contains transaction details, that contains items involved in the transactions. Transactional database have two common layouts vertical layout and horizontal layout. In vertical layout there are two columns first column represents the item id and the second column contains transaction ids. In Horizontal layout first column contains transaction id and the second column contains item ids. In this paper we consider horizontal layout and the sample transactional database showed in Fig 1.

Transaction ID	Items
1001	a,c,g,f
1002	e,a,c,b
1003	e,c,b,i
1004	b,f,h
1005	b,f,e,c,d

Table 1. A Transactional database

Let Transaction database T is denoted as,
T = {T1, T2, T3,…. ,Tn} where each Tk ($1 \leq k \leq n$) is a Transaction which contains the set of items that Tk$\subseteq$ I.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 2, February 2019

Let I be the itemset for the transaction database then it is denoted as,
I = {i1, i2, i3,…. ,im}

A frequent pattern is a pattern which occurs in comparatively more transactions. A frequent itemset is an itemset whose support is greater than some user specified minimum support.

Apriori, Eclat and FP-Growth are the initial basic algorithms in the frequent pattern mining. Apriori algorithm was proposed by Agrawal and Srikant in 1994. It designed to operate on database containingtransactions. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. It uses bottom up approach. Eclat was proposed by Zaki in 2001. This operates on transactions database. It uses depth first search. Frequent Pattern Growth (FP-Growth) was introduced by Han, Pei and Yin in 2000. The FP-growth method adopts a highly condensed data structure called FP-tree to store databases and employs a divide-and-conquer strategy to mine frequent itemsets directly, which make it much more efficient than the Apriori-like method. Some algorithms are based on the FP-growth method. However, the FP-growth method becomes inefficient when datasets are sparse because FP-trees become very complex and larger.

In this paper the more efficient frequent pattern mining algorithms are analyzed. The section 2 describes the PrePost, FIN and H-Mine algorithms and section 3 and 4 analyses the algorithms and the conclusion section will give the overall report about the frequent mining algorithms.

II. RELATED WORK

Recent frequent pattern mining algorithms -PrePost, FIN, H-Mine algorithms are studied.

PrePost

PrePost was proposed by DENG ZhiHong, WANG ZhongHui and JIANG JiaJian in 2012. PrePost uses N-list data structure for mining frequent itemsets.

PrePost algorithm works as follows:

1. Construction of PPC tree and identifying all frequent-1 itemsets
2. Constructing the N-list of each frequent 1-itemset based on the PPC tree
3. Scanning PPC tree to find all frequent 2-itemsets
4. Mining all frequent k-itemsets.

Definition of PPC tree (Pre and Post order Coding tree) structure

1. It consists of one root labeled as “null”, and a set of item prefix subtrees as the children of the root.
2. Each node in the item prefix subtree consists of five fields: item-name, count, children-list, preorder, and post-order. item-name registers which item this node represents. count registers the number of transactions presented by the portion of the path reaching this node. children-list registers all children of the node. pre-order is the pre-order rank of the node. post-order is the post-order rank of the node.

Definition of N-list

For each node N in a PPC-tree, {(N.pre_order, N.post_order):count} the PP-code of N. the N-list of a frequent item is a sequence of all the PP-codes of nodes registering the item in the PPC-tree. The PP-codes are arranged in an ascending order of their pre-order values.

FIN

Fast mining frequent Itemsets using Nodesets (FIN) was proposed by Zhi-Hong Deng, Sheng-Long Lv in 2014. FIN uses Nodeset data structure for mining frequent itemsets. FIN directly discovers frequent itemsets in a set-enumaration tree. It adopts a pruning strategy to avoid repetitive search.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 2, February 2019

FIN algorithm works as follows:

1. Construction of POC tree and identifying all frequent-1 itemsets
2. Constructing the Nodeset of each frequent 1-itemset
3. Scanning the POC tree to find all frequent 2-itemsets
4. Mining all frequent k-itemsets.

Definition of POC tree (Pre Order Coding tree) structure:

1. The tree contains root labeled with "null" and children's of root with the set of item prefix sub trees
2. Item prefix sub tree node consists of four fields: item_name – item name registers which item this node represents. count - registers the number of transactions presented by the portion of the path reaching this node. children_list - registers all children of the node. pre_order - The pre order rank of the node.

Definition of Nodeset
In a POC tree each node N contains the pair value pre order and the count is represented as N-info of N. Nodeset of frequent item i is the sequence of all the N-info's of nodes in the POC tree.

