*Journal of Advances and Scholarly Researches in Allied Education*
*Vol. XI, Issue No. 22, July-2016, ISSN 2230-7540*

# An Overview on the Design of Frequent Pattern Mining Algorithms

**Krishna Chaitanya Sanagavarapu***

Student, Masters in Information Systems and Security, University of the Cumberlands, KY

***Abstract – Association rule mining is one of the most important data mining problems. The purpose of association rule mining is the discovery of association relationship among a set of items. The mining of association rule include two sub problems (1) finding all frequent Item-sets that appear more often than a minimum support threshold and (2) generate association rules using these frequent Item-sets. The first subproblem plays an important role in association rules mining. This paper provides an overview on the design of frequent pattern mining algorithms***

***Index Terms : Association Rule Mining, Pattern Mining***

----------X----------

## I. INTRODUCTION

After the introduction by Aggrawal, many algorithms for frequent Item-set mining have proposed for association rule discovery. These algorithms can be grouped into two types, namely, the candidate generate-and-test approach and the pattern growth approach. Examples of the first type include algorithms proposed among which Apriori is the most frequently used algorithm. The Apriori algorithm has the advantage of reducing the size of candidate sets but performs many scans to check the candidate's support, which is in most of the cases as long as the length of patterns. So, when there exist a large number of frequent patterns and/or long patterns, candidate generate-and-test approach may suffer a large overhead of I\0.

The second class comprises pattern-growth methods, over the past few years, several pattern-growth algorithms have been proposed, such as FP-Growth Tree-projection H-Mine and COFI of which, FP-Growth has gained much popularity. A pattern-growth algorithm uses the FP-Tree to store the database, instead of generating candidates, it mining the FP-Tree recursively by building conditional trees that are of the same order of magnitude in number as the frequent pattern.

However, this massive creation of conditional trees makes these algorithms not scalable to mine large datasets beyond few millions. The concerns identified are summarized below:

(i) Fails with huge sized temporal databases

(ii) Selection of support

(iii) Number of redundant rules is very high

(iv) Number of scans

(v) Generates huge number of association rules

This study aims to solve these concerns, by designing two algorithms, namely, Temporal Mining using Enhanced Aprior (TMEA) algorithm (Phase II) and Temporal Mining using Enhanced FP-Growth (TMEF) algorithm (Phase III). Both of these algorithms are developed to enhance the process of finding frequent Item set and association rules. Additionally, to farther improve the quality of the mining output, these two enhanced versions are unified to form hybrid or ensemble model, called Temporal Mining using Ensemble Apriori and FP-Tree (TMEAF) algorithm. First, the traditional Apriori and FP-Growth algorithms are enhanced and converted to work with temporal database.

## II. ENHANCED APRIORI ALGORITHM (BTAM-A)

The problem of the discovery of association rules comes from the need to discover patterns in transaction data, which exhibits time dimension. According to Ralph (1996), "The time dimension is the one dimension virtually guaranteed to be present in every data warehouse, because virtually every data warehouse is a time series". The design and development of association rules discovery algorithm that can identify frequent item sets from large temporal databases is a challenging task and has attracted several researchers and academicians.

It has been estimated that 80% of their existing applications contain some form of temporal data. Inclusion of time characteristics in snapshot databases promises to deliver major productivity savings for many new developments. A detailed study related to temporal characteristics with snapshot data was published in and the current research status of association rule discovery on temporal data was published.

While including time parameter with large sized transactional database, information related to a particular event or a product may not exist throughout the data gathering period. For example, a new product introduced after the beginning of data gathering. These events form discontinuities during the mining process. These transactions, are vital during association rule discovery, are often not included in any rule because of support restrictions in many scenarios.

Consider for example the situation where the total number of transaction is 3, 00,00,000 with minimum support 0.5%. Also assume that a particular product should appear in at least 150,000 transactions to be considered as frequent Item-set. Moreover, suppose that these transactions were recorded during the last 30 months, at 10, 00,000 per month. Further, from a transaction database consisting of 1,20,000 transaction, let the sales details of a product over 30 months that satisfy minimum support are 5000 transactions per month and consider a new product that was incorporated in the last 6 months and that appears in 20,000 transactions per month.

In this scenario, these products are not considered frequent, even though it is four times as popular as the first. However, if rule discovery process is restricted to the transactions generated since the product appeared in the market, its support might be above the stipulated minimum. In the example, the support for the new product is 2%, relative to its lifetime, since in 6 months the total of transactions would be about 60, 00,000 and this product appears in 1,20,000 of them. Therefore, these new products would appear in interesting and potentially useful association rules. An easiest manner to solve the above problem is to incorporate time in the association rule discovery process. These are termed as Temporal Association Rules (TAR).

