



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY  
Volume 11 Issue 17 Version 1.0 October 2011  
Type: Double Blind Peer Reviewed International Research Journal  
Publisher: Global Journals Inc. (USA)  
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

## Frequent Pattern Mining With Closeness Considerations: Current State Of The Art

By Anurag Choubey, Dr. Ravindra Patel, Dr. J.L. Rana

*Rajiv Gandhi Technological University*

*Abstract* - Due to rising importance in frequent pattern mining in the field of data mining research, tremendous progress has been observed in fields ranging from frequent itemset mining in transaction databases to numerous research frontiers. An elaborative note on current condition in frequent pattern mining and potential research directions is discussed in this article. It's a strong belief that with considerably increasing research in frequent pattern mining in data analysis, it will provide a strong foundation for data mining methodologies and its applications which might prove a milestone in data mining applications in mere future.

*GJCST-C Classification : H.2.8*



*Strictly as per the compliance and regulations of:*



# Frequent Pattern Mining With Closeness Considerations: Current State Of The Art

Anurag Choubey<sup>α</sup>, Dr. Ravindra Patel<sup>Ω</sup>, Dr. J.L. Rana<sup>β</sup>

**Abstract** - Due to rising importance in frequent pattern mining in the field of data mining research, tremendous progress has been observed in fields ranging from frequent itemset mining in transaction databases to numerous research frontiers. An elaborative note on current condition in frequent pattern mining and potential research directions is discussed in this article. It's a strong belief that with considerably increasing research in frequent pattern mining in data analysis, it will provide a strong foundation for data mining methodologies and its applications which might prove a milestone in data mining applications in mere future.

## I. INTRODUCTION

Frequent patterns are itemsets, subsequences, or substructures which are often found as a repetitive pattern in any database, with rate of recurrence depending upon individual verge. For example: Itemsets are sets of item such as vegetables and spices that are often observed in a transaction data. Subsequence is a chronological pattern habitually followed, such as buying a camera, then a camera carry-bag and then lenses. Substructure denotes particular structural forms which may be pooled with data of itemsets or subsequences. Regular occurrences of a substructure in a data-base is called structural pattern. For enhancement in results for data indexing, classification, clustering and other data mining tasks, discovery of frequent pattern mining has turned out to be an essential data mining task.

Agarwal et al. (1993) was the pioneer, proposing frequent pattern mining for market basket analysis in form of association rule mining which examines customer trading behavior by ruling out relations between dissimilar items that customers buy. These statistics facilitate a retailer to be careful regarding further marketing. This relevance provided a podium for various other research publications on various kinds of extensions and applications, and hence it's time to examine what more needs to be inspected to make this expertise a keystone approach for all supplementary demanding research issues.

## II. MINING PATTERNS WITH CLOSENESS AS CONSIDERATION

With large data, mining generates colossal frequent patterns satisfying  $\text{min\_sup}$  threshold,

principally when  $\text{min\_sup}$  is set low, because with frequent patterns comes along frequent sub patterns too. Hence the principal of closed frequent pattern mining and maximal frequent pattern mining came to subsistence.

The mining of frequent closed itemsets was proposed in [2], where an Apriori-based algorithm called A-Close for such mining was presented. Other closed pattern mining algorithms include [3, 4, 5, 6, and 7]. Pattern closeness identification is the big obstacle in closed or maximal pattern mining. The usage of hashed transaction ids is on e significant approach to overcome this complication [4] and the other approach is to construct hierarchical tree structure using discovered patterns as nodes [5, 7, and 8]. Mining closed itemsets provides an interesting and important alternative to mining frequent itemsets since it inherits the same analytical power but generates a much smaller set of results. Better scalability and interpretability is achieved with closed itemset mining.

Bayardo et al [9] proposed level-wise, breadth-first search method to find max-itemset. This approach includes subspace reduction by pruning process that prunes frequency of superset and infrequency of subset for search space reduction. Transaction id list compression approach was discussed in another efficient method MAFIA [10] that targets the counting efficiency. NP-Hard is the complexity of reciting maximal itemsets.

## III. MINING PATTERNS WITH SEQUENCE AS CONSIDERATION

Sequential pattern mining, the mining of frequently occurring ordered events or subsequences as patterns in set of ordered elements or events that referred as sequence dataset, was first bring out by Agarwal et al[11], and has become an important problem in data mining. We first introduce the preliminary concept about sequential patterns.