H-Mine
Memory-Based HyperStructure Mining (H-Mine) was proposed by Pei et al in 2001. It uses pattern growth approach to discover frequent itemsets. H-Mine algorithm takes advantage of a novel hyper-linked data structure known as H-Struct that dynamically adjusts links in the mining process. This algorithm features a limited and precisely predictable space overhead that can run really fast in memory-based setting. Moreover, it can be scaled up to very large databases by database partitioning.
A header table H is created, where each frequent item entry has three fields: an item-id, a support count, and a hyper-link. When the frequent-item projections are loaded into memory, those with the same first item (in the order of F-list) are linked together by the hyper-links as a queue, and the entries in header table H act as the heads of the queues. It will process till the queues are empty since there is no frequent-item projection that begins with any of these items. Clearly, it takes one scan (the second scan) of the transaction database to build such a memory structure (called H-struct). Then the remaining of the mining can be performed on the H-struct only, without referencing any information in the original database.

III.METHODOLOGY

The frequent pattern mining algorithms were developed in Java with 4GB RAM capacity. Synthetic dataset from SPMF data generator was used to analyze the memory usage and the execution time of the frequent pattern algorithms.

IV.COMPARATIVE ANALYSIS

This section analyses the memory usage and the execution time of the algorithms with varying parameters. The unit of measurement for memory usage is MB and the execution time is Milliseconds. The parameters used here are transaction count, item count. The support count be the 10 transactions.

Case 1: Dataset with 100 itemsFIN, PrePost, H-Mine algorithms are compared with the 100 itemset. Memory usage and execution time comparison of the algorithms with 100 items illustrated in the Fig 1, Fig 2. Initially H-Mine is efficient but increase in transaction affects the H-Mine memory usage. Execution time of H-Mine is stable.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 2, February 2019

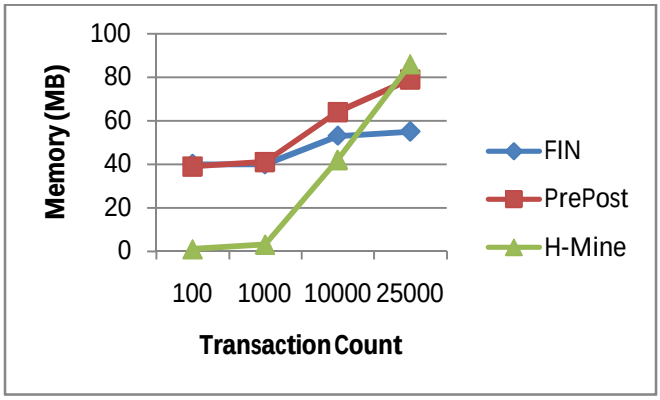

Fig 1: Memory usage comparison with item count 100

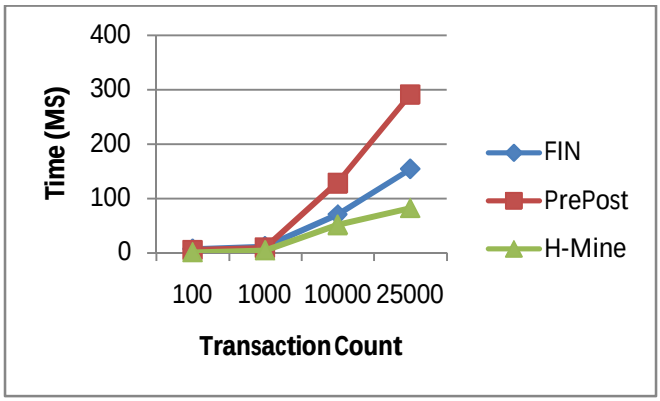

Fig 2: Execution time comparison with item count 100

Case 2: Dataset with 1000 itemsFIN, PrePost, H-Mine algorithms are compared with the 1000itemset and varying transaction count. Memory usage and execution time comparison of the algorithms with 1000 items illustrated in the Fig 3, Fig 4. Initially H-Mine is efficient but increase in transaction affects the H-Mine memory usage. Execution time of H-Mine is stable.