As noted previously, the major concern during frequent item set mining and association rule generation tasks is the number of rules generated. The BTAM algorithm handles this by

- Eliminating outdated rules (specified according to user criteria)
- Deleting obsolete item sets by treating them over a function of their lifetime

Incorporation of the above will reduce the amount of time spent during frequent item set discovery and will have a direct impact on the number of association rules discovered.

The BTAM algorithm is designed as an extension to non-temporal model where the basic idea is to reduce the time related to the search of frequent item sets to the lifetime of the item set's members. Thus, each association rule has an associated time frame, corresponding to the lifetime of the items participating in the rule. If the extent of a rule's lifetime exceeds a minimum time, stipulated by the user, the algorithm then analyzes to determine whether the rule is frequent in that period or not. This design facilitates to find rules that are not discovered by the traditional algorithm.

This approach takes into account the items' lifespan, which is defined as the period between the first and the last time the item appears in transactions in the database. The support of an Item-set in the interval is defined by its lifespan and defines temporal support as the minimum interval width. This approach differs from the others in that it is not necessary to define interval or calendars since the lifespan is intrinsic to the data. This section presents the steps involved while implementing this design to the Apriori algorithm. The general steps involved in BTAM are given in Figure 1.

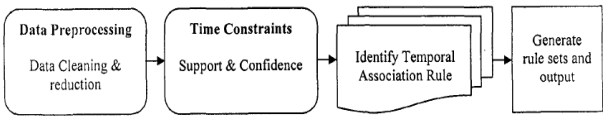


**Figure 1 : Architecture of BTAM**

**Definitions**

Let T = { . . ., to, ti, t2, . . . } be a set of times that are countable infinite and over which a linear order <T is defined. Here, the notational ti <T tj means ti occurs before or is earlier than tj (Tansel *et al.,* 1993). It is assumed that T is isomorphic to N (natural numbers) and the description is restricted to closed intervals [ti, tj].

Let R = {Al, ..., Ap} where the Ai's are called items, d is a collection of subsets of R called the transaction database. Each transaction s in d is a set of items such that s c R. The definition of R includes every item of d, independently of the moment in which it appears. A timestamp, ts, is associated to s which represents the valid time of transaction s. Every item has a period of life or lifespan in the database, which explicitly represents the temporal duration of the item information, that is, the time in which the item is relevant to the user. The lifespan of an item Ai is given by an interval [ti, t2], with ti < *X2.*

Let Ai be an item of R. A lifespan defined by a time interval [Ai.ti, Ai.ti] or simply [ti, *Xi]* is associated with each item Ai and database d, if Ai is understood. L : Ai

-> 2^ is a function assigning a lifespan to each item Ai in R. This lifespan is referred to as LA,. Then, Ld, the lifespan of d, is defined as Ld = u LAi, V i.

Let X e R a set of items, s contains X, or X is verified in s, if X c s. The set of transactions in d that contain X is indicated by V(X, d) = {s | s e d A X c s}. If the cardinality of X is k, X is called a k-Item-set. The lifespan of a k-Item-set X, with k > 1, is [t, t'] where t = max{ti| [ti, tz] is the lifespan of an item Ai in X} and t' = min{t2| [ti, ta] is the lifespan of an item Ai in X }. As set operations are valid over life spans, then the lifespan Lx of the k-Item-set X, where X is the union of the (k-l)-item sets V and W with lifespans Lv and Lw, respectively, is given by Lx = Lv n Lw-

Let X c R be a set of items and Lx its lifespan. If d is the set of transactions of the database, then dtx is the subset of transactions of d whose timestamps tj e Lx.

Then, |dLx| indicates the number of transactions of dix-

In the non-temporal association rules model the definition of support is described as follows. The support of X in d, denoted by s(X, d), is the fraction of the transactions in d that contains X: |V(X, d)| / |d|. The frequency of a set X is its support. Given a support threshold a e [0, 1], X is frequent if s(X, d) > a . In this case, it is said that X has minimum support.

In BTAM, the definition of support is modified to incorporate time, referred to as temporal support, in order to determine whether an Item-set is frequent by computing the ratio between the number of transactions that contain the Item-set and the number of transactions in the database such that their valid time is included in the Item-set's lifespan. Evidently, the need to filter items and then the Item-sets with very short life is required. For example, consider an item that has been sold only once, basic support calculation would result with 100% support. Thus, to handle such cases, the temporal support is defined as the amplitude of the lifespan of an Item-set. A threshold is also defined for the temporal support (T).

