A COMPARATIVE STUDY OF SEQUENTIAL PATTERN MINING ALGORITHMS

Jawahar. S

Assistant Professor, RVS College Of Arts & Science, Coimbatore

Abstract

The concept of sequence Data Mining was first introduced by Rakesh Agrawal and Ramakrishnan Srikanth in the year 1995. Sequential pattern mining are used in broad applications. It is used to uncover all sequential pattern which satisfy the user specified constraint, from the given sequence database. However, it is also a difficult to generate and examine a number of intermediate sub-sequences. Sequential pattern mining is the method of finding interesting sequential patterns among the large database. The application can be used in various domain like natural disaster, sales record analysis, marketing strategy, shopping sequences, medical treatment and DNA sequences etc. This paper presents the traditional sequential pattern mining algorithms. The sequential pattern mining algorithms are classified into four classes: First, on the basis of Apriori-based algorithm, Second on Breadth First search-strategy, third on Depth First search-strategy and fourth, on Sequential closed-pattern algorithm. The surveys of all these algorithms are made on the basis on of various research perspectives. First algorithms are categorized using various approaches to solve the mining problem and then algorithms are compared each one with another by their various provided features and performance point of view.

Keywords: Sequential Pattern Mining, Apriori, Breadth First search, Depth First Search, Sequential closed patterns.

1. INTRODUCTION

Data Mining is the discipline of finding novel remarkable patterns and relationships in vast quantity of data. Data mining technique is not developed only for a particular industry. Data Mining is considered to be very important for almost all software based applications [21]. It consists of effective techniques which help to bring out the hidden knowledge in huge volume of data. The major issue of data mining in the recent years was focused on mining sequential patterns in a set of data sequence. The major assignment of sequential pattern mining is to determine the complete set of sequential patterns in a given sequence database with minimum user defined minimum support. Given a set of sequences and the user-specified minimum support threshold, the sequential pattern mining finds all frequent sub-sequences that identifies the sub-sequences whose occurrence frequency in the set of sequences is not less than minimum_support threshold [2].

Sequential pattern mining was first introduced by Agrawal and Srikant [2]. The sequential pattern mining is a very important concept of data mining, a further extension to the concept is association rule mining [3], which has a wide range of real-life application. This sequential pattern mining algorithm solves the problem of discovering the presence of frequent sequences in the given database [2]. Sequential pattern mining is the method of finding interesting sequential patterns among the large databases. It also finds out frequent sub-sequences as patterns from a sequence database. Large amount of data are being collected and stored in many industries and they are interested in finding the sequential patterns from their database. Sequential pattern mining have many applications including, the analysis of customer purchase patterns, web-access patterns, disease treatments, finding telephone calling patterns, analysis of DNA sequences etc [4,5,6].

A. Basic Concepts of Sequential Pattern Mining [7]

1. Let *I* = {*x*1, . . . , *xn*} be a set of *items*, each perhaps being associated with a set of *attributes*, such as value, price, profit, calling distance, period, etc. The value on an attribute *A* of item *x* is denoted by *x.A*. An *itemset* is a non-empty subset of items, and an itemset with k iems is called k itemset.
2. A *sequence*=<*X*· · *α* · *X* > is an ordered list of itemsets. An itemset *Xi (*1 *≤i ≤l)* in a sequence is called as a *transaction*, a term emerged from shopping sequences in a transaction database. A transaction *Xi* may have an exceptional attribute, *time-stamp*,denoted by *Xi.time*, which registers the time when the transaction was executed. For a sequence *α*= <*X*1 · · · *Xl*>, we assume that *Xi.time* < *Xj.time* for 1 *≤i* < *j ≤l*.
3. The number of transaction in a sequence is called as *the length of sequence*. A sequence with length *l* is called *as* an *l-sequence*. For an *l*-sequence *α*, we have *len (α)=l*. Moreover, the *i*-th itemset is denoted by *α*[*i*]. An item can arise at most once in an itemset, but can arise multiple times in various itemsets in a sequence.
4. A sequence *α*= <*X*1 . . . *Xn*> is called as a *subsequence* of another sequence *β*= <*Y*1 . . .*Ym*> *(n ≤ m),),* and *β*a *super-sequence* of *α*, if there exist integers 1 *≤i*1 <..< *in ≤m* such that *X*1 *Yi*1,.., *Xn Yin*.
5. A *sequence database SDB* is a set of 2-tuples *(sid, α)* where *sid* is a *sequence-id* and *α is* a sequence. A tuple *(sid, α)* in a sequence database SDB *is said to hold* sequence *γ* if *γ* is a subsequence of *α*. The number of tuples in a