Generalized Sequential Patterns (GSP) [12] is opt to multiple pass approach and uses the down the level closeness of patterns, and other consideration is the candidature maintenance approach SPADE [13] proposed by Zaki et al is a sequential pattern mining model that works based on vertical format, which is an extension to Eclat[14] and CHARM[4]. The use of vertical data format minimizes the passes through dataset. Since the search process relies on breadth first

Author <sup>α,Ω,β</sup> : Rajiv Gandhi Technological University, Bhopal, India  
E-mail : <sup>α</sup> anuragphd11@gmail.com

and pruning is apriori based, which causes complex candidature maintenance in order to manage longer sequences.

Prefix Span[15, 8] was developed by Pei et al that works in a divide-and-conquer way. In this model each sequential pattern considers as prefix and projects the complete set of sequential patterns and later partitioned into different subsets according to different prefixes. Finally projected datasets are mined recursively to find sub sequent patterns. Empirical study concludes that the prefixspan[15] is the best when compared to GSP and SPADE. And the study also shows that PrefixSpan has the best overall performance. The CloSpan[16] was proposed by Yan et al. The method is based on a property of called equivalence of projected databases, where two projected sequence datasets with same prefix and same size in total number of items, this model can prune the non-closed sequences from further consideration during the mining process. BIDE[17], a bidirectional search for mining frequent closed sequences was developed by Wang and Han et al, which can further perform projecting sequences in bidirectional.

#### IV. MINING FREQUENT ITEMSETS AND NO CANDIDATURE MAINTENANCE

The Apriori principle is significant in reducing the size of candidate sets. But multiple passes through dataset and maintaining large number of candidate sets are two major obstacles of the apriori principle. In the context of these obstacles Han et al[18] proposed a model called FP-growth that mines frequent itemsets without using candidature maintenance.

FP-growth works by compressing the given dataset by ordering items as extended prefix-tree structure. FP-growth targets the decomposition of given dataset and mining activities, in this process the FP-growth endorsed by divide-and-conquer approach. It uses an incremental pattern fragmentation to eliminate the candidature maintenance (generation and testing), which is used by Apriori.

Empirical studies in literature demonstrate that fp-growth considerably minimizes the search time. The FP-growth algorithm performs well by searching for shorter frequent patterns recursively and then concatenates least frequent items as suffix. There are suitable alternative and extensions too for this approach, few of them are Agarwal et al[19], H-mine[20] Liu et al. [21, 22] and Grahne [6].

#### V. CURRENT STATE OF ART

Authors Ya-Han Hu, Fan Wu, and Tzu-Wei-Yeh [23] have presented an in-detail thesis about Market based Analysis. This dat helps in finding correlation between purchasing items in databases. This study extends RFM analysis into mining process which

measures frequent patterns extending in mining. Initially a RFPM tree is prepared depending upon the RFM-patterns, to compress and store entire transactional database and later RFPM growth is developed, hence efficiently find RFM patterns.

Association rule mining, usefull for discovering relationships among items or events in various app domains, discovers complete set of frequent patterns. Here, instead of filtering patterns after discovery, constraints (specific algorithms possessing meaningfull info) are directly pushed in into process of discovering and hence desired patterns are discovered.

