IMPROVING MULTIDIMENSIONAL SEQUENTIAL PATTERN MINING ALGORITHM

Mitul V. Patel, Prof. Risha Tiwari

Student, Computer Science and Engineering, Hasmukh Goswami Collage of Engineering, Gujarat, India
Assistant Professor, Computer Science and Engineering, Hasmukh Goswami Collage of Engineering, Gujarat, India

ABSTRACT

Data mining is an influential technology to help companies to extract hidden and significant predictive information from massive collection of data. In present time, it has become very extensive field for research. In data mining, Frequent pattern mining is receiving increasing attention amongst researchers. It is important and elementary technique used to find frequent patterns. It is having numerous algorithms to find frequent items from given database. Sequential pattern mining, which finds the set of frequent subsequences in sequence databases, is an important data mining task and has broad applications. Usually, sequence patterns are associated with different circumstances, and such circumstances form a multiple dimensional space. For example, customer purchase sequences are associated with region, time, customer group, and others. It is interesting and useful to mine sequential patterns associated with multi-dimensional information. we propose the theme of multi-dimensional sequential pattern mining, which integrates the multidimensional analysis and sequential data mining. We also thoroughly explore efficient methods for multi-dimensional sequential pattern mining. We examine feasible combinations of efficient sequential pattern mining and multidimensional analysis methods, as well as develop uniform methods for high-performance mining.

---

1. INTRODUCTION

As Information Technology is growing, databases created by the organizations are becoming huge. These organization sectors include marketing, telecommunications, manufacturing, banking, transportation etc. To extract valuable data, it necessary to explore the databases completely and efficiently. Data mining helps to identify valuable information in such huge databases. Data Mining is an analytic process designed to explore data in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. Data mining has been very interesting topic for the researchers as it leads to automatic discovery of useful patterns from the database. This is also known as Knowledge Discovery [14]. Many techniques have been developed in data mining amongst which primarily Frequent pattern mining or Association rule mining is very important which results in association rules. These rules are applied on market based analysis, medical applications, science and engineering, music data mining, banking etc for decision making.

1.1 Objective

The main intention of this thesis is to propose an efficient method to extract frequent patterns from database .From these patterns, interesting association patterns are found base on factors like frequency of items, profit, significance and confidence. Mainly frequency of items is important in the proposed method.

1.2 Motivation

Databases are increasing in the fields like business, medical, sports, education, transportation, IT etc. With the increase of database, for a particular user who is looking for a pattern not for complete data in the database, it is better to read wanted data than unwanted data. This wanted data and other advantage by data mining technique is

that only required pattern will be drawn from database with in short time. Clearly, if sequential pattern mining can be associated with customer category or other multi-dimensional information, it will be more effective since the classified patterns are often more useful. Similar situations exist in many practical applications. This motivates our study of Multidimensional sequential pattern mining [7].

2. Literature Survey

In paper [1] two approach sequential pattern mining and multidimensional analysis has to combine. It can be categorized in two categories: (1) Sequential pattern are integrated into multidimensional analysis (2) To incorporate multidimensional information in sequential and then apply the sequence pattern mining. Appriori algorithm is used to generate the sequential pattern but there is two problems generate in Apriori: (1) Scan the database multiple time (2) Generate the large dataset of sequence pattern. Uniseq algorithm is used to take the last element or consider sequence pattern after that apply multidimensional (<bf>,(*,mumbai,*). Dim-seq algorithm first find multidimensional pattern and then from that sequence pattern is find. Seq-dim algorithm first find sequence pattern and apply on that dimension pattern. In paper [1] only one dimention is consider for sequential pattern. But now a day's day by day database are incresed an online stocktrading site may provide a lot of related services, including trading stock, providing investment information, offering personal finance services, and supporting various banking activities. In a single day, the customer may take a series of sessions, where a session starts when the user logs in and ends when the user logs out. Adding this new dimension can help us to analyze the patterns of stock trading and to understand how the activities evolve and are related in a session, a day, or a longer period of time. In paper [2] two algorithm are modify apriori and prefixspan. AprioriMD algorithm is used first same generate candidate at apriori then use a tree structure to generate the cardinality. PrefixMDspan algorithm is find. In Paper [2] the aim is discovering co-relation between event through time rule can be built like customer who brought a TV and a DVD player at the same time later bought a recorder. This type of work is already done in previous research. But rule like A customer who bought a surfboard and a bag in NY later bought a wetsuit in SF. In this rule city, product and time three dimension are consider. Therefore in paper [3] M2SP algorithm is develop in this sequential pattern are mined from relational table that can be seen as a fact table in multidimensional database. Inter pattern multidimensional sequential is work. So for this jokrised value * be consider. In paper [4] propose a novel HYPE (HierarchY Pattern Extension) approach which is an extension of previous M2SP proposition [3]. The main unique feature of this approach is that no single hierarchy level is considered and that several levels can be mixed. Extracted sequential patterns are automatically associated with the most relevant hierarchy levels.

