AN EFFICIENT ALGORITHM FOR SEQUENCE GENERATION IN DATA MINING

Dr.S.Vijayarani and Ms.S.Deepa,

[1]Assistant Professor, Department of computer Science, School of Computer Science & Engineering, Bharathiar University, Coimbatore, Tamilnadu, India.
[2]M.Phil Research Scholar, Department of computer Science, School of Computer Science & Engineering, Bharathiar University, Coimbatore, Tamilnadu, India.

ABSTRACT

Data mining is the method or the activity of analyzing data from different perspectives and summarizing it into useful information. There are several major data mining techniques that have been developed and are used in the data mining projects which include association, classification, clustering, sequential patterns, prediction and decision tree. Among different tasks in data mining, sequential pattern mining is one of the most important tasks. Sequential pattern mining involves the mining of the subsequences that appear frequently in a set of sequences. It has a variety of applications in several domains such as the analysis of customer purchase patterns, protein sequence analysis, DNA analysis, gene sequence analysis, web access patterns, seismologic data and weather observations. Various models and algorithms have been developed for the efficient mining of sequential patterns in large amount of data. This research paper analyzes the efficiency of three sequence generation algorithms namely GSP, SPADE and PrefixSpan on a retail dataset by applying various performance factors. From the experimental results, it is observed that the PrefixSpan algorithm is more efficient than other two algorithms.

KEYWORDS

Sequential Pattern, GSP, SPADE, PrefixSpan, candidate generation, minimum_support, projection based.

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 [15]. It consists of effective techniques that help to bring out the hidden knowledge in huge volume of data. The major issue of data mining in the recent years has been 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 subsequences that is it identifies the subsequences whose occurrence frequency in the set of sequences is not less than minimum_support threshold [1].

Sequential pattern mining [11] has been emerging as an important data mining task since it has broad application in market and customer analysis, web log analysis, intrusion detection system and mining protein, gene and in DNA sequence patterns. The revealed information and knowledge are widely used in various applications including learning status analysis, decision

DOI: 10.5121/ijci.2014.3103

support, disease prediction and fraud detection. It is a resourceful technique for discovering recurring structures or patterns from very large dataset. Many algorithms are proposed for sequential pattern mining [10]. The algorithms for Sequential Pattern Mining mainly differ in two ways [6]:

- The algorithms may differ in the way in which candidate sequences are generated and stored. Reducing the number of candidate sequences generated is the main task of these algorithms which in turn minimizes the I/O cost.
- The support value is calculated in various methods in these algorithms and the method of testing the candidate sequences for frequency is also different in these algorithms. The database has to be removed or the data structure has to be maintained all the time for support of counting purposes only.

Based on these conditions sequential pattern mining can be divided broadly into two parts [2]:

- Apriori based(GSP, SPADE, SPAM)
- Pattern growth based(FreeSpan, PrefixSpan)

In this paper, comparison is made between GSP and SPADE from Apriori-Based algorithm and PrefixSpan from pattern growth approach. The study shows that PrefixSpan outperforms better than GSP and SPADE algorithm for very large dataset.

The rest of the paper is organized as follows: Section 2 gives the review of literature. Section 3 deals with the problem objective and the proposed methodology. Section 4 discusses the GSP, SPADE and PrefixSpan algorithms. Performance analysis and experimental results are presented in Section 5 and conclusions are given in Section 6.

2. LITERATURE REVIEW

Sequential pattern mining is computationally challenging because such mining may generate and/or test a combinatorial explosive number of intermediate sequence. Many novel algorithms are proposed such as Apriori, AprioriALL, GSP, SPADE, SPAM and PrefixSpan.