Constraints from RFM analysis examine uniqueness from frequent patterns into mining algorithm. A RFM pattern is one in which recency score, frequency score and monetary score which satisfies minimum thresholds. Wu et al. discussed the recency problem about the change of data distribution between the past data and the new data. Emerging patterns as introduced by Dong and Li discover significant pattern changes in datasets with different time periods. In mining process, it cant reflect requirement of RFM completely. RFM-Apriori defined the recency constraint of the last transaction that should satisfy the recency threshold. The paper proposed an algorithm for mining patterns satisfying RFM constraints using a fixed time gap. However their approach the different stripe lengths of recency measuring the importance of transaction. This method better interfaced and also measured RFM importance of patterns. . For a set  $I$  of items in a database, to measure the scores of RFM of itemsets, the database should contain the more information of a transaction  $X$ , like  $\langle tx, (a_1, qa_1) (2 a_2, qa_2) \dots (a_m, qa_m) \rangle$ , where  $tx$  stands for the transaction time when transaction  $X$  occurs, the  $m$  pairs,  $(a_i, qa_i)$ , where  $1 \leq i \leq m$  and  $a_i \in I$ , denotes that there are  $m$  different items are purchased in the transaction and for item  $a_i$ , the purchased quantity is  $qa_i$ . Given an itemset  $Y$ ,  $Y$  is said to be contained in  $X$  if all items in  $Y$  also occur in  $X$ . Definition 1. Assume that an itemset  $Y$  is contained in transaction  $X$ .  $Y$ 's transaction recency score gained from  $X$ , denoted by  $trscore(Y, X)$ , is defined as:  $trscore(Y, X) = (1-\delta) t_{current-tx}$ , where  $\delta$  is a user-specified decay speed ( $\delta \in [0, 1]$ ),  $t_{current}$  denotes the current timestamp. Definition 2. Following Definition 1, the recency score of itemset  $Y$ , denoted by  $Rscore(Y)$ , is defined as the sum of transaction recency score gained from all of the transactions containing it. Definition 3. The frequency score of itemset  $Y$  in  $DB$ , denoted by  $Fscore(y)$ , is the number of transactions containing itemset  $Y$ . Definition 4. Let  $p(a_i)$  denotes the unit price of item  $a_i$ . Assume that an itemset  $Y$  is contained in transaction  $X$ . Definition 5. Following Definition 4, the monetary score of itemset  $Y$  in  $DB$ , denoted by  $MscoreDB(Y)$ , is the sum of monetary score gained from transactions containing itemset  $Y$ . Definition 5. Following the Definition 4, the transaction amount for

transaction  $X$ , denoted by  $ta(X)$ , is the sum of all items' monetary score in transaction  $X$ s.

An RFMP-tree contains RFM-header and RFMP-tree. A RFM-header contains all 1-RFT-patterns. Each node in RFMP-tree consist item-name, Rscore, Fscore, Tta, parent-link, child-link, and sibling-link as fields, where item-name registers. A complete RFMP-tree will be constructed through two database scans, in the first DB scan, all 1-RFT-patterns will be stored in the RFMheader to pool up. After that, all 1-RFT-patterns in RFM-header will be arranged by thier Fscore in descending order. In other DB scan the root node of the tree will be sorted and label it as null, is created and prunes items for every transaction if item not found in RFM-Header. In the first step it finds all RFT-patterns from the RFMP-tree. The second step scans database once to compute the Mscore of all RFT-patterns and finally discover all RFM-patterns. Inserting of transactions in RFMP-tree generates 2 possible cases. Due to branch in the transaction, scores related to transactions are updated to nodes. If no branches are observed, nodes will have to be developed and have to calculate scores for it. With new nodes, comes along creation of parent link and node link.

**Observation:** Three of the thresholds performs outstandingly with experiments depicting significant values which are not generalized. In my observations values should be generalized and experiments on datasets should entertain noise and uncertainty.

A deisgn determining interesting pattern, minimum support & itemset count is designed by ZalifahAwang Long et al[24]. 1.Pre-processing 2.frequent item set algorithm and 3.data re-transformation are 3 stages in it which involves 2 modules of pre-processing. The Apriori algorithm is constructed in second module. In 2<sup>nd</sup> stage minimum support and sequence length 'k' is found. In third stage, output is found which is threshold in outbreak of detection task.

Experiments were conducted in 3 datasets ZOO, TIC TAC TOE (UCI data repositories) and third for WSARE implementation. Results concluded that less the support more interesting pattern observed. Depending on the results, range for minimum support is 30%-50%. Figure below depicts support applied while experiment ranged from 10%-100% graphically. Positions of high gain are detected based on mean running collected output. The graph is analyzed state wise.Hence 40%-50% of minimum support is sufficient to generate frequent itemsets. High frequency count determines number of item sites for max detection rate. This paper is so designed to detect prospective interesting pattern from sequence item sets count and hence outbreaks in pattern are identified.