sequence database *SDB* holding sequence γ is called the *support* of γ, denoted by *sup* (γ). Given a positive integer *min_sup* as the *support threshold*, a sequence γ is a *sequential pattern* in sequence database *SDB* if $sup \geq min(\gamma) sup$. The *sequential pattern mining* problem is to discover the *complete* set of sequential patterns with respect to a given sequence database *SDB* and a support threshold *min_sup.*

2. CATEGORIES OF SEQUENTIAL PATTERN MINING ALGORITHM

As described by Yen-Liang Chen and Ya-Han Hu [9] in recent years, many approaches in sequential pattern mining have been proposed, these studies cover a broad spectrum of issues. In general, there are two main research issues in sequential pattern mining.

1. **Improve the efficiency of the mining process [7]**. This strategy focus on improving the efficiency in sequential pattern mining process.

2. **Extend the mining of sequential patterns to other time-related patterns [8]**. This strategy focuses on finding other patterns in time-related databases such as finding frequent patterns in a web log, cyclic patterns in a time-stamped transaction database etc.

Based on these criteria's sequential pattern mining can be divided broadly into four parts:

- Apriori-Based algorithm
- Breadth First-based strategy
- Depth First-based strategy
- Sequential closed-pattern

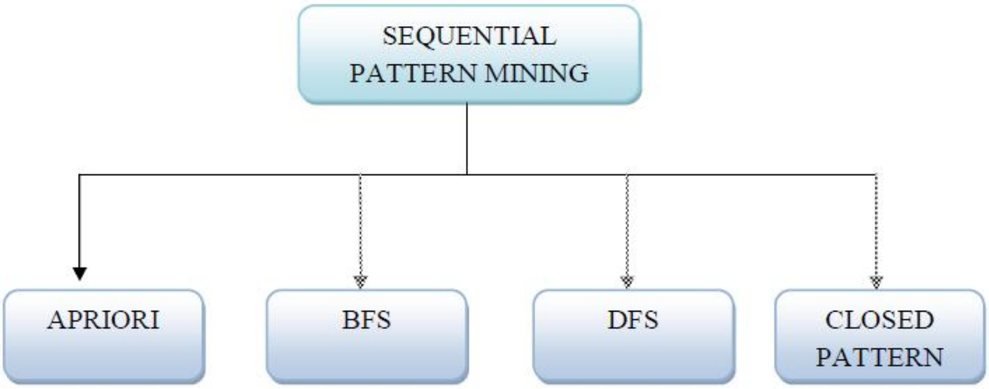


Figure 1: Classification of sequential pattern mining algorithms

2.1 Apriori-Based algorithm

The first introduction of classical Apriori-based sequential pattern mining algorithms was in the year 1995. In apriori algorithms mimimum-support is specified by users on the basis of assumption. It is impossible that users give a suitable minimum-support for a database to be mined if the users are without knowledge concerning the databases. For this reason fuzzy mining strategy with database-independent minimum-support can be used, which provides a good machine interface that allows users to specify the minimum-support threshold without any knowledge concerning their databases to be mined. Apriori is used in transaction database which includes customer sequences. This database is contains three attributes (customer-id, transaction-time and purchased-item).