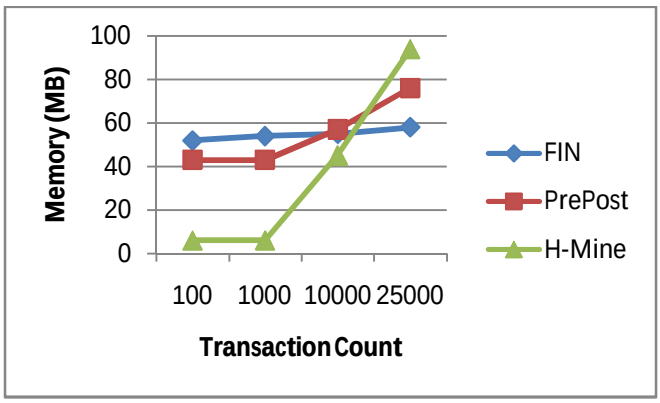

Fig 3: Memory usage comparison with item count 1000

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 2, February 2019

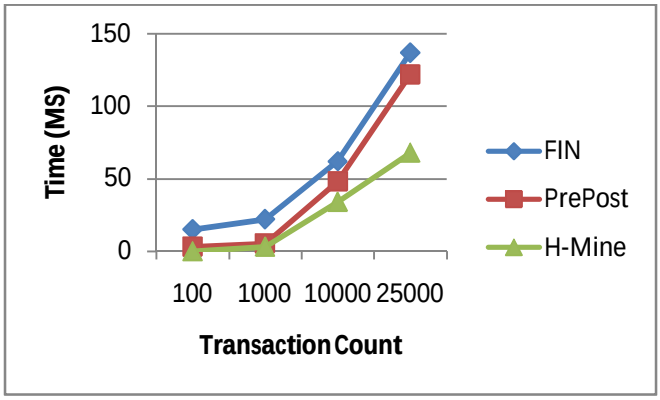

Fig 4: Execution time comparison with item count 1000

Case 3: Dataset with 10000 itemsFIN, PrePost, H-Mine algorithms are compared with the 10000 itemset and varying transaction count. Memory usage and execution time comparison of the algorithms with 10000 items illustrated in the Fig 5, Fig 6. H-Mine is efficient in memory usage, it ignores transaction count. Execution time of H-Mine is stable.

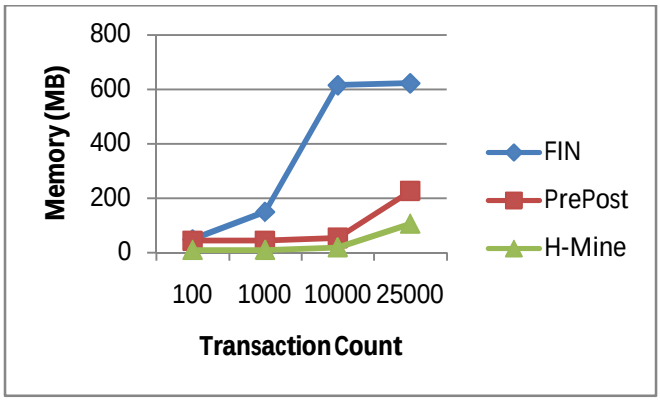

Fig 5: Memory usage comparison with item count 10000

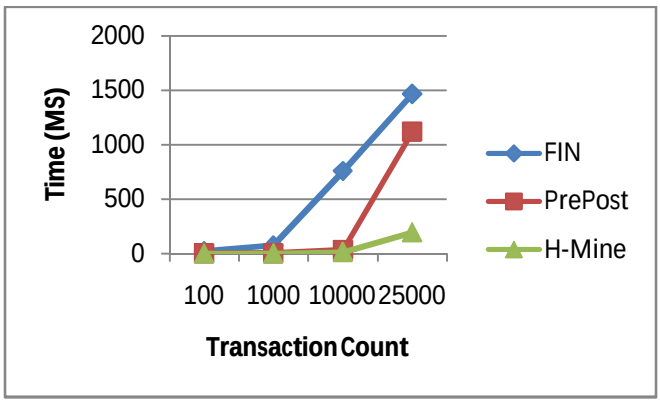

Fig 6: Execution time comparison with item count 10000

Case 4: Dataset with 25000 itemsFIN, PrePost, H-Mine algorithms are compared with the 25000 itemset and varying transaction count. Memory usage and execution time comparison of the algorithms with 25000 items illustrated in the

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 2, February 2019

Fig 7, Fig 8. H-Mine is efficient in memory usage, it ignores transaction count. Execution time of H-Mine is stable. FIN failed when the item count increases.