**Observation:** Exact number of min. support and length of item set reduces false rate of outbreak detection. Approx values are considerate and

investigations on detection of curve are obligatory. Based upon domain driver framework, a new approach to mine pattern among data stored as schema was proposed by CláudiaAntunes et al [25], which is performed by a D2Apriori algorithm. Problems on Onto4ARframework are centered using ontology and assumes varied problem in context of ontology. The transactions provided in transactional patterns are by knowledge base and since its tuple with ontology characterization by set of features, it corresponds to traditional items. By adapted transactional pattern mining algorithm, new context mining can be achieved, to avail knowledge represented by ontology. The applications are executed on Apriori-based algorithm-D2Apriori with described procedures performed in a two step process as:-

1. Receives the dataset as input followed by reception of knowledge base and the constraints applicable.
2. Data file is then read in the context of the knowledge base, creating the dataset with the incorporation of semantics for each read transaction. As a special feature for reduction of time spent in the discovery process, measures adopted by authors are:

- Read the set of concepts and its attributes.
- Propagate concept properties through the taxonomy.

Hence problem for pattern mining and discovery of info in data stored by framework Onto4AR is resolved by overcoming results through traditional pattern approached.

**Observation:** D2Apriori algorithm is inefficient although due to memory consumption.

Ya-Han Hu et al[26] have highlighted study on Sequential Pattern Frequent Mining with multiple minimum supports. This determines correlations among items or itemsets which ows to varied frequencies, specyfyng single minimum support cannot accurately discover interesting patterns. Based on these solutions authors argued that they are not straight forward. So an extended version of PLWAP-tree[28](Preorder linked multiple support tree) structure has been proposed. An efficient algorithm, a extended version of PLWAP-tree, named MSCP-growth(multiple supports candidate pattern) is used for detecting set of sequential patterns with MMS.

PLMS-tree obtained from PLWAP tree-like algorithm backups necessary information in sequence database. later, MSCP-growth develops sequential pattern with MMS based on PLMS-tree. Only items with less support than minsup are contained in PLWAP-tree while in PLMS-tree not less than MIS(I). in PLMS-tree extra info is required to tree structure but not required for the later part.

**Observation:** In this argument minute inconveniences are detailed. The concept of MMS on sequential pattern generates interesting patterns and an algorithm named MSCP growth is described. With

advantages of several algorithms combined, And experimental results exhibiting better performances than Compressed and Arranged Transaction Sequences tree(CATS tree) and Grouping Compressed tree mining algorithm, Chuang-Kai et al [27] developed a new structure called Sorted Compression tree as mining algorithm for association rule. In order to overcome repeated mining due to several minimum support, many algorithms were proposed, of which CATS tree is one of them. It was ruled out due to drawbacks and better algorithm called Sorted compression tree(SC tree) was brought in, which was efficient and less memory consumptive.

Knowledge is derived from statistical data and relationship among products. The later is obtained from "Association Rule Mining" which is usually used. Apriori algorithms and Frequent patterns growth algorithm must use CATS-tree to adjust minimum support value. Its effectiveness is improved by usage of SC-tree, which by pre-sorting dat-set SC-tree results can be reliable.

A. Frequent Pattern Growth Algorithm: The FP growth algorithm proposed by Han and et al. is the most typical algorithm which is employed. The technique avoids the scanning of the database frequently, avoiding generating candidate item sets and reducing the hunt space. It is resourceful, scalable and faster than Apriori algorithm.

B. Compressed and Arranged Transaction Sequences Tree Algorithm: In CATS tree algorithm, there is no need for reconstruction when the minimum support is altered and it preserves all elements without decreasing. When minimum support threshold is distorted, the tree structure is unnecessary to be reconstructed. CATS algorithm constructs without sorting and hence, a need to adjust the tree structure.

C. Grouping Compress Tree Algorithm: The GC-tree increases performance of CATS-tree by pre-processing and compressing data into GC data structure and hence looks out large item set depending upon complexity of data.

D. Sorted Compress Tree Algorithm: This algorithm is used for tree construction and rule mining method. It consumes less memory space by the usage of SC-tree in large databases. SC algorithm mines frequent items in 3 stages.

1. Data processing- It sorts and arranges transactions for consistency which in turn help in improving tree constructing and rule mining.
2. SC-tree constructing –SC-tree is constructed by sorting transaction database.
3. Rule Mining in SC-tree- initially nodes are sorted, so as to mine in single direction i.e bottom-up. This process builds conditional mining tree and generates frequent item set.