2.1.1 Apriori All

Sequential pattern mining was first proposed in [2]. There are five steps in the AprioriAll algorithm - sort phase, item sets phase, transformation phase, sequence phase and maximal phase [2,21]. In sort phase the database is sorted with customer-id as the major key and transaction time as the minor key and after that the database is converted into the sequence database. In the item sets phase the database is scanned to obtain frequent 1-sequences and also large 1-itemset. The set of item sets are mapped to a set of contiguous integers. In transformation phase the sequence database is transformed into set of sequence database by those large item sets. Now if the sequence database does not contain any large item sets, then that sequence is not carry in the transformed sequence. In the sequence phase multi-passes are performed over the database to found the sequential pattern [21].

In the AprioriAll algorithm the frequent 1-sequences are identified from the transformed database and this frequent1-sequence becomes the seed set for finding the frequent 2-sequences and the process will carryout up to the point when no more frequent sequences generation are possible. After the sequence phase lots of frequent sequences are generated, so to reduce this huge number of frequent sequences are used [2,21]. The sequences are called maximal sequences if s is not contained in any other sequence [2]. One drawback of AprioriAll algorithm is that it generates lots of candidate sequences so it is very time consuming process to prune the candidate sequences [21].

2.1.2 SPIRIT

SPIRIT (Garofalakis, 1999) algorithms are a sequential pattern mining with regular expression constraints. SPIRIT algorithm uses regular expressions as flexible constraint specification tool. On the mined patterns a generic user-specified constraint is involved which is considerably versatile and powerful restrictions are forced inside the mining process. But this algorithm uses less restrictive and more relaxed version of constraint. Many algorithms are available but each will differ in the degree to which the constraints are used to prune the search space while discovering patterns.

2.2 BFS-based algorithms

Breath-first (level-wise) search algorithms describe the Apriori-based algorithms because all *k*-sequences are constructed together in each *k*th iteration of the algorithm as they traverse the search space. Several algorithms are developed using the principle of BFS algorithms. Some of them are illustrated in the following sections:

2.2.1 GSP algorithm

The GSP algorithm proposed in [15], is same as AprioriAll algorithm, but it doesn't require finding all the frequent item sets first. This algorithm allows a) placing bounds on the time separation between adjacent elements in a pattern, b) allowing the items included in the pattern element to span a transaction set within a time window specified by user, c) permitting the pattern discovery in different level of a taxonomy defined by user. Additionally, GSP is designed for discovering generalized sequential patterns. The GSP algorithm makes multiple passes over sequence database as follows: 1) in the first pass, it finds the frequent sequences that have the minimum support. 2) At each pass, every data sequence is examined in order to update the occurrence number of the candidates contained in this sequence.

2.2.2 MFS algorithm

It is a modified version of GSP, proposed [16] with the aim to reduce the I/O cost needed by GSP. MFS computes as a first step the rough estimate of all the frequent sequences set as a suggested frequent sequence set and to maintain the set of maximal frequent sequences known previously it uses the candidate generation function of GSP. The results obtained in [16] show that MFS saves I/O cost significantly in comparison with GSP.

2.3 DFS-based algorithms

The algorithms adopting this feature show only an ineffective pruning method and engender a great number of candidate sequences, which requires consuming a lot of memory in the early stages of mining. Several algorithms are developed using the principle of DFS algorithms. Some of them are mentioned in the following sections:

2.3.1 SPADE algorithm

This algorithm is proposed in [17] and it includes the features of a search space partitioning where the search space includes vertical database layout. The search space in SPADE is represented as a *lattice* structure and it use the notion of equivalence classes to partition it. It decomposes the original lattice into slighter sub-lattices, so that each sub-lattice can be entirely processed using either a breadth-first or depth-first search method (SPADE is also DFS-based method). The SPADE support counting of the candidate sequence method includes bitwise or logical operations. The advantage of SPADE is, it uses a more efficient support counting method based on the structure and SPADE shows a linear scalability with respect to the number of sequences.