If Ld is the lifespan of the database and |Ld| is its duration, then the threshold of the temporal support T is a fraction of |Ld|. Thus, for example, if the transactions correspond to a period of n months, x, a fraction of n months, represents a lower bound for the temporal support of an Item-set.

If the quantity of transactions of the database is |d|, then |d|. x / |Ld| would give us an approximation to the minimum quantity of transactions to be considered as sample size. Then |d|.T / |Ldi should be a statistically significant value, at the user's criteria. On the other hand, the user could specify a time instant to, such that any item whose lifespan is [tl, t2] and t2 < to is considered obsolete. The new definition for support in the temporal model incorporating these is given below.

The support of X in d over its lifespan Lx, denoted s(X, Lx, d), is the fraction of transactions in d that contains X during the interval of time corresponding to Lx: |V(X, d)| / Idtxl- The frequency of a set X is its support. Given a threshold of support e [0, 1] and a threshold of temporal support T, X is frequent in its lifespan Lx if s(X, Lx, d) > a and |Lx| > T . In this case, it is said that X has minimum support in Lx-The support threshold or frequency a is a parameter given by the user and is dependent on the application. Likewise, the temporal support threshold *x* is given by the user. The above definitions are explained using the following example.

(1) Example

Let R = {A, B, C, D, E, F, G, H, 1} be the transaction database and let d be composed by the following, six transactions:

sl = {A,C,F,H,I}, t: 1

s2 = {A, B, C, G}, t: 2

s3 = {B,C,D,G,I}, t:3

s4 = {A,C,I}, t:4

s5 = {C,D,E,H,I}, t:5

s6 = {A, D, F, G}, t: 6

Let minimum support a = 0.45 and minimum temporal support x = 3. The problem is to find the frequent Item-set X, resulting in XL = {A}, Lxi = [1, 6], since A is found between si and s6, which have stamped times 1 and 6, respectively. The support of {A} is computed as s({A}, L{A}, d) = |V({A}, d)| / |d[i.6]| = 4 / 6 = 0.67. The temporal support of A is |LA| = 6. Then XL = {A} is frequent because s({A}, L{A}, d) - 0.67 > a and |LA| = 6 > T. The same method is used to find the frequent Item-sets as listed below and indicates just their lifespan:

X2 = {C},Lx2 = [l,5]

X3 = {D},Lx3 = [3,6]

X4 = {G},Lx4 = [2,6]

X5 = {I},Lx5 = [l,5]

X6 = {A,C},Lx6 = [l,5]

X7={C,D},Lx7 = [3,5]

X8 = {C,I},Lx8 = [l,5]

The empty set 0 is trivially frequent, so it is not considered as it is not interesting. A temporal association rule expresses a set of items that tends to appear along with another set of items in the same

transactions, in a specific time fi-ame. It is defined as follows.

A Temporal Association Rule for d is an expression of the form X => Y [ti, ta], where X c R , Y c R \ X and [ti, t2] is a time frame corresponding to the lifespan of X u Y expressed in a granularity determined by the user. A temporal association rule has three factors associated with it: support, temporal support and confidence.

The confidence of a rule X => Y [ti, t2], denoted by conf(X => Y, [tj, ta], d) is the conditional probability that a transaction of d, randomly selected in the time frame [ti, t2], that contains X also contains Y (Equation 5.1).

conf(X => Y,[t,, t2], d) = s(X U Y, Lxu Y, d) / s(X, Lxu Y, d)

where LxuY= {[ti,t2]}.

The temporal association rule X => Y [ti, t2] holds in d with support s, temporal support |Lxu Y| and confidence c if s% of the transactions of d contain X u Y and 0% of the transactions of d that contain X also contain Y, in the time frame [ti, t2]. Given a set of transactions d and minimum levels of support, temporal support and confidence, the problem of temporal association rule discovery is to generate all the association rules that have at least the given support, temporal support and confidence.

Following with the previous example, let the level of minimum confidence 0 is established as 0.7. From the frequent set {A, C}, two possible rules can be considered. The two rules are A => C [1, 5] and C =^ A [1, 5]. The first has confidence conf(A => *C,* [1, 5], d) = (3/5) / (3/5) =1 .0 which is superior to the minimum 9 = 0.7. The second has confidence conf(C =^ A,[l,5], d) = (3/5) / (5/5) = 0.6 and therefore, it is discarded.

## III. TEMPORAL RULES DISCOVERY

The discovery of all the association rules in a transaction set d can be made in two modules. The first module finds every set of items (item sets) X c R that is frequent, that is, their frequency exceeds the established minimum support a. The second module uses the frequent Item-sets X to find the rules and test for every Y c X, with Y ?i 0 , if the rule X \ Y => Y satisfies with enough confidence and determines whether it exceeds the established minimum confidence 9.