In paper [5] building rules like when the sales of soft drinks are high in Europe, exports of Perrier in Paris and exports of soda in the US become high later on. This rule not only combines two dimensions (Location and Product) but it also combines them over time and at different levels of granularity (as Perrier is considered as a kind of soft drink). This is designed so as to take into account the most appropriate level of granularity, including partially instantiated tuples in sequences where the highest level is considered. More precisely, algorithms are designed in order to mine frequent multi-dimensional sequences that may contain several levels of hierarchy, including the most general, usually referred to as the ALL value. However, in order to avoid mining non relevant patterns, only the most specific patterns mining non relevant patterns, only the most specific patterns (meant to be the most informative) are shown to the user. For that they have use Spade algorithm to generate Sequential pattern from multidimensional sequential data. So we propose that we can use the Prefixspan algorithm instead of Spade algorithm to generate Sequential pattern.

3. PROPOSED ALGORITHM

Algorithm 1: Mining Minimal Atomic Frequent Sequences and Pruning Hierarchies [5]

Data: The set B (DR), the minimum support threshold minsup
Result: The set MAFS of all maf-sequences
begin
foreach i = 1,...,m do
Associate every value d in Hi with count(d) = 0;
foreach B ∈ β (DR) do
foreach tuple c = (t,d1,...,dm) in B do
Update all values count(δi), for every i = 1,...,m and δi ∈ δ↑ i ;
/*All counts are computed */

```
L1 ← φ;
foreach dj ∈ Si=m i=1 Hi do
  if count(dj) < minsup then
        /*Pruning Hj */
        Remove dj from Hj
 else
      /*Computing L1 */
       if up(dj) = {ALLj} then
            L1 ← L1 ∪ {<{(ALL1,...,ALLj−1,dj ,ALLj+1,...,ALLm)}>};
   MAFS ← ϕ;
   /*Recursive computation of maf-sequences */
 foreach a ∈ L1 do
            call get_rec_maf_seq(a,minsup,σa(B (DR )));
  return MAFS
  end
```

Algorithm 2: Routine get_rec_ma f_ seq

Data: A multi-dimensional item a, the minimum support threshold minsup, the set of blocks B (DR)

Subroutine: PrefixSpan(α, L, S|α).

Parameters:

α: sequential pattern,
L: the length of α;
S|α: the α-projected database, if α ≠<>; otherwise; the sequence database S.

Result: The current set MAFS

Method: [11]

1. Scan L1 once,
   find the set of frequent items b such that:
   b can be assembled to the last element of α to form a sequential pattern; or
   <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';
   output α';
3. For each α':
   construct α'-projected database S|α' and
   call PrefixSpan(α', L+1, S|α').

In our approach we can use the Prefix span algorithm instead of Spade algorithm. Because of MAFF sequences are generated from algorithm-1 is then applying to Spade algorithm for generated the sequence. And this sequence generated time is more as compare to Prefix span. Prefixspan use the Projection based technique to generate sequence, it take less time for result and result is also more efficient compare to Spade[13].

3.1 Pre-process Data Set:

We use The American Sign Language database created by the National Centre for Sign Language and Gesture Resources at Boston University, consists of a collection of utterance, a dataset of sign language utterance containing approximately 800 sequences.

3.2 Experimental Result

For generating the output we have used JAVA language to implement the algorithm and To run the algorithm we have used Eclipse framework.

Minimum Support	Spade	Prefixspan	Prefixspan-BUC	Spade-BUC
0.2	267	1928	919	194
0.3	267	1928	919	194
0.4	267	646	265	194
0.6	267	64	36	194
0.8	267	9	7	194

Table-1 Frequent Count table from Experiments result

Here table show Different Frequent pattern are generated as per different min support count. Different algorithm use and as per table we can see different frequent pattern are generated from dataset we have apply.

