International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 12 Issue 1–DECEMBER 2014.

# Frequent Itemsets Mining Algorithms: A Review

**A. Vaishnavi[a], B. Anantharaj[b], N. Saravanan[c], and N. Balaji[b]**

[a] PG Student, Department of Computer Science and Engineering, Thiruvalluvar College of Engg & Tech, Vandavasi, Tamil Nadu
[b] Asst. Professor, Department of Computer Science and Engineering, Thiruvalluvar College of Engg & Tech, Vandavasi, Tamil Nadu
[c] Asst. Professor, Department of Information and Technology, Thiruvalluvar College of Engg & Tech, Vandavasi, Tamil Nadu
csevaishu@gmail.com,
ananthu_arun72@yahoo.com,
mnsaran@hotmail.com
nbalajime1983@gamil.com

**ABSTRACT –**
Frequent Itemstets mining has been a plays an important role in the field of data mining research for over a decade. The most important usage of data mining is customer segmentation in marketing, shopping cart analyzes, management of customer relationship, campaign management, Web usage mining, text mining, player tracking and so on. This paper presents an overview of various algorithms for frequent itemsets in data mining and to motivate for the research. We have explored the unifying feature among the internal working of various mining algorithms. The comparative study of algorithms includes aspects like different support values.

**Keywords – Data Mining, Frequent Itemsets, Frequent pattern Mining, Association rules.**

## I. INTRODUCTION

Frequent itemsets play an important role in many data mining tasks such as the mining of association rules and classification [1]. Therefore,a lot of frequent itemset mining algorithms, such as AIS Algorithm (Agrawal et al. 1993)[3], Apriori Algorithm (Agrawal and Srikant 1994) [4], Multiple Dimensional ARM (Srikant and Agrawal 1996) [5], Maintaining of Association Rules (Cheung et al. 1997) [6], Multiple Concept Level ARM (Han and Kamber 2000) [7], Constraints based ARM (Pei and Han 2000) [8], (FP-Tree (Frequent Pattern Tree) Algorithm (Han et al. 2000)[9] , Rapid Association Rule Mining (RARM) (Das et al. 2001) [10], Hashing and Pruning (IHP) for mining association rules (John D. Holt and Soon M. Chung. 2002) [11], Fuzzy Grids Based Rules Mining Algorithm (FGBRMA), (Yi-Chung Hu et al. 2003) [12], Hierarchical Bisecting Medoids Algorithm (HBM) (Feng-Hsu Wang and Hsiu-Mei Shao. 2004) [13], Cluster-Based Association Rule (CBAR) (Yuh-Jiuan Tsay and Jiunn-Yann Chiang. 2005) [14], Gain based Association Rule Classification (GARC) (Guoqing Chen et al. 2006)[15], Classification Association Rule Mining (CARM) (Frans Coenen and Paul Leng. 2007) [16], Weighted Association Rules (WARs) (He Jiang et al. 2008) [17], Weighted Negative Association Rules (WNARs) (Yuanyuan Zhao et al. 2009) [18], Association Rules Mining based Alarm Correlation Analysis System (ARM-ACAS) (Tongyan Li and Xingming Li. 2010) [19], Mining Fuzzy Association Rules (WeiminOuyang et al. 2011) [20], Interesting Multiple Level Minimum Supports (IMLMS) Algorithm (IdhebaMohamad Ali O. Swesi et al. 2012) [21], Traditional Algorithm for Association

Mining Rules (AnjanaGosainet al. 2013) [22], Variable Neighbourhood Search (VNS) Algorithm (Yiyong Xiao et al. 2014) [23], Multi-Objective Particle Swarm Optimization Algorithm (MOPAR) (Vahid Beiranvand et al. 2014) [24].

The most basic and important task of data mining is the mining of frequent item sets, which are sets of items frequently purchased together in a transaction. Frequent item sets (in data mining literature, item sets is usually spelled as itemsets) represent intrinsic and important properties of the transactional database, and provide the foundation for many essential data mining tasks such as association/correlation analysis, pattern analysis, classification, cluster analysis and data warehousing (Han et al., 2011) [2].

Many people take data mining as a synonym for another popular term, Knowledge Discovery in Database (KDD). Alternatively other people treat Data Mining as the core process of KDD. The KDD processes are shown in Fig 1. [Han and Kamber 2000] [7]. Usually there are three processes. One is called preprocessing, which is executed before data mining techniques are applied to the right data. The pre-processing includes data cleaning, integration, selection and transformation. The main process of KDD is the data mining process, in this process different algorithms are applied to produce hidden knowledge.