In order to include the temporal aspects, the above module is modified as below. The first module, now, finds every Item-set X e R such that X is frequent in its lifespan Lx, that is, s(X, Lx, d) > a and |Lx| > x . The second module, now, uses the frequent Item-sets X to find the rules and verify for every Y c X, with Y ^t 0 , if the rule X \ Y ^ Y [t], t2] is satisfied with enough confidence, in other words, exceeds the minimum confidence 9 established in the interval [ti, t2].

## IV. CONCLUSION

Many authors have compared the performance of these first and second types of approaches. According to their results, the second type of approaches is more efficient but needs more memory to story the intermediate data structure in opposition to the first type of approaches. This paper provides an overview on the design of frequent pattern mining algorithms

## REFERENCES


1. Sugandhi Maheshwaram (2016). “A Comprehensive Review on the Implementation of Big Data Solutions” in “International Journal of Information Technology and Management”, Vol. XI, Issue No. XVII, [ISSN : 2249-4510]

2. Sriramoju Ajay Babu, Dr. S. Shoban Babu (2014). “Improving Quality of Content Based Image Retrieval with Graph Based Ranking” in “International Journal of Research and Applications”, Volume 1, Issue 1, Jan-Mar 2014 [ISSN : 2349-0020]

3. Dr. Shoban Babu Sriramoju, Ramesh Gadde (2014). “A Ranking Model Framework for Multiple Vertical Search Domains” in “International Journal of Research and Applications” Vol 1, Issue 1,Jan-Mar 2014 [ ISSN : 2349-0020].

4. Mounika Reddy, Avula Deepak, Ekkati Kalyani Dharavath, Kranthi Gande, Shoban Sriramoju (2014). “Risk-Aware Response Answer for Mitigating Painter Routing Attacks” in “International Journal of Information Technology and Management”, Volume VI, Issue I, [ISSN : 2249-4510]

5. Guguloth Vijaya, A. Devaki, Dr. Shoban Babu Sriramoju (2016). “A Framework for Solving Identity Disclosure Problem in Collaborative Data Publishing” in “International Journal of Research and Applications”, Volume 2, Issue 6, 292-295, Apr-Jun 2016 [ISSN : 2349-0020]

6. Shoban Babu Sriramoju (2015). “A Framework for Keyword Based Query and Response System for Web Based Expert Search” in “International Journal of Science and Research” Index Copernicus Value :78.96 [ ISSN : 2319-7064 ].

7. Sriramoju Ajay Babu, Dr. S. Shoban Babu (2014). “Improving Quality of Content Based Image Retrieval with Graph Based Ranking” in “International Journal of Research and Applications”, Volume 1, Issue 1, [ISSN: 2349-0020]

8. Dr. Shoban Babu Sriramoju, Ramesh Gadde (2014). “A Ranking Model Framework for

Multiple Vertical Search Domains" in "International Journal of Research and Applications" Vol 1, Issue 1, [ISSN: 2349-0020].

9. Mounika Reddy, Avula Deepak, Ekkati Kalyani Dharavath, Kranthi Gande, Shoban Sriramoju (2014). "Risk-Aware Response Answer for Mitigating Painter Routing Attacks" in "International Journal of Information Technology and Management", Volume VI, Issue I, [ISSN: 2249-4510]

10. Shoban Babu Sriramoju (2014). "An Application for Annotating Web Search Results" in "International Journal of Innovative Research in Computer and Communication Engineering" Vol. 2, Issue 3, [ISSN (online): 2320-9801, ISSN(print) : 2320-9798]

11. Shoban Babu Sriramoju (2014). "Multi View Point Measure for Achieving Highest Intra-Cluster Similarity" in "International Journal of Innovative Research in Computer and Communication Engineering" Vol. 2, Issue 3, [ISSN(online) : 2320-9801, ISSN(print): 2320-9798]

12. Mounica Doosetty, Keerthi Kodakandla, Ashok R, Shoban Babu Sriramoju (2012). "Extensive Secure Cloud Storage System Supporting Privacy-Preserving Public Auditing" in "International Journal of Information Technology and Management", Volume VI, Issue I, [ISSN : 2249-4510]

13. Shoban Babu Sriramoju, Madan Kumar Chandran (2014). "UP-Growth Algorithms for Knowledge Discovery from Transactional Databases" in "International Journal of Advanced Research in Computer Science and Software Engineering", Vol. 4, Issue 2, [ISSN : 2277 128X]