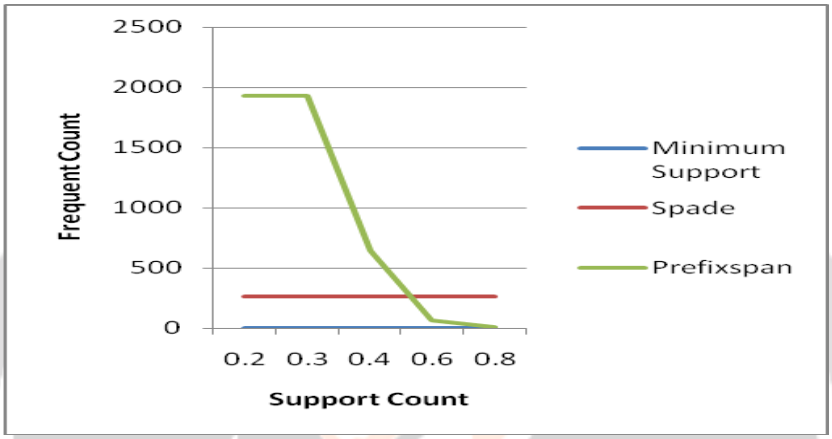

Fig 1 Frequent generated comparison between Spade and Prefix Span

Fig shows the comparison between Spade algorithm and Prefix span algorithm. Here we generate the chart from Experiment result. In this figure we see that Number of Frequent pattern are generated through Prefixspan is more as compare to spade algorithm. Here if minimum support count = 0.2 then frequent pattern generated from prefixspan is 1928 as in Spade is only 267. So if the number of frequent counts more then analysis is more powerful. From this figure we can say that Prefix span is better than Spade in terms of frequent pattern generation. As compare to Spade result Prefix span give more efficient result. It use projection based technique to generate the sequence pattern.

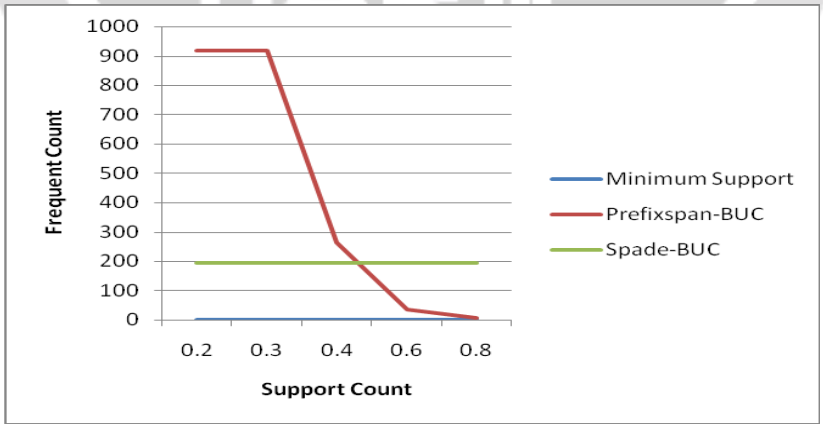

Fig 2 Frequent generated comparison between BUC-Spade and BUC-Prefix Span

Fig shows the comparison between BUC-Spade algorithm and BUC-Prefix span algorithm. Here we generate the chart from Experiment result. In this figure we see that Number of Frequent pattern are generated through BUC-Prefixspan is more as compare to BUC-spade algorithm. Here if minimum support count = 0.2 then frequent pattern generated from BUC-prefixspan is 919 as in BUC-Spade is only 194. So if the number of frequent counts more then

analysis is more powerful. From this figure we can say that BUC-Prefix span is better than BUC-Spade in terms of frequent pattern generation. As compare to BUC-Spade result BUC-Prefixspan give more efficient result.

Minimum Support	Spade	Prefixspan	Prefixspan-BUC	Spade-BUC
0.2	56	18246	10711	34
0.4	33	1891	1334	21
0.6	49	190	134	26
0.8	36	110	89	14

Table-2 Runtime(ms) table from Experiment result

Here table Show the time required to execute different algorithm as per Support Count. Different algorithm use set different support count and different runtime is generated.