Jian Pei, et al., [4] have performed a logical study on the mining of sequential patterns in Gazelle data set from Blue Martini and a pattern-growth approach had been proposed for the efficient and scalable mining of sequential patterns. Instead of refinement of sequence patterns like in the apriori-like and also instead of candidate generation-and-test approach such as in GSP, a divide-and-conquer approach called the pattern-growth approach, PrefixSpan is promoted which proves to be an efficient pattern-growth algorithm for mining frequent patterns without candidate generation. PrefixSpan recursively projects a sequence database into a set of smaller projected sequence databases and grows sequential patterns in each projected database by exploring only locally frequent fragments. The entire set of sequential patterns is mined and substantially reduces the efforts of candidate subsequence generation.

Thomas. Rincy. N and Yogadhar Pandey [12] observed the performance evaluation trend in "BMS-Webview1 dataset and Toxin-Snake dataset and showed that the SPAM method performs much better and has a better scalability than PrefixSpan in terms of execution time while in terms of memory usage, the method clearly indicates that SPAM has stable memory usage than PrefixSpan for all minimum support values. PrefixSpan algorithm is implemented with pseudoprojection technique and still by observing the performance evaluation trend it clearly shows that SPAM can be faster on sparse and dense datasets, also the memory consumption is stable as compared to PrefixSpan which is contradictory to the traditional standpoint. SPAM it

generally consumes more memory than PrefixSpan and SPAM is faster on dense datasets with long patterns and less efficient on other dataset and also it consumes more memory.

Mahdi Esmaeili and Fazekas Gabor [16] theoretically have shown three types of sequential patterns and some of their properties. These models fall into three classes are called periodic pattern, approximate pattern and statistically pattern. Periodicity can be full periodicity or partial periodicity. In full periodicity method, every time point contributes to the cyclic behavior of a time series. In contrast some time points in partial periodicity contribute to the cyclic behavior of a time series. This model of pattern is so rigid. The information gain can be used as a new metric that help us to discover the surprising patterns.

Niti Desai and Amit Ganatra [8] had performed a theoretical and simulation study on various sequential pattern mining algorithms. They proposed that PrefixSpan is an efficient pattern growth method because it outperforms GSP, FreeSpan and SPADE. They showed that the PrefixSpan Algorithm is more efficient with respect to running time, space utilization and scalability than Apriori based algorithms. Most of the existing SPM algorithms work on objective measures Support and Confidence. Their experiments showed that the percentage reduction of rule generation is high in case of interestingness measures lift. They also explained that use of interestingness measures can lead to make the pattern more interesting and can lead to indentify emerging patterns.

3. PROBLEM OBJECTIVE AND METHODOLOGY:

3.1 Definition

Let I= $\{i_1,i_2,\ldots i_n\}$ be a set of all items. An itemset is a subset of items. A **sequence** is an ordered list of itemsets. A sequence s is denoted by $<s_1,s_2,\ldots s_n>$ where s_i is an itemset. The number of instances of items in a sequence is called the **length of the sequence**. A sequence $\alpha=<a_1,a_2,\ldots a_n>$ is called a **subsequence** of another sequence $B=<b_1,b_2,\ldots b_m>$ and β a **supersequence** of α, if there exist integers $1 \leq j_1 < j_2 < \ldots < j_n \leq m$ such that $a_1 \subseteq b_{j1}$; $a_2 \subseteq b_{j2}$; . . . ; $a_n \subseteq b_{jn}$ .A **sequence database** S is a set of tuples <sid, si> where sid is a sequence_id and s a sequence[1].

3.2 Problem Statement: Given a sequence database and the minimum_support threshold value, the charge of sequential pattern mining is to find the complete set of sequential patterns in the database. Three techniques namely GSP, SPADE and PrefixSpan are used for generating sequences and the performance of these algorithms are analyzed and compared for finding the efficient technique.