**Observation:** CATS tree and GC-tree are verified under observations of SC-tree. Focus is on execution

and memory consumption in minimum support and transaction size. Under execution and memory requirements spotlight is on efficiency on tree construction and association rule mining. "Fast and memory efficient mining of frequent closed item sets"[29] was authored by Claudio Lucchese et al. It mainly cogitate an ascending algorithm to provide a distilled approach to represent transactional often item sets which are exploited from the database. For the purpose of exploiting the often item sets, it preferred the techniques such as "divide-and-conquer" and "bitwise vertical representation" of the database. To achieve this approach, they utilized the services of separation technique of the "search space". It relies on authentic theoretical model for defining the often item sets clearly. As this algorithm faces the job of managing production of similar item set, it needs to keep an account of concise the space and time complexity for calculating item sets. To satisfy this problem, they opted for an efficient and capable trimming methodology. This methodology has the advantageous edge that it need not place all the fetched data in the main memory. It also has the facility to fetch data from each divided part of search space autonomously. The fetching operation can be pipelined and also it can be done in any sequence. Various examinations were performed over the proposed algorithm mainly commonly accessible data sets which proved this algorithm as an ascendable one which is portable for algorithms as CHARM, CLOSET+ AND FP-CLOSE even for multiple ordered applications. The efficient increments are considered more important in the case of decrement of support threshold. Here, a new walkthrough across the search space which is specific for every item set is proposed. It relies on the idea that there exist producers to maintain available pattern of the often item sets. This technique produces profitable results in the form of enabling us to create autonomous subtasks which can be pipelined.

**Observation:** The methodology proposed in this paper is practically applied to "DCI\_CLOSED". This approach adopts techniques such as "depth first traversal" and "vertical bitmap representation" of dataset. This approach satisfies all the specifications proposed in the new technique of the paper. Due to the implementation of these techniques and methodologies, the proposed algorithm has limited space and time complexity. From the conducted examinations, it can be drawn that "DCI\_CLOSED" is ascendable and supports fetching of heavy data sets even with minimum support threshold.

Liping Jiet al[30] proposed a clode pattern detection model called "Compressed Hierarchical Mining of Frequent Closed Patterns from Dense Data Sets". The common technique followed here in all algorithms is that result will be the same whether the data is entirely mined or individually mined. Here, data is

partitioned in to several blocks and mining algorithm is applied on each individual block and then combined finally. Another advantage is that this process can be done in parallel, saving time and effort with every performance measures rating good. Both the algorithms can compress the mining space and can divide data hierarchically to apply mining. The only difference between the two algorithms is the technique they follow. C-mining follows Compact-Mining technique and B-mining follows Base rows Projection. These two versions can be done in parallel successfully and is shown in this proposal. We do have many other data mining techniques like PB-mining, PC-mining, D-mining along with C and B mining algorithms.

**Observation:** The experimental results can be studied under several results by Varying data set density, Experiments on real micro data set, varying the number of processors and scalability. These show the success of parallelism in the implementation of both C and B mining techniques.

An Efficient Algorithm for Mining Top-K Frequent Closed Itemsets was proposed by Jianyong Wang et al[31]. Authentic way of data set fetching includes considering “minimum support” threshold. It may not be feasible to specify “minimum support” everytime. Moreover, the conventional way may demand to fetch whole set of often data sets obeying the threshold. These specifications may restrict its availability as it is hard to avail “minimum support” threshold according to the situation. Moreover, “closed” item sets are compressed than a whole set of often data sets, making it portable and more applicable. Thus, to avoid such limitations, a broader methodology called “mining top-k closed item sets” is proposed. The technique followed here is that the minimum length should be “min\_l”. Here, “k” represents the required count of item sets to be fetched and “min\_l” stands for the minimum length to be considered. This methodology, TFP, is constructed avoiding the need of “minimum support” threshold. The methodology initializes with the “minimum support” threshold value as 0. This value along with characteristics of top-k often data sets and length limitations, we increment the value of “minimum support” gradually. Also, utilize of “node count” and “descendant sum” techniques, allows the active trimming of FP-tree even prior or posterior to its construction. Even the fetching process can be fasten with the utilization of FP-tree including “top-down” and “bottom-up” traversals, applying heap of “search space” trimming techniques, high speed 2-stepped “hash-indexed result tree” and a new item set closure checking technique. Various examining processes proved that the proposed methodology was more efficient and linear ascendable in case of database space. The methodologies of proposed approach are: incrementing “minimum support” threshold utilizing “closed node count array” and “descendant sum”, fetching only required elements initially and leaving rest

for the fetch during “FP-tree” fetching, fastening fetching process by embedding “search space” trimming techniques and engaging “closure verification scheme” for packing up item sets.