2.3.2 FreeSpan algorithm

FreeSpan is an algorithm proposed by Pei et al in 2001[17] with the aim to reduce the generation of candidate sub-sequences. It uses projected databases to generate database annotations in order to guide the mining process to find frequent patterns. The general idea of FreeSpan is to use frequent items to project sequence databases into a set of smaller projected databases recursively using the current mined frequent sets, and subsequence fragments in each projected database are generated, respectively. Two alternatives of database projections can be used Level-by-level projection or Alternative-level projection. The method used by FreeSpan divide the data and the set of frequent patterns to be tested, and limits each test being conducted to the corresponding smaller projected database. FreeSpan scan the original database only three times, whatever the maximal length of the sequence. Experimental results show that FreeSpan is efficient and mines the complete set of patterns and it is considerably faster than the GSP algorithm. The major cost of FreeSpan is to deal with projected databases.

2.3.3 PrefixSpan algorithm

This algorithm proposed in [18], this algorithm uses projection based algorithm. The general idea is to check only the prefix sub-sequences and only their corresponding postfix sub-sequences are projected into projected databases, rather than projecting sequence database. PrefixSpan uses a direct application of the apriori property in order to reduce candidate sequences alongside projected databases. Additionally, PrefixSpan is efficient because it mines the complete set of patterns and has a significantly faster running than both GSP algorithm and FreeSpan. The major cost of PrefixSpan, similarly to FreeSpan, is the construction of projected databases. For every sequential database, PrefixSpan needs to construct a projected database. After the database projection is made, the use of bilevel projection represented in FreeSpan and PrefixSpan by the S-Matrix [17][18] is an additional faster way to mine. The main idea of PrefixSpan algorithm is to use frequent prefixes to divide the search space and to project sequence databases. Its aim is to search the relevant sequences.

2.3.4 SPAM algorithm

SPAM Proposed in [19], this algorithm uses a depth-first traversal method combined with a vertical bitmap representation to store each sequence allowing a significant bitmap compression as well as an efficient support counting. SPAM uses a vertical bitmap representation of the data which are created for each item in the dataset. Each bitmap contains a bit representing each transaction in the dataset, if item i appears in transaction j, then the bit relative to transaction j of the bitmap for item i is set to 1; otherwise it is set to 0. An efficient counting and candidate generation can be enabled if the bitmap should be partitioned aiming to make sure all transaction sequences in the database appear together in the bitmap. The bitmap representation idea of SPAM requires quite a lot of memory, so it is very efficient for those databases which have very long sequential patterns. Additionally, a significant feature of this algorithm is the outputs of new frequent item sets in an online and incremental fashion. Experimental results show that this algorithm is more efficient compared to SPADE and PrefixSpan on large datasets, but it consumes more space compared to SPADE and PrefixSpan.

2.4 Closed sequential pattern-based algorithms

The algorithms of sequential pattern mining presented earlier mine the full set of frequent sub-sequences satisfying a minimum support threshold. Nevertheless, because a frequent long sequence contains a combined number of frequent sub-sequences, the mining process will generate a large number of frequent sub-sequences for long patterns, which is expensive in both time and space. The frequent pattern mining (item sets and sequences) needs not mine *all* frequent patterns but the *closed* ones since it leads to a better efficiency, which can really reduce the number of frequent sub-sequences [12]. In the following section, two algorithms CloSpan and BIDE [20]:are described:

2.4.1 CloSpan algorithm

This algorithm proposed by [12] with the aim to reduce the time and space cost when generating explosive numbers of frequent sequence patterns. CloSpan mines only frequent closed sub-sequences i.e., the sequences containing no super sequence with the same support, instead of mining the complete set of frequent sub-sequences. The mining process used by CloSpan is divided into two stages, a) A candidate set is generated in the first stage which is larger than the final closed sequence set. This set is called suspicious closed sequence set. b) A pruning method is called in the second stage to eliminate non-closed sequences. The main difference between CloSpan and PrefixSpan is that CloSpan avoids the unnecessary traversing of search space. The use of backward sub-pattern and backward super-pattern methods, some patterns will be absorbed or merged which, reduce the search space growth.