The system architecture of the research work is as follows:

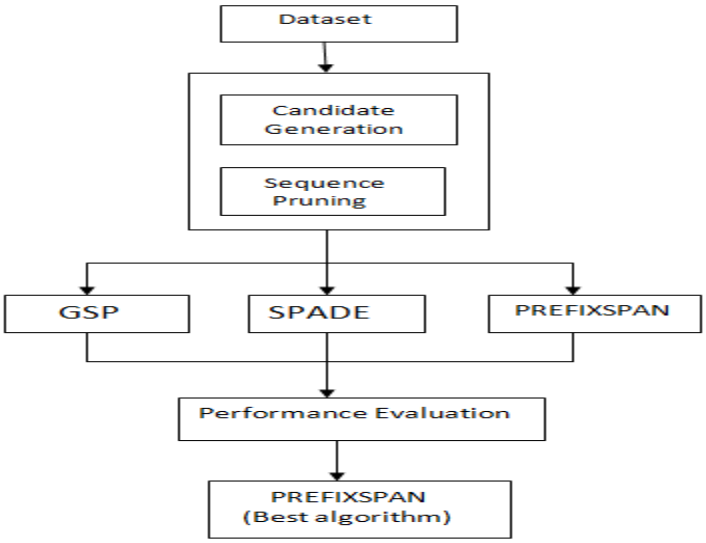

Fig 1: System architecture

4. ALGORITHMIC APPROACHES TO THE TASK OF SEQUENCE MINING

When developing an algorithm to the task of sequence mining, the idea is to make it more efficient in terms of memory requirements and to reduce as much as possible the response time [10]. That will imply the use of appropriated data structures and the use of old and novel algorithmic approaches to perform this task.

4.1 GSP:

GSP Algorithm (Generalized Sequential Pattern algorithm) is the initial algorithm that is used for sequence mining [7]. The GSP algorithm is a breadth-first search algorithm. It has the anti-monotone property where all the subsequences of a frequent sequence must be also frequent. The algorithm works in phases and performs multiple passes over the database.

For solving the sequence mining problems, various algorithms are used and they are mostly based on the a priori (level-wise) algorithm. The level-wise theory first identifies all the frequent items in a level-wise fashion. The occurrences of all the singleton elements in the database are counted. Then the transactions are modified by removing the non-frequent items. Then each of the transaction consists of only the frequent elements that it originally contained. This modified database is made as an input to the GSP algorithm. This process scans the whole database once.

GSP Algorithm makes multiple passes over the database. In the first pass, a set of candidate 1-sequences are identified. From the identified frequent items, a set of candidate 2-sequences are generated and another pass is made to calculate their frequency of occurrence. The frequent 2-sequences are then used to generate the candidate 3-sequences and this process is repeated until no more frequent sequences are found. The algorithm has two important steps.

- Candidate Generation: Given the set of frequent sequences F(p-1), the candidates item sets for the next pass are generated by joining F(p-1) with itself. Then pruning of the database is performed that eliminates any sequence at least one of whose subsequences is not frequent.
- Support Counting: The search based on hash tree structure is employed for efficient support counting. Finally the sequences that are non-minimal are removed.

GSP Algorithm [14]:

F1={frequent 1-sequences};
For ($p=2;p_{k-1}\neq \theta;p=p+1$) do
C_p=Set of candidate k-sequences;
for all input-sequences ε in the database do
Increment count of all $\alpha \varepsilon C_p$ contained in ε
$F_k=\{\alpha \varepsilon C_p | \alpha.sup>=min\_sup\}$;
Set of all frequent sequences=U_pF_p;

One of the pitfalls of the GSP is that it generates large number of candidates that is with increasing length of sequences and the number of frequent sequences that has the tendency to decrease where the number of candidates generated by GSP is still enormous. Additionally the GSP algorithm performs multiple scans of the database which also slows down the process.

4.2 SPADE:

Spade utilizes the prefix-based equivalence classes that decompose the original problem in to smaller sub-problems that can be solved independently in main memory using simple join operations [14]. All sequences are identified in three database scans. It uses vertical representation of the database, that is each row consists of event uniquely identified by sequence Id (sid for short) and event id (eid for short).