**Observation:** This algorithm can be broadened to produce association principles and to embed specified limitations. From the analysis over authentic mining, the proposed algorithm is desirable as it avoids the need for specifying “minimum support” threshold. Even many reworks have been started over various elements such as top-k often closed item sets’ efficiency, tractability, environment and various sequences.

Max-Clique[32] is a top-down Graph-based Approach to Frequent Pattern Mining was proposed by Yan Xie et al. The actual working process here is unlike in traditional methods where entire data is divided into parts and each is applied algorithm and then combined. These traditional approaches can be named Bottom-up since for mining entire data each detail is to be considered which is possible only in bottom-up process. The new strategies are that the data is processed from top to down and the most frequent top K long maximal patterns are identified and are resolved. Here the other small data sets are ignored and concentrated on those large sets. The large frequent patterns identified are called Cliques technically. These cliques are matched with some predefined pattern groups. Then apply Maximal Clique Detection on such cliques. Identifying the cliques is followed by Refinement phase where cliques are transformed into true maximal patterns.

**Observation:** This proposed model let mine only those long and interesting patterns which improve the speed and quality of the product with out negative effects on growth of other patterns which may be considered small. In fact the idea behind and the challenging factor is the identification of those large patterns to be mapped to cliques and is done just opposite to traditional bottom-up process as top-down.

## VI. CONCLUSION

The analysis of the pattern and itemset mining algorithms should target performance, scalability, flexibility and reusability for making generation of association rules in robust way. Based on the reviews drafted in this paper, we can conclude that the frequent pattern mining is a price high process. All of the algorithms are worthy to achieve required goals in terms of results. But much of them influenced by an obstacle called multiple pass through given dataset that requires much number of disk reads. The rest algorithms those are able to perform mining with single passes are not portray the importance of these patterns. Further research should aim the concerns like flexibility, reusability and prunes in no. of derived patterns without compromising at pattern quality. These requirements influence to gain interest in further research in Frequent Itemset and pattern mining.