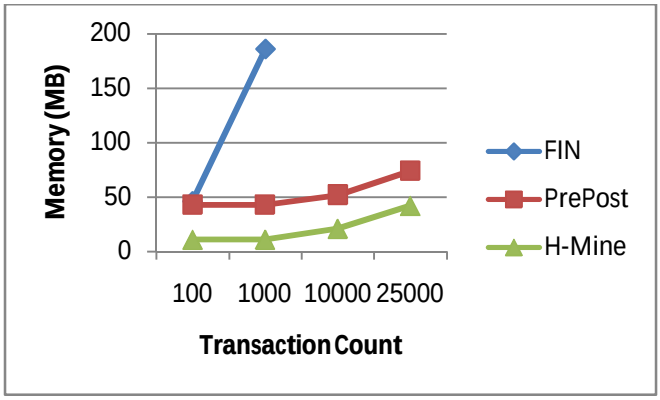

Fig 7: Memory usage comparison with item count 25000

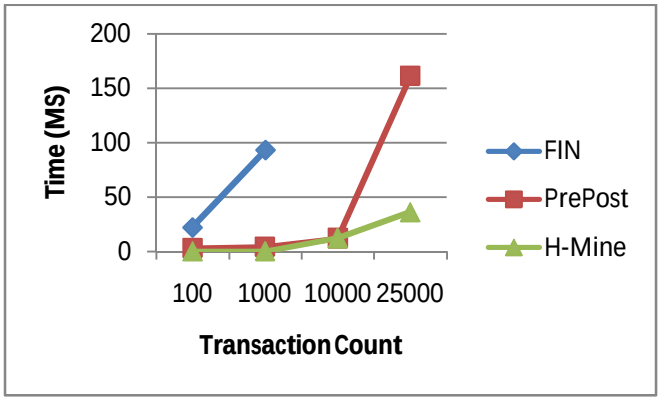

Fig 8: Execution time comparison with item count 25000

It appears that H-Mine algorithm uses more memory / time compared to the other two algorithms with minimum transaction count. But when the dataset has increasing transaction count, H-mine is found to be stable in terms of both memory and time.

V. CONCLUSION

The frequent pattern mining algorithms are analysed with the synthetic datasets. Each algorithm has its own advantages and disadvantages. This report can be used to choose the appropriate algorithm for the recommendation system depending on the use case. H-Mine algorithm is stable when increasing in transaction count.

REFERENCES

1. Agarwal, R.C., Agarwal, C.C., and Prasad, V.V.V., "A tree projection algorithm for generation of frequent item sets", Journal of Parallel and Distributed Computing, vol.61, pp.350–371, 2001.
2. Agrawal, R., and Srikant, R., "Fast Algorithms for Mining Association Rules in Large Databases", VLDB Conference, pp. 487–499, 1994.
3. Agrawal, R., Imielinski, T., and Swami, A., "Mining association rules between sets of items in large databases", ACM SIGMOD Conference, 1993.
4. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A.I., "Fast discovery of association rules, Advances in Knowledge Discovery and Data Mining", pp. 307–328, 1996.
5. Bhadoria, et., al., "Analysis of Frequent Itemset Mining on Variant Datasets", int.J.comp. Tech.appl., vol.5, pp.1328-1333.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 2, February 2019

6. Borgelt, C., “Efficient Implementations of Apriori and Eclat”, IEEE ICDM Workshop on Frequent Item Set Mining Implementations, CEUR Workshop Proceedings 90, Aachen, Germany, 2003.
7. Goswami, D.N., et., al., “An Algorithm for Frequent Pattern Mining Based On Apriori ”, (IJCSE) International Journal on Computer Science and Engineering, Vol.02, No.04, Pp. 942-947, 2010.
8. Rahul Mishra, et., al., “Comparative Analysis of Apriori Algorithm and Frequent Pattern Algorithm for Frequent Pattern Mining in Web Log Data.”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol.3(4), pp.4662 – 4665, 2012.
9. Deepak Garg, et., al., “Comparative Analysis of Various Approaches Used in Frequent Pattern Mining”, (IJACSA)International Journal of Advanced Computer Science andApplications, Special Issue on Artificial Intelligence.
10. Kanwal Garg, Deepak Kumar, “Comparing the Performance of Frequent Pattern Mining Algorithms”, (IJCA) International Journal of Computer Applications (0975 – 8887) Vol. 69, No.25, May 2013.
11. Jian Pei, Jiawei Han, Hongjun Lu, ShojiroNishio, Shiwei Tang and Dongqing Yang, "H-Mine: Fast and space-preserving frequent pattern mining in large databases IIE Transactions",pp:593–605, 2003.
12. Zhi-Hong Deng, and Sheng-Long Lv, "PrePost+ : An efficient N-lists-based algorithm for mining frequent itemsets via Children–Parent Equivalence pruning", Key Laboratory of Machine Perception (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China.
13. DENG ZhiHong, WANG ZhongHui, and JIANG JiaJian, "A new algorithm for fast mining frequent itemsets using N-lists", Key Laboratory of Machine Perception (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China.