The key features of approach are as follows:

- A vertical id-list database format is used where each of the sequence is associated with a list of objects in which it occurs along with the time-stamps. All frequent sequences can be enumerated via simple temporal joins on id-lists.
- A lattice-theoretic approach is used to divide the original search space (lattice) into smaller pieces (sub-lattices) which can be then processed independently in main-memory. The approach is performed in three database scans or only a single scan with some pre-processed information thus minimizing the I/O costs.
- The problem is then decomposed by decopuling from the pattern search. Two different search strategies are proposed for enumerating the frequent sequences within each sublattice: breadth-first and depth-first search.

SPADE algorithm [14]:

SPADE (min_sup,D):
F1={frequent items or 1-sequences};
F2={frequent 2-sequences};
E={equivalence classes[X]θ_1};
For all [X]εE do Enumerate-Frequent-Seq([X]);

SPADE minimizes the I/O costs by reducing database scans and as well as minimizes the computational costs by employing efficient methods for searching. Data-skew can occur since the vertical id-list based approach is insensitive to it.

4.3 PREFIXSPAN:

PrefixSpan is different from the normal way of generating candidates and testing them such as GSP and SPADE. The PrefixSpan algorithm has two key features [5]:

- It is projection-based.
- The patterns are generated sequentially in the projected databases by investigating only locally frequent segments.

The PrefixSpan algorithm:

Algorithm PrefixSpan
Input a sequence database S and the minimum support threshold, min_support
Call PrefixSpan(<>,0,S)
Procedure PrefixSpan (α, L, Sα)
1) Scan Sα once, find each frequent item b, such that:
   a) b can be assembled to the last element of α to form a sequential pattern; or
   b) <b> can be appended to α to form a sequential pattern.
2) For each frequent item b, append it to α to form a sequential pattern α' and output α'.
3) For each α', construct α'-projected database Sα'.
4) Call PrefixSpan (α', L+1, Sα')

The PrefixSpan method is significantly different from the two algorithms mentioned above. The initial process of PrefixSpan is to scan the sequential database and to extract the length-1 sequence. Then the sequential database is divided into various partitions based on the number of length-1 sequences and each partition is the projection of the sequential database that takes the corresponding length-1 sequences as prefix. The projected databases only contain the postfix of these sequences by scanning the projected database all the length-2 sequential patterns that have the parent length-1 sequential patterns as prefix can be generated. Then the projected database is partitioned again by those length-2 sequential patterns. The same process is executed recursively until the projected database is empty or no more frequent length-k sequential patterns can be generated. An essential advantage of the PrefixSpan is that no candidate sequence needs to be generated [3].

PrefixSpan outperformed the other methods mainly in three ways:

- It grows patterns without candidate generation.
- The data reduction can be performed effectively by the projections.
- The memory space utilization is approximately steady.

5. PERFORMANCE EVALUATION

To compare the performance of GSP, SPADE and PrefixSpan algorithms, a series of experiments are performed with retail market basket data set. The dataset used in this paper is taken from Frequent ItemSet Mining Repository. http://fimi.ua.ac.be/data/retail.dat. Retail dataset is used in this research work. It is a real time dataset collected from a Belgian Retail Supermarket store. The dataset consists of 88,163 transactions and 16,440 different products that are sold in various transactions carried over in a certain period of time. The transactions consist of unique ids that are given for each product that was provided by the store.

In this research work, three sequential pattern mining algorithms namely GSP, SPADE and PREFIXSPAN are used to generate the sequential patterns from the retail dataset. For this study, various threshold levels are used and their results are analyzed. The size of the transaction is 100 and the average length of the transactions is 8. The samples of sequence patterns generated by these algorithms are given in Table 1 below:

Table 1: Sample Sequences generated by the algorithms

S.No	SEQUENCES PRODUCED
1	(105)
2	(105, 152)
3	(105, 225)
4	(105, 32)
5	(105, 152, 225)
6	(105, 152, 225, 32, 36)

The Table 2 shows the execution time of GSP, SPADE and PREFIXSPAN algorithms at various threshold levels for the retail data set.