## REFERENCES RÉFÉRENCES REFERENCIAS

1. Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proceedings of the 1993ACM-SIGMODinternational conference on management of data (SIGMOD'93), Washington, DC, pp 207–216.
2. Pasquier N, Bastide Y, Taouil R, Lakhal L (1999) Discovering frequent closed itemsets for association rules. In: Proceeding of the 7th international conference on database theory (ICDT'99), Jerusalem, Israel, pp 398–416.
3. Pei J, Han J, Mao R (2000) CLOSET: an efficient algorithm for mining frequent closed itemsets. In: Proceeding of the 2000 ACM-SIGMOD international workshop data mining and knowledge discovery (DMKD'00), Dallas, TX, pp 11–20.
4. Zaki MJ, Hsiao CJ (2002) CHARM: an efficient algorithm for closed itemset mining. In: Proceeding of the 2002SIAMinternational conference on data mining (SDM'02), Arlington, VA, pp 457–473.
5. Wang J, Han J, Pei J (2003) CLOSET+: searching for the best strategies for mining frequent closed itemsets. In: Proceeding of the 2003 ACM SIGKDD international conference on knowledge discovery and data mining (KDD'03), Washington, DC, pp 236–245.
6. Grahne G, Zhu J (2003) Efficiently using prefix-trees in mining frequent itemsets. In: Proceeding of the ICDM'03 international workshop on frequent itemset mining implementations (FIMI'03), Melbourne, FL, pp 123–132.
7. Liu G, Lu H, Lou W, Yu JX (2003) On computing, storing and querying frequent patterns. In: Proceeding of the 2003 ACM SIGKDD international conference on knowledge discovery and data mining (KDD'03), Washington, DC, pp 607–612.
8. Pei J, Han J, Mortazavi-Asl B, Wang J, Pinto H, Chen Q, Dayal U, Hsu M-C (2004) Mining sequential patterns by pattern-growth: the prefixspan approach. *IEEE Trans Knowl Data Eng* 16:1424–1440
9. Bayardo RJ (1998) Efficiently mining long patterns from databases. In: Proceeding of the 1998 ACM-SIGMOD international conference on management of data (SIGMOD'98), Seattle, WA, pp 85–93.
10. Burdick D, Calimlim M, Gehrke J (2001) MAFIA: a maximal frequent itemset algorithm for transactional databases. In: Proceeding of the 2001 international conference on data engineering (ICDE'01), Heidelberg, Germany, pp 443–452.
11. Agrawal R, Srikant R (1995) Mining sequential patterns. In: Proceedings of the 1995 international conference on data engineering (ICDE'95), Taipei, Taiwan, pp 3–14.
12. Srikant R, Agrawal R (1996) Mining sequential patterns: generalizations and performance improvements. In: Proceeding of the 5th international conference on extending database technology (EDBT'96), Avignon, France, pp 3–17.
13. Zaki M (2001) SPADE: an efficient algorithm for mining frequent sequences. *Mach Learn* 40:31–60.
14. Zaki MJ (1998) Efficient enumeration of frequent sequences. In: Proceeding of the 7th international conference on information and knowledge management (CIKM'98), Washington, DC, pp 68–75.
15. Pei J, Han J, Mortazavi-Asl B, Pinto H, Chen Q, Dayal U, Hsu M-C (2001) PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. In: Proceeding of the 2001 international conference on data engineering (ICDE'01), Heidelberg, Germany, pp 215–224.
16. Yan X, Han J, Afshar R (2003) CloSpan: mining closed sequential patterns in large datasets. In: Proceeding of the 2003 SIAM international conference on data mining (SDM'03), San Francisco, CA, pp 166–177.
17. Wang J, Han J (2004) BIDE: Efficient mining of frequent closed sequences. In: Proceeding of the 2004 international conference on data engineering (ICDE'04), Boston, MA, pp 79–90.
18. Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. In: Proceeding of the 2000 ACM-SIGMOD international conference on management of data (SIGMOD'00), Dallas, TX, pp 1–12.
19. Agarwal R, Aggarwal CC, Prasad VV (2001) A tree projection algorithm for generation of frequent itemsets. *J Parallel Distribut Comput* 61:350–371.
20. Pei J, Han J, Mortazavi-Asl B, Pinto H, Chen Q, Dayal U, Hsu M-C (2001) PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. In: Proceeding of the 2001 international conference on data engineering (ICDE'01), Heidelberg, Germany, pp 215–224.
21. Liu J, Pan Y, Wang K, Han J (2002) Mining frequent item sets by opportunistic projection. In: Proceeding of the 2002 ACM SIGKDD international conference on knowledge discovery in databases (KDD'02), Edmonton, Canada, pp 239–248.
22. Liu G, Lu H, Lou W, Yu JX (2003) On computing, storing and querying frequent patterns. In: Proceeding of the 2003 ACM SIGKDD international conference on knowledge discovery and data mining (KDD'03), Washington, DC, pp 607–612.
23. Ya-Han Hu; Fan Wu; Tzu-Wei Yen "Considering RFM-values of frequent patterns in transactional databases", 2nd International Conference on Software Engineering and Data Mining (SEDM), June 2010, pages: 422 - 427.
24. Long, Z.A. Hamdan, A.R. Bakar, A.A; Parameter setting procedure via quick parameter evaluation in

- frequent pattern mining for outbreak detection, 2nd Conference on Data Mining and Optimization, 2009. DMO '09, Issue Date: 27-28 Oct. 2009, On page(s): 90 - 93.
25. Antunes, C.; Pattern Mining over Star Schemas in the Onto4AR Framework, IEEE International Conference on Data Mining Workshops, 2009, ICDMW '09, Issue Date: 6-6 Dec. 2009, On page(s): 453 - 458.
  26. Ya-Han Hu; Fan Wu; Yi-Chun Liao; Sequential pattern mining with multiple minimum supports: A tree based approach, 2nd International Conference on Software Engineering and Data Mining (SEDM), Issue Date: 23-25 June 2010 On page(s): 428 – 433.
  27. Chuang-Kai Chiou, Judy C. R. Tseng; Sorted Compressed Tree: An Improve Method of Frequent Patterns Mining without Support Constraint, 2nd International Conference on Software Engineering and Data Mining (SEDM), 2010, Issue Date: 23-25 June 2010, On page(s): 328 - 333.
  28. Ezeife, C.I.; Min Chen; Incremental mining of Web sequential patterns using PLWAP tree on tolerance MinSupport, Database Engineering and Applications Symposium, 2004, Issue Date: 7-9 July 2004, On page(s): 465 – 469.





This page is intentionally left blank