After that comes another process called post processing, which evaluates the mining result according to users' requirements and domain knowledge. Regarding the evaluation results, the knowledge can be presented if the result is satisfactory, otherwise we have to run some or all of those processes again until we get the satisfactory result.

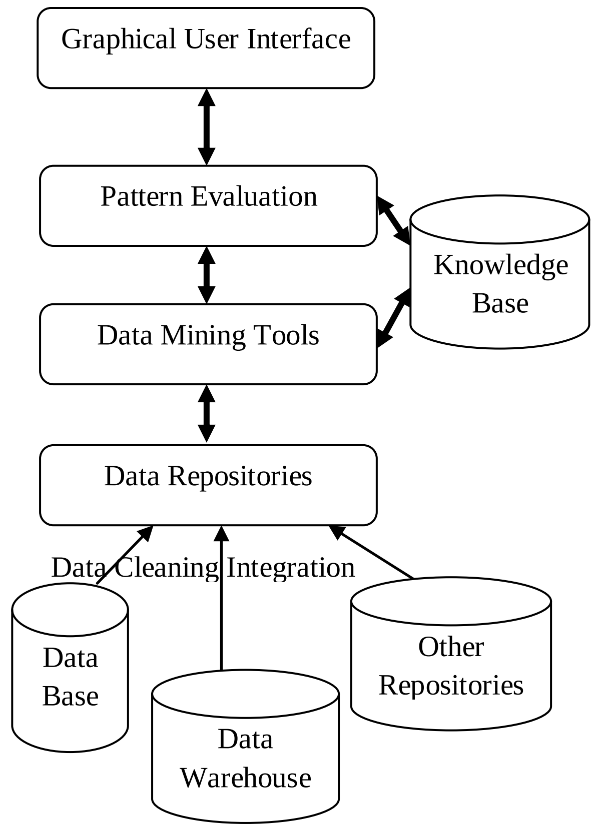


Fig. 1. Knowledge Discovery in Database processes

## II. FREQUENT ITEMSET MINING PROBLEM

Studies of Frequent Itemset (or pattern) Mining is acknowledged in the data mining field because of its broad applications in mining association rules, correlations, and graph pattern constraint based on
frequent patterns, sequential patterns, and many other data mining tasks. Efficient algorithms for mining frequent itemsets are crucial for mining association rules as well as for many other data mining tasks. The major challenge found in frequent pattern

mining is a large number of result patterns. As the minimum threshold becomes lower, an exponentially large number of itemsets are generated**.**

**Horizontal layout based techniques:**
**Apriori algorithm:**
Apriori is used to find all frequent itemsets in a given database DB. The key idea of Apriori algorithm is to
make multiple passes over the database. It employs an iterative approach known as a breadth-first search (level-wise search) through the search space, where k-itemsets are used to explore(k+1)-itemsets. The working of Apriori algorithm is fairly depends upon the Apriori property which states that" All nonempty subsets of a frequent itemsets must be frequent" [3] (Table 1).

**Table-1**
**Apriori algorithm parameters**

| **Storage Structure** | **Array based** |
|---|---|
| Technique | Use Apriori property and join and prune menthod |
| Memory utilization | Due to large amount of candidate are produced so require large memory space |
| Databases | Suitable for sparse datasets as well as dense datasets |
| Time | Execution time is more as time wasted in producing candidates at every time |

**Direct Hashing and Pruning (DHP):**
A DHP technique was proposed to reduce the number of candidates in the early passes Ck for k > 1 and thus the size of database [25]. In this method, support is counted by mapping the items from the candidate list into the buckets which is divided according to support known as Hash table structure. As the new itemset is encountered if item exist earlier then increase the bucket count else insert into new bucket. Thus in the end the bucket whose support count is less the minimum support is removed from the candidate set (Table 2).

**Table–2**
**Direct Hashing and Pruning algorithm parameters**

| **Storage Structure** | **Array based** |
|---|---|
| Technique | Use hashing technique for fining frequent itemsets |
| Memory utilization | Require less space at earlier passes but more in later stages |
| Databases | Suitable for medium databases |
| Time | Execution time is small for small databases |

**Partitioning algorithm:**
Partitioning algorithm is based to find the frequent elements on the basis partitioning of database in n parts [26]. It overcomes the memory problem for large database which do not fit into main memory because small parts of database easily fit into main memory (Table 3).