Table 2: Execution Time for GSP, Spade and PrefixSpan

Algorithm	Execution Time(in millisecs)		
	Min_sup=5	Min_sup=10	Min_sup=15
GSP	36	37	39
Spade	28	30	27
Prefixspan	22	20	21

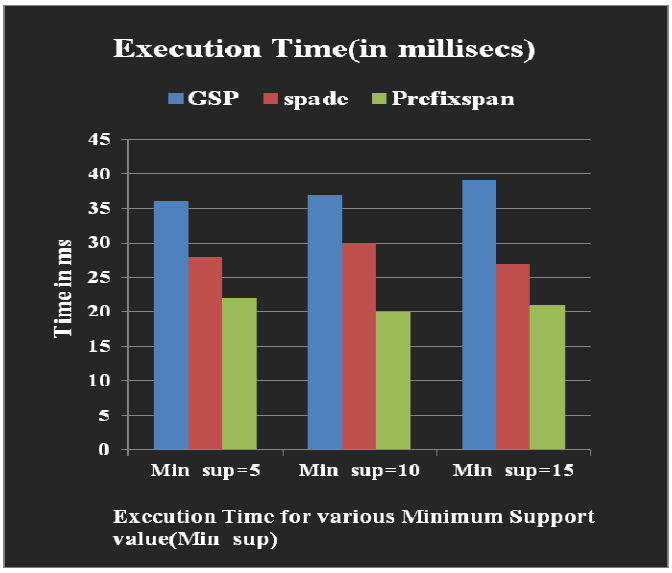

Fig 2: EXECUTION TIME: GSP, SPADE and PrefixSpan

The above graph shows the execution time of three algorithms. From the experimental results, the PrefixSpan algorithm needs less execution time than GSP and SPADE algorithm.

The Table 3 shows the memory space utilized by GSP, SPADE and PREFIXSPAN algorithms at various threshold levels.

Table 3: Memory Space Utilization

Algorithms	**Memory Space Utilization(in kb)**		
	Min_sup=5	**Min_sup=10**	**Min_sup=15**
GSP	422	386	388
spade	350	351	346
PrefixSpan	323	323	317

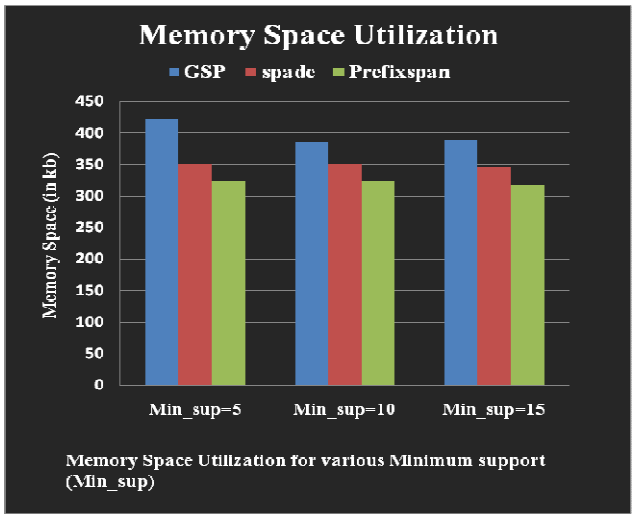

Fig 3: MEMORY SPACE UTILIZATION: GSP, SPADE and PrefixSpan

The above graph shows the memory space utilization of sequence generation algorithms. The results shows that the memory space utilized by PrefixSpan occupies less memory space compared to GSP and SPADE algorithm.