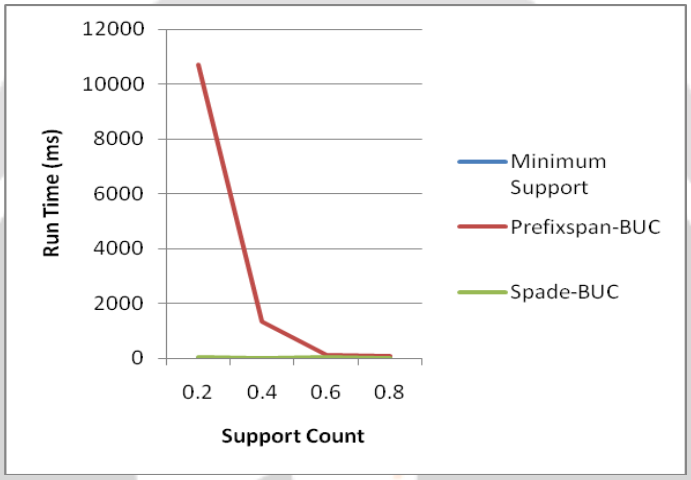

Fig 3 Runtime comparison between BUC-Spade and BUC-Prefix Span

Figure show the runtime required to execute the algorithm. The runtime is major in the ms. As per shown in figure BUC-Spade algorithm required less time as compare to BUC-Prefixspan. In BUC-Prefixspan algorithm if support count is between 20% to 50% then required more time. But here as compare to time and frequent pattern result, the result is more required factor in now-days. Because of now-days data base is very huge. So we have to required accurate result, and that for BUC-prefixspan give the better result as compare to BUC-Spade.

4. CONCLUSIONS

In this paper we improve the Multidimensional Sequential Pattern Algorithm. We improve the algorithm with the use of Prefixspan algorithm instead of Spade algorithm. Because of Prefixspan algorithm is projection based algorithm. From our experiment we saw the comparison between Spade and Prefixspan, from that we conclude that Frequent pattern are generated from Prefixspan is better than Spade. So we use Prefixspan algorithm with BUC based algorithm instead Spade algorithm. There are many frequent pattern algorithm are available. For generate frequent pattern we can use any classical algorithm for Future work.

REFERENCES

[1] H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal. Multi-dimensional sequential pattern mining. In ACM CIKM, pages 81.88, 2001.
[2] C.-C. Yu and Y.-L. Chen. Mining sequential patterns from multidimensional sequence data. IEEE Transactions on Knowledge and Data Engineering, 17(1):136.140, 2005.
[3]M.Plantevit,Y.W.Choong,A.Laurent,D.Laurent,and M.Teisseire. M2SP: Mining sequential pattern among several dimentions. In PKDD , page 205-216, 2005.

[4] M.Plantevit,Y.W.Choong,A.Laurent,D.Laurent,and M.Teisseire. HYPE: Mining Hierarchical sequential pattern ACM, 2006.
[5] M. Plantevit, A. Laurent, D. Laurent, M. Teisseire, and Y. W. Choong, "Mining multidimensional and multilevel sequential patterns," TKDD, vol. 4, no. 1, pp. 1–37, 2010.
[6] A Servey on Sequential pattern mining algorithms by Vishal S. Motegaonkar, Prof. Madhav V. Vaidya, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2486-2492.
[7] V.Chandra Shekhar Raoand P.Sammulal, "SurveyOnSequential Pattern Mining Algorithms". International Journal of computer application(0975-8887),Vol 76-No.12, August 2013.
[8] M.Garofalakis, R.Rastogi, and K.Shim, "SPIRIT: Sequential pattern mining with regular expression constraints",VLDB'99,1999
[9] NIZAR R. MABROUKEH and C. I. EZEIFE, A Taxonomy of Sequential Pattern Mining Algorithms, ACM Computing Surveys, Vol. 43, No. 1, Article 3, Publication date: November 2010.
[10] ZAKI, M. J. 2001. SPADE: an efficient algorithm for mining frequent sequences. Machine Learning Journal, Special issue on Unsupervised Learning 42, 1/2, 31–60.
[11] PEI, J., HAN, J.,MORTAZAVI-ASL, B.,WANG, J., PINTO, H., CHEN, Q., DAYAL, U., AND HSU, M.-C. 2004. Mining sequential patterns by pattern-growth: The prefixspan approach. IEEE Transactions on Knowledge and Data Engineering 16, 11, 1424–1440.
[12] Sequential Pattern Mining: Survey and Current Research Challenges by Chetna Chand, Amit Thakkar, Amit Ganatra, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012.
[13]Tan,Steinbach,Kumar,"IntroductiontoDatamining",http://www.users.cs.umn.edu/~kumar/dmbook/dmslides/chap6_basic_association_analysis.pdf
[14] Hilderman R. J., Hamilton H. J.,"Knowledge Discovery and Interest Measures",In: Kluwer Academic Publishers, Boston, 2002.
[15] Sequential Pattern Mining: A Comparison between GSP, SPADE and Prefix SPAN by 1Manika Verma, 2Dr. Devarshi Mehta, © 2014 IJEDR | Volume 2, Issue 3 | ISSN: 2321-9939.
[16] Object-oriented programming "The History of Java Technology". Sun Developer Network. c. 1995. Retrieved 2010-04-30.
[17] Design Goals of the Java™ Programming Language". Oracle. 1999-01-01. Retrieved 2013-01-14.