**Table–3**
**Partitioning algorithm parameters**

| **Storage Structure** | **Array based** |
|---|---|
| Technique | Partition the database for finding local frequent item first |
| Memory utilization | Each partition is easily occupy in main memory |
| Databases | Suitable for large databases |
| Time | Execution time is more because of finding locally frequent then globally frequent |

**Dynamic Itemset Counting (DIC):**
This algorithm also used to reduce the number of database scan [27]. It is based upon the downward disclosure property in which adds the candidate itemsets at different point of time during the scan. In this dynamic blocks are formed from the database marked by start points and unlike

the previous techniques of Apriori it dynamically changes the sets of candidates during the database scan. Unlike the Apriori it cannot start the next level scan at the end of first level scan, it start the scan by starting label attached to each dynamic partition of candidate sets (Table 4).

**Table–4**
**Dynamic Itemset Counting algorithm parameters**

| Storage Structure | Array based |
|---|---|
| Technique | Based upon dynamic insertion of candidate items |
| Memory utilization | Require different amount of memory at different point of time |
| Databases | Suitable for medium and low databases |
| Time | Execution time is small because dynamic itemset are added according to situation |

**Sampling algorithm:**
This algorithm is used to overcome the limitation of I/O overhead by not considering the whole database for checking the frequency[28]. It is just based in the idea to pick a random sample of itemset R from the database instead of whole database D. The sample is picked in such a way that whole sample is accommodated in the main memory (Table 5).

**Table–5**
**Sampling algorithm parameters**

| Storage Structure | Array based |
|---|---|
| Technique | Pick any random sample for checking frequency of whole database at lower threshold support |
| Memory utilization | Very less amount of memory is needed |
| Databases | Suitable for any kind of dataset but mostly not give accurate results |
| Time | Execution time is very much small |

**Vertical layout based technique: Eclat algorithm:** Eclat algorithm is basically a depth-first search algorithm using set intersection [29]. It uses a vertical database layout i.e. instead of explicitly listing all transactions; each item is stored together with its cover (also called tidlist) and uses the intersection based approach to compute the support of an itemset. In this way, the support of an itemset X can be easily computed by simply intersecting the covers of any two subsets Y, Z ⊆ X, such that Y U Z = X. It states that, when the database is stored in the vertical layout, the support of a set can be counted much easier by simply intersecting the covers of two of its subsets that together give the set itself (Table 6).

**Table–6**
**Eclat algorithm parameters**

| Storage Structure | Array based |
|---|---|
| Technique | Use intersection of transaction ids list for generating candidate itemsets |
| Memory utilization | Require less amount of memory compare to apriori if itemsets are small in number |
| Databases | Suitable for medium and dense datasets but not suitable for small datasets |
| Time | Execution time is small then apriori algorithm |

**FP-Growth algorithm:**
The most popular frequent itemset mining called the FP-Growth algorithm[30]. The problem of Apriori algorithm was dealt with, by introducing a novel, compact data structure, called frequent pattern tree, or FP-tree then based on this structure an FP-tree-

based pattern fragment growth method was developed. Essentially, all transactions are stored in a tree data structure (Table 7).

**Table–7**
**FP-Growth algorithm parameters**

| **Storage Structure** | **Array based** |
|---|---|
| Technique | It constructs conditional frequent pattern tree and conditional pattern base from database which satisfy the minimum support |
| Memory utilization | Due to compact structure and no candidates generation require less memory |
| Databases | Suitable for large and medium datasets |
| Time | Execution time is large due to complex compact data structure |

**H-mine algorithm:**
H-mine algorithm is the improvement over FP-tree algorithm as in H-mine projected database is created using in-memory pointers [31]. H-mine uses an H-struct new data structure for mining purpose known as hyperlinked structure. It is used upon the dynamic adjustment of pointers which helps to maintain the processed projected tree in main memory therefore H-mine proposed for frequent pattern data mining for data sets that can fit into main memory (Table 8).

**Table–8**
**H-mine algorithm parameters**

| **Storage Structure** | **Array based** |
|---|---|
| Technique | It uses the hyperlink pointers to store the partitioned projected database in main memory |
| Memory utilization | Memory is utilized according to needs and partitions of projected database |
| Databases | Suitable for sparse and dense datasets |
| Time | Execution time is large then FP-tree and others because of partition the database |

## III. RESULT AND DISCUSSION

**Data set:** The dataset was obtained from the UCI repository of machine learning databases12. The characteristics of adult dataset selected for the experiment (Table 9).