2.4.2 BIDE algorithm

Proposed by Wang and Han[20] which mines closed sequential patterns without candidate maintenance by adopting a closure checking scheme, called BI-Directional Extension. BIDE avoids the problem of the *candidate maintenance-and-test* paradigm used by CloSpan. It prunes totally the search space and checks efficiently the pattern closure which consumes a much less memory in contrast to the previously developed closed pattern mining algorithms. BIDE has a linear scalability with regards to the number of sequences in the database. Nevertheless, it will lose some all-frequent-sequence mining algorithms with a high support threshold, like other closed sequence mining algorithms. Experimental results conducted in [20] show that BIDE is more efficient than CloSpan.

3. SEQUENTIAL PATTERN MINING ALGORITHMS

Comparative analysis of sequential pattern mining algorithm is performed on the basis of various important features. For comparison various features are used. The acronym of the various algorithms used are:

1. **SPIRIT:** Sequential Pattern Mining with Regular Expression Constraints
2. **GSP:** Generalised Sequential Patterns
3. **SPADE**: Sequential Pattern Discovering using Equivalence classes
4. **FREESPAN**: Frequent Pattern Projected Sequential Pattern Mining
5. **PREFIXSPAN**: Prefix-Projected Sequential Pattern Mining
6. **SPAM:** Sequential Pattern Mining
7. **CloSpan**: Closed Sequential Pattern Mining
8. **BIDE**: Bi-directional extension

3.1 Characteristics of sequential pattern mining algorithm are

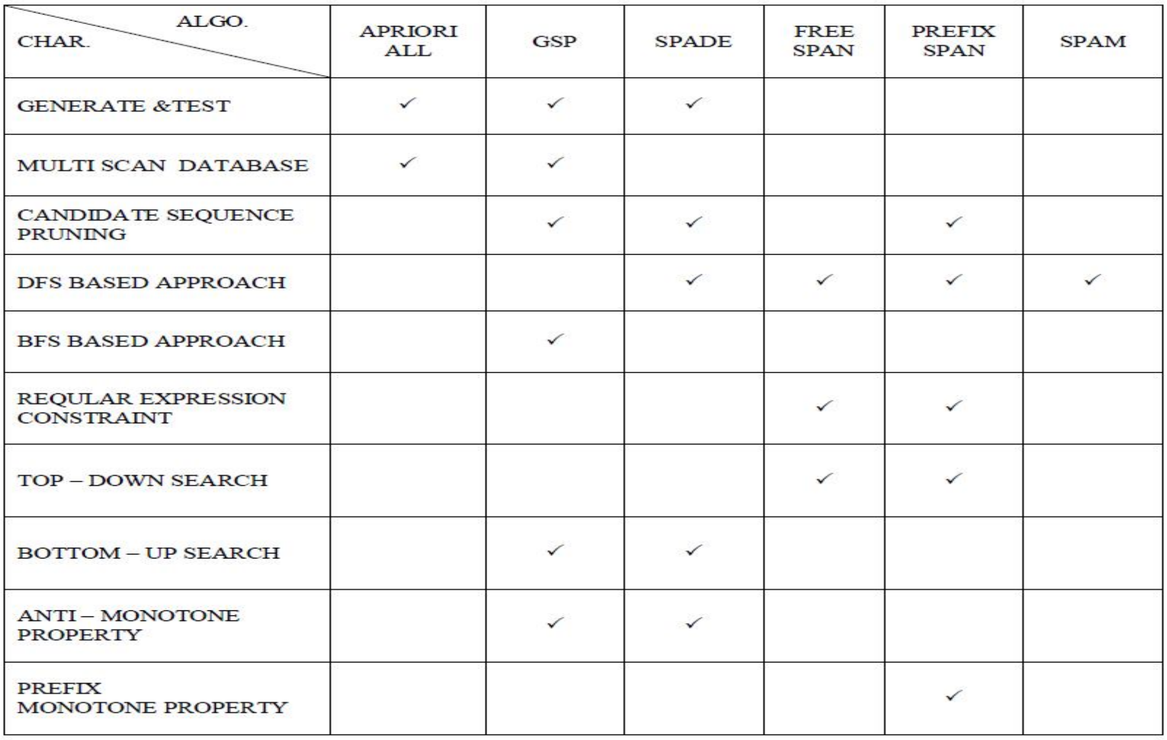
BFS-Based Approach Vs. DFS-Based Approach: In BFS approach level-by-level search is used to find the complete set of patterns i.e., all the child node are processed before moving to the next level. But in DFS approach all the sub-arrangements on a path must be explored before moving into the next one. The advantage of DFS is it can very quickly reach large frequent arrangements, so it avoids the expansion of other paths in the tree.