14. Shoban Babu Sriramoju, Azmera Chandu Naik, N.Samba Siva Rao (2014). "Predicting The Misusability Of Data From Malicious Insiders" in "International Journal of Computer Engineering and Applications" Vol. V, Issue II, [ISSN : 2321-3469]

15. Ajay Babu Sriramoju, Dr. S. Shoban Babu (2014). "Analysis on Image Compression Using Bit-Plane Separation Method" in "International Journal of Information Technology and Management", Vol. VII, Issue X, [ISSN: 2249-4510]

16. Shoban Babu Sriramoju (2014). "Mining Big Sources Using Efficient Data Mining Algorithms" in "International Journal of Innovative Research in Computer and Communication Engineering" Vol. 2, Issue 1, [ISSN(online): 2320-9801, ISSN(print) : 2320-9798]

17. Ajay Babu Sriramoju, Dr. S. Shoban Babu (2013). "Study of Multiplexing Space and Focal Surfaces and Automultiscopic Displays for Image Processing" in "International Journal of Information Technology and Management" Vol. V, Issue I, August 2013 [ISSN : 2249-4510]

18. Dr. Shoban Babu Sriramoju (2014). "A Review on Processing Big Data" in "International Journal of Innovative Research in Computer and Communication Engineering" Vol.2, Issue-1, January 2014 [ISSN(online) : 2320-9801, ISSN(print) : 2320-9798 ]

19. Shoban Babu Sriramoju, Dr. Atul Kumar (2012). "An Analysis around the study of Distributed Data Mining Method in the Grid Environment : Technique, Algorithms and Services" in "Journal of Advances in Science and Technology" Vol.IV, Issue No-VII, [ISSN : 2230-9659]

20. Shoban Babu Sriramoju, Dr. Atul Kumar (2012). "An Analysis on Effective, Precise and Privacy Preserving Data Mining Association Rules with Partitioning on Distributed Databases" in "International Journal of Information Technology and management" Vol-III, Issue-I, [ISSN : 2249-4510]

21. Shoban Babu Sriramoju, Dr. Atul Kumar (2011). "A Competent Strategy Regarding Relationship of Rule Mining on Distributed Database Algorithm" in "Journal of Advances in Science and Technology" Vol-II, Issue No-II, [ISSN : 2230-9659]

22. Shoban Babu Sriramoju, Dr. Atul Kumar (2011). "Allocated Greater Order Organization of Rule Mining utilizing Information Produced Through Textual facts" in "International Journal of Information Technology and management" Vol-I, Issue-I, [ISSN : 2249-4510]

23. Monelli Ayyavaraiah (2016). "Review of Machine Learning based Sentiment Analysis on Social Web Data" in "International Journal of Innovative Research in Computer and Communication Engineering" Vol. 4, Issue 6, [ISSN(online): 2320-9801, ISSN(print): 2320-9798]

24. Anusha Medavaka, P. Shireesha (2015). "Review on Secure Routing Protocols in MANETs" in "International Journal of Information Technology and Management", Vol. VIII, Issue No. XII, [ISSN: 2249-4510]

25. Anusha Medavaka, P. Shireesha (2016). "Classification Techniques for Improving Efficiency and Effectiveness of Hierarchical Clustering for the Given Data Set" in "International Journal of Information Technology and Management", Vol. X, Issue No. XV, [ISSN : 2249-4510]

26. Anusha Medavaka, P. Shireesha (2016). "Optimal framework to Wireless Rechargeable Sensor Network based Joint Spatial of the Mobile Node" in "Journal of Advances in Science and Technology", Vol. XI, Issue No. XXII, [ISSN : 2230-9659]

27. Anusha Medavaka (2015). "Enhanced Classification Framework on Social Networks" in "Journal of Advances in Science and Technology", Vol. IX, Issue No. XIX, [ISSN : 2230-9659]

28. Anusha Medavaka, Dr. P. Niranjan, P. Shireesha (2015). "USER SPECIFIC SEARCH HISTORIES AND ORGANIZING PROBLEMS" in "International Journal of Advanced Computer Technology (IJACT)", Vol. 3, Issue No. 6, 2015 [ISSN: 2319-7900]

29. Yeshwanth Rao Bhandayker (2016). "Artificial Intelligence and Big Data for Computer Cyber Security Systems" in "Journal of Advances in Science and Technology", Vol. 12, Issue No. 24, [ISSN: 2230-9659].

---

**Corresponding Author**

**Krishna Chaitanya Sanagavarapu***

Student, Masters in Information Systems and Security, University of the Cumberlands, KY