**Table–9**
**The characteristics of adult dataset**

| **File name** | **Number of Records** | **Number of Columns** |
|---|---|---|
| adult.D14.N48842.C2.num | 48842 | 14 |

**Result analysis:** A detailed study to assess the performance of Apriori, Eclat and FP-Growth algorithms. The performance metrics is the total execution time taken and the number of frequent itemsets generated using an adult datasets. For this comparison also same dataset were selected as for the experiments with minimum support ranging from 30% to 70%.

**Table–10**
**Total execution time using Adult dataset**

| **Support** | **Total Execution Time in Seconds** | | | | |
|---|---|---|---|---|---|
| | **Apriori** | **Eclat** | **FP-Growth** | **Relim** | **SaM** |
| 30 | 9.85 | 0.54 | 0.56 | 0.49 | 0.47 |
| 40 | 6.72 | 0.49 | 0.5 | 0.44 | 0.44 |
| 50 | 4.51 | 0.45 | 0.49 | 0.42 | 0.41 |
| 60 | 2.69 | 0.44 | 0.48 | 0.4 | 0.4 |
| 70 | 1.7 | 0.4 | 0.42 | 0.39 | 0.37 |

The total execution time of Apriori, Eclat, FP-Growth, Relim and SaM algorithms with different support threshold using an adult data set. The execution time is decreased when the support threshold increased. The SaM algorithm is better than the Apriori and near the Relim. The difference between execution time of these algorithms decreases with increasing a support threshold. The Apriori takes more time as that compared to other algorithms (Table 10).

## IV. CONCLUSION

In this paper we briefly reviewed the existing frequent itemsets mining in data mining algorithms. This review would be helpful to researchers to focus on the various issues of data mining. A comparison framework has developed to allow the flexible comparison of Apriori, Eclat and FP-growth algorithms. Using this framework this paper presented the comparative performance study of these algorithms such as, Apriori, Eclat and FP-growth. The execution time is decreased when the support threshold increased. The Eclat algorithm is better than the Apriori and near the FP-Growth. This study is prepared to our new project titled Top-K frequent itemsets mining using compressed POC Tree.

## V. REFERENCES

[1] Hai Duong, TinTruong, BayVo, "AN efficient method for mining frequent itemsets with double constraints", Engineering Applications of Artificial Intelligence 27(2014)148–154, Elsevier.

[2] Han, J., Kamber, M., & Pei, J. (2011). Data mining: Concepts and techniques (3rd ed.). San Francisco, CA: Morgan Kaufmann.

[3] Agrawal, R., Imielinski, T., and Swami, A. N. 1993. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, P. Buneman and S. Jajodia, Eds. Washington, D.C., 207-216.

[4] Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules. In Proc. 20th Int. Conf. Very Large Data Bases, VLDB, J. B. Bocca, M. Jarke, and C. Zaniolo, Eds. Morgan Kaufmann, 487{499.

[5] Srikant, R. and Agrawal, R. 1996. Mining quantitative association rules in large relational tables. In Proceedings of the 1996 ACM SIGMOD international conference on Management of data. ACM Press, 1-12.

[6] Cheung, D. W.-L., Lee, S. D., and Kao, B. 1997. A general incremental technique for maintaining discovered association rules. In Database Systems for Advanced Applications. 185-194.

[7] Han, J. and Kamber, M. 2000. Data Mining Concepts and Techniques. Morgan Kanufmann.

[8] Pei, J. and Han, J. 2000. Can we push more constraints into frequent pattern mining? In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM Press, 350-354.

[9] Han, J. and Pei, J. 2000. Mining frequent patterns by pattern-growth: methodology and implications. ACM SIGKDD Explorations Newsletter 2, 2, 14-20.

[10] Das, A., Ng, W.-K., and Woon, Y.-K. 2001. Rapid association rule mining. In Proceedings of the tenth international conference on Information and knowledge management. ACM Press, 474 - 481.

[11] John D. Holt, Soon M. Chung, "M ining association rules using inverted hashing and pruning", Information Processing Letters, Volume 83, Issue 4, 31 August 2002, Pages 211-220, Elsevier.

[12] Yi-Chung Hu, Ruey-Shun Chen, Gwo-Hshiung Tzeng, "Discovering fuzzy association rules using fuzzy partition methods", Knowledge-Based Systems, Volume 16, Issue 3, April 2003, Pages 137-147, Elsevier.