Top-Down Search Vs. Bottom-Up Search: Bottom-up search is used in Apriori-based algorithms which are used to display single frequent sequence. In top-down approach the subsets of sequential patterns can be mined by constructing the corresponding set of projected data bases and mining each repetition from top to bottom.

Anti-Monotone Vs. Prefix-Monotone Property: Anti-Monotone property states that every non-empty sub-sequence of a sequential pattern is a sequential pattern and prefix-Monotone property states that for each α sequence satisfying the constraint, every sequence having α as a prefix also stratifies the constraint.

Regular Expression Constraint: Complexity of regular expression can be roughly measured by the number of state changes in their corresponding deterministic finite automata. A regular expression constraint contains a property called growth-based anti-monotonic. A constraint is growth-based anti-monotonic if it has the following property: If a sequence satisfies the constraint must be reachable by growing from any component which matches parts of the regular expression.

Table1: Comparative features of different sequential pattern mining algorithms



ALGO. CHAR.	APRIORI ALL	GSP	SPADE	FREE SPAN	PREFIX SPAN	SPAM
GENERATE &TEST	✓	✓	✓			
MULTI SCAN DATABASE	✓	✓				
CANDIDATE SEQUENCE PRUNING		✓	✓		✓	
DFS BASED APPROACH			✓	✓	✓	✓
BFS BASED APPROACH		✓				
REQULAR EXPRESSION CONSTRAINT				✓	✓	
TOP – DOWN SEARCH				✓	✓	
BOTTOM – UP SEARCH		✓	✓			
ANTI – MONOTONE PROPERTY		✓	✓			
PREFIX MONOTONE PROPERTY					✓	

The table1 represents the various comparative features of sequential pattern mining algorithms. For comparison six different algorithms are employed and their specific features are included in the above table.

4. RESEARCH CHALLENGES

Many methods are available today for discovering efficient sequential patterns. Such patterns are widely applicable for a large number of applications. But in the field of data mining there are various research challenge, some of them are:

- The development of sequential pattern mining methods for particular applications, such as DNA sequences, and handling sequential process analysis.
- PrefixSpan mining methodology can be extended for mining sequential patterns with user-specified constraints, efficient mining of other kinds of frequent patterns, can be extended to bio-informatics challenging problems.
- To handle constraints on set of sequential patterns such as closedness, relevant sub-group and sky-pattern constraints.
- To discover sequential patterns of item sets in a sequence data base.
- Parallel/Distributed techniques can be used for analysing and mining huge data base.
- Apriori algorithm can be extended to web content mining and web structure mining.

5. CONCLUSION

In this paper a very important and complex data mining problem known as sequential pattern mining has been analysed in detail. This concept is being introduced in 1995, has gone through remarkable advancement in recent years. The sequential pattern mining algorithms are classified into four broad categories namely: apriori-based algorithm, breadth first search, depth first search and sequential closed pattern algorithms. The paper presents a detailed explanation of their features, advantages and disadvantages. The comparative analysis of these algorithms is made based on various parameters. In depth research work needs to be made for extending the capabilities of existing sequential mining approaches.

References

[1] Pei, Jian, Helen Pinto, Qiming Chen, Jiawei Han, Behzad Mortazavi-Asl, Umeshwar Dayal, and Mei-Chun Hsu. "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth." 29th International Conference on Data Engineering (ICDE), pp. 0215-0215. IEEE Computer Society, 2001.