6. CONCLUSION:

Sequential mining has been attracting attention in recent research in the field of data mining. Since the search space is very large and data volume is huge, it has made many problems for mining sequential patterns. In order to effectively mine the sequential patterns, efficient sequential pattern mining algorithms are needed. Among the sequential pattern algorithms GSP, SPADE and PrefixSpan, PrefixSpan is an efficient pattern growth method because it outperforms the other two algorithms. It is clear that PrefixSpan Algorithm is more efficient with respect to running time, space utilization and scalability than Apriori based algorithms. Future research may involve the development of novel measures which can make the pattern more interesting and can be helpful to identify emerging patterns.

REFERENCES:

[1] R.Agrawal and R.Srikant, “Mining Sequential Patterns," In Proceedings of International conference on data engineering. pp. 3-14

[2] Chetna Chand, Amit Thakkar, Amit Ganatra- Sequential Pattern Mining: Survey and Current Research Challenges, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012

[3] George Aloysius, D. Binu,” An approach to products placement in supermarkets using PrefixSpan algorithm”, Journal of King Saud University – Computer and Information Sciences (2013) 25, 77–87

[4] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen,Umeshwar Dayal, Mei-Chun Hsu, “Mining Sequential Patterns by Pattern-Growth:The PrefixSpan approach”, IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 10, October 2004

[5] Manan Parikh, Bharat Chaudhari and Chetna Chand, “A Comparative Study of Sequential Pattern Mining Algorithms”, International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 2, Issue 2, February 2013 ISSN 2319 – 4847

[6] Manish Gupta, Jiawei Han, “Approaches for Pattern Discovery Using Sequential Data Mining”

[7] Minghua Zhang, Ben Kao, Chi-Lap Yip, David Cheung, “A GSP-based Efficient Algorithm for Mining Frequent Sequences”

[8] Niti Desai1, Amit Ganatra, "Sequential Pattern Mining Methods: A Snap Shot", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 10, Issue 4 (Mar. - Apr. 2013), PP 12-20
[9] Qiankun Zhao, Sourav S. Bhowmick, "Sequential Pattern Mining: A Survey".
[10] Pedro Gabriel Dias Ferreira," A survey on Sequence Pattern Mining Algorithms".
[11] Sushila Umesh Ratre, Prof. Ravindra Gupta, "An Efficient Technique for Sequential Pattern Mining", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X , Volume 3, Issue 3, March 2013.
[12] Thomas. Rincy. N, Yogadhar Pandey, "Performance Evaluation on the State of the art of Sequential Pattern Mining Algorithms", International Journal of Computer Applications (0975 – 8887) Volume 65– No.14, March 2013
[13] Tomas Kacur, "Mining of frequent subsequences in databases"
[14] M. Zaki, "SPADE: An efficient algorithm for mining frequent sequences, Machine Learning, 2001.
[15] Zheng Zhu, "Data Mining Survey - ver 1.1009"
[16] Mahdi Esmaeili and Fazekas Gabor, "Finding Sequential Patterns from Large Sequence Data", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 1, No. 1,January 2010 ISSN (Online): 1694-0784 ISSN (Print): 1694-0814
[17] Huan-Jyh Shyur, Chichang Jou1, Keng Chang, "A data mining approach to discovering reliable sequential patterns", The Journal of Systems and Software 86 (2013) 2196– 2203.

Authors

Dr. S.Vijayarani has completed MCA, M.Phil and Ph.D in Computer Science. She is working as Assistant Professor in the School of Computer Science and Engineering, Bharathiar University, Coimbatore. Her fields of research interest are data mining, privacy and security issues, bioinformatics and data streams. She has published papers in the international journals and presented research papers in international and national conferences.

Ms. S.Deepa has completed M.Sc in Computer Science. She is currently pursuing her M.Phil in Computer Science in the School of Computer Science and Engineering, Bharathiar University,Coimbatore. Her fields of interest are Sequence pattern mining and privacy preserving data mining.