[13] Feng-Hsu Wang, Hsiu-Mei Shao, "E ffective personalized recommendation based on time-framed navigation clustering and association mining", Expert Systems with Applications, Volume 27, Issue 3, October 2004, Pages 365-37, Elsevier.

[14] Yuh-Jiuan Tsay, Jiunn-Yann Chiang, “BAR: an efficient method for mining association rules”, Knowledge-Based Systems, Volume 18, Issues 2–3, April 2005, Pages 99-10, Elsevier.

[15] Guoqing Chen, Hongyan Liu, Lan Yu, Qiang Wei, Xing Zhang,” A new approach to classification based on association rule mining “,Decision Support Systems, Volume 42, Issue 2, November 2006, Pages 674-68, Elsevier.

[16] Frans Coenen, Paul Leng, “T he effect of threshold values on association rule based classification accuracy”, Data & Knowledge Engineering, Volume 60, Issue 2, February 2007, Pages 345-360, Elsevier.

[17] He Jiang; Yuanyuan Zhao; Xiangjun Dong, "Mining Positive and Negative Weighted Association Rules from Frequent Itemsets Based on Interest," Computational Intelligence and Design,ISCID '08. International Symposium on , vol.2, no., pp.242,245, 17-18 Oct. 2008.

[18] YuanyuanZhao,He Jiang; RunianGeng; Xiangjun Dong, "Mining Weighted Negative Association Rules Based on Correlation from Infrequent Items," Advanced Computer Control,ICACC '09. International Conference on, vol., no., pp.270,273, 22-24 Jan. 2009.

[19] Tongyan Li, Xingming Li, “Novel alarm correlation analysis system based on association rules mining in telecommunication networks“, Information Sciences, Volume 180, Issue 16, 15 August 2010, Pages 2960-2978, Elsevier.

[20] WeiminOuyang; Qinhua Huang, "Mining direct and indirect fuzzy association rules with multiple minimum supports in large transaction databases," Fuzzy Systems and Knowledge Discovery (FSKD), Eighth International Conference on , vol.2, no., pp.947,951, 26-28 July 2011.

[21] WeiminOuyang,” Mining Positive and Negative Fuzzy Association Rules with Multiple Minimum Supports”, International Conference on Systems and Informatics, 2012.

[22] AnjanaGosain and ManeelaBhugra,” A Comprehensive Survey of Association Rules On Quantitative Data In Data Mining”, IEEE Conference on Information and Communication Technologies, 2013.

[23] Yiyong Xiao, Yun Tian, Qiuhong Zhao, “Optimizing frequent time-window selection for association rules mining in a temporal database using a variable neighbourhood search”, Computers & Operations Research, Volume 52, Part B, December 2014, Pages 241-250, Elsevier.

[24] Vahid Beiranvand, Mohamad Mobasher-Kashani, Azuraliza Abu Bakar, “Multi-objective PSO algorithm for mining numerical association rules without a priori discretization “, Expert Systems with Applications, Volume 41, Issue 9, July 2014, Pages 4259-4273, Elsevier.

[25] Park. J.S, Chen M.S., Yu P.S., An effective hash-based algorithm for mining association rules, In Proc. ACMSIGMOD Int’l Conf. Management of Data (SIGMOD), 175–186 (1995)

[26] Savasere E. Omiecinski and Navathe S., An efficient algorithm for mining association rules in large databases, In Proc. Int’l Conf. Very Large Data Bases (VLDB), 432– 443 (1995)

[27] Brin. S., Motwani. R., Ullman. J.D. and Tsur. S., Dynamic itemset counting and implication rules for market basket analysis, In Proc. ACM-SIGMOD Int’l Conf. Management of Data (SIGMOD), 255–264 (1997)

[28] C Toivonen. H., Sampling large databases for association rules, In Proc. Int’l Conf. Very Large Data Bases (VLDB), 134–145 (1996)

[29] Borgelt C., Efficient Implementations of Apriori and Eclat, In Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (2003)

[30]. Han J., Pei H., and Yin. Y., Mining Frequent Patterns without Candidate Generation, In Proc. Conf. on the Management of Data (2000)

[31] Pei. J, Han. J, Lu. H, Nishio. S. Tang. S. and Yang. D., Hmine: Hyper-structure mining of frequent patterns in large databases, In Proc. Int'l Conf. Data Mining (2001)

[32]. Blake C.L. and Merz. C.J., UCI Repository of Machine Learning Databases, Dept. of Information and Computer Science, University of California at Irvine, CA, USA (1998)