[2] Agrawal, R.; Srikant, R., "Mining sequential patterns," Proceedings of the Eleventh International Conference on Data Engineering, 1995. , vol., no., pp.3,14, 6-10 Mar 1995 doi: 10.1109/ICDE.1995.380415.

[3] J. Han and M. Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufman publishers, 2001.

[4] Han, Jiawei, Micheline Kamber, and Jian Pei. Data mining: concepts and techniques. Morgan kaufmann, 2006.

[5] Pei, Jian, "Mining sequential patterns by pattern-growth: The PrefixSpan approach." Knowledge and Data Engineering, IEEE Transactions on 16.11 (2004): 1424-1440.

[6] Masseglia, Florent, Maguelonne Teisseire, Pascal Poncelet. "Sequential Pattern Mining." (2009): 1800-1805.

[7] Manan Parikh, Bharat Chaudhari and Chetna Chand, A Comparative Study of Sequential Pattern Mining Algorithms, Volume 2, Issue 2, February 2013, International Journal of Application or Innovation in Engineering & Management (IJAIEM) .

[8] Thabet Slimani, and Amor Lazzez, Sequential Mining: Patterns And Algorithms Analysis.

[9] J.Pei, J.Han, B.MortazaviAsl, J.Wang, H.Pinto, Q.Chen, U.Dayal and M.-C.Hsu, ―Mining sequential patterns by pattern-growth: The PrefixSpan approach‖, IEEE Transactions on Knowledge and Data Engineering, vol.16, no.11, 2004, pp. 1424-1440.

[10] Helen Pinto Jiawei Han Jian Pei Ke Wang, ―Multidimensional Sequential Pattern Mining‖, In Proc. 2001 Int.
Conf. Information and Knowledge Management (CIKM'01), Atlanta, GA, Nov. 2001 pp. 81–88.

[11] Jian Pei, Jiawei Han, Wei Wang, ―Constraint-based sequential pattern mining: the pattern growth methods‖, J Intell Inf Syst , Vol. 28, No.2, ,2007, pp. 133 –160.

[12] Yen-Liang Chen, Mi-Hao Kuo, Shin-Yi Wu, Kwei Tang, ‖Discovering Recency, frequency, and monetary (RFM) sequential patterns from customers' purchasing data‖, Electronic Commerce Research and Applications 8 (2009), 2009, pp. 241–251.

[13] Hao-En Chueh, ―Mining Target-Oriented Sequential Patterns with Time-Interval‖, International journal of computer science & information Technology (IJCSIT) Vol.2, No.4, August 2010.

[14] Yan, X., Han, J., and Afshar, R., ―CloSpan: Mining closed sequential patterns in large datasets‖, In Third SIAM International Conference on Data Mining (SDM), San Fransico, CA, 2003, pp. 166–177

[15] R. Srikant, R. Agrawal: Mining Quantitative Association Rules in Large Relational Table., Proc. of the ACMSIGMOD 1996 Conference on Management of Data, Montreal, Canada, June 1996.

[16] M.Zhang, B.Kao, CL.Yip, D.Cheung. A GSP-based efficient algorithm for mining frequent sequences. In Proc. of IC-AI'2001. June 2001, Las Vegas, Nevada, USA.

[17] M.J. Zaki. Scalable algorithms for Association Mining. IEEE Transactions on Knowledge and Data Engineering. 12(3), 372-390, 2000.

[18] J.Han , J.Pei, B.Mortazavi-Asl, Q. Chen, U.Dayal, And M.-C. Hsu. Freespan: Frequent pattern projected sequential pattern mining. In Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York, 355–359, 2000.

[19] J.Ayres, J.Flannick, J.Gehrke and T. Yiu. Sequential Pattern Mining Using a Bitmap Representation, Proceedings of Conference on Knowledge Discovery and Data Mining, pp. 429–435, 2002.

[20] J.Wang, J.Han. BIDE: Efficient mining of frequent closed sequences. In Proc. of 2004 Int. Conf. on Data Eng. Apr. 2004, Boston, MA. 79–90.

[21] Zheng Zhu, "Data Mining Survey - ver 1.1009".