Information Technology Journal, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.

Mining Web Frequent Multi-dimensional Sequential Patterns

Guoyan Huang, Na Zuo and Jiadong Ren
College of Information Science and Engineering, Yanshan University,
Qinhuangdao 066004, People's Republic China

Abstract: Although, numerous methods have been proposed to mine sequential patterns, previous approaches can't effectively find web frequent multi-dimensional sequential patterns from d-dimensional sequence data with multi-dimensional information, where d>2. The main objective of web frequent multi-dimensional sequential pattern mining is to provide the end user with more useful and interesting patterns. To mine web frequent multi-dimensional sequential patterns, in present study, we propose a new algorithm ExtSeq-MIDim. It employs extseq (Extended sequential pattern mining method) to mine sequential patterns from d-dimensional sequence data, then forms projected multi-dimensional database for each sequential pattern and uses an algorithm MIDim (Memory Indexing for mining multi-dimensional pattern) to mine multi-dimensional patterns within projected databases. During the multi-dimensional pattern mining process, MIDim takes advantage of the idea of memory indexing without multiple scanning projected databases and handles fewer and shorter multi-dimensional tuples as the discovered patterns get longer. The experimental results show that ExtSeq-MIDim scales up linearly and is efficient to find web multi-dimensional sequential patterns.

Key words: Multi-dimensional sequential pattern, projected database, prefixMDSpan, memory indexing

INTRODUCTION

Sequential pattern mining (Agrawal and Srikant, 1995) which aims at discovery of useful sequential patterns in a mass of sequences, is a hot issue in data mining. It has far-ranging applications, such as mining web logs, customer purchase behavior analysis and disease diagnosis.

Many studies have contributed to mine sequential patterns. The typical Apriori-based method such as GSP (Srikant and Agrawal, 1996) applies a multiple-pass, candidate-generation-and-test approach to mine sequential patterns. Although, this method reduces the search space, it bears two inherent costs, multiple database scanning and candidate generation. To overcome these problems, projected-based pattern growth method, PrefixSpan is proposed by Pei *et al.* (2004). PrefixSpan explores a divide-and-conquer strategy and projects database into small projected databases. For further broad applications, Pei *et al.* (2007) pushed constraint into sequential pattern mining under pattern growth methodology. Chang (2011) put forward TiWS to use pattern-growth principle to mine weighted sequential patterns in a sequence database with a time-interval weight. Lin *et al.* (2008a) proposed CTSP, this algorithm loads database into memory and then constructs time-indexes to facilitate both pattern mining and closure checking within the pattern-growth framework. However, these projected-based pattern growth methods generate multiple intermediate databases. For this reason (Lin and Lee, 2005) came up with memory indexing-based sequential pattern mining method MEMISP which fast discovers sequential patterns through memory indexing and database partitioning without multiple scanning databases and generating projected databases. Then Lin *et al.* (2008b) presented METISP to use effective time-indexing and a pattern-growth strategy to find time constraint sequential patterns rapidly in large databases.

The above-mentioned methods only mine sequential patterns. Nevertheless, in real life, sequential patterns are associated with interesting multi-dimensional information, such as the customer's group, age and region; the previous sequential pattern mining methods can't effectively deal with it. Hence, Pinto *et al.* (2001), proposed efficient Seq-Dim algorithm for mining multi-dimensional sequential patterns. These patterns will be more consistent with business needs and more useful and interesting. Although, Seq-Dim is efficient and scalable to mine multidimensional patterns in high dimension space, this method requires multiple scanning projected databases to mine multidimensional patterns, as the

Corresponding Author: Guoyan Huang, College of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, People's Republic China

amount of data increases and the dimensionality becomes higher, time cost will be great.

In the real world, there exists some multi-dimensional sequence data, for example, web logs of an on-line store on Taobao or eBay by data preprocessing. Mining sequential pattern in multi-dimensional sequence data was first put forward by Yu and Chen (2005). If multidimensional sequence data merges the relevance multidimensional information into a web multidimensional sequence database, mining patterns from such database can help the marketing manager to analyze customer purchase behavior and to embellish web design so as to meet consumers access pattern.

To reduce the times of scanning projected databases when mining multidimensional patterns as much as possible and to mine web frequent multidimensional sequential patterns effectively, we proposes ExtSeq-MIDim. This algorithm explores the PrefixMDSpan to mine sequential patterns from multi-dimensional sequence data, then forms projected multi-dimensional database for each sequential pattern and uses our algorithm MIDim to mine multidimensional patterns within projected databases. MIDim which scans projected database only one time, makes the best of prefix-index technique for focused searching and finds MD-patterns rapidly.

PROBLEM DEFINITIONS

Let a set of records <Cid; A_1, A_2,…, A_m; S> be a multidimensional sequence database (DB), where Cid is a customer id (A_1, A_2,…, A_m) is a set of dimensions and S is a web sequence. Let * be a meta-symbol which does not belong to any domain of (A_1, A_2,…,A_m). p = (a_1, a_2… a_m; s) is a multidimensional sequence (where $a_i \epsilon$ ($A_i \epsilon \cup \{*\}$) (1 = I = m), whose support is the number of sequences containing p in DB, denoted as sup (p). If sup (p) exceeds a given threshold min-sup, p is a multi-dimensional sequential pattern and (a_1,a_2… a_m) is a multi-dimensional pattern (abbreviated as MD-pattern).

S is a web sequence in multidimensional sequence data. Let I = {i_1, i_2, i_3… i_n} be a set of items. A 1- dimensional sequence *S* is an ordered list <s_1, s_2,…,s_r>l, where $s_i \epsilon \hat{o} I$ for (1<i<r). An n-dimensional sequence (n>1) denoted as <s_1, s_2,…,s_r>n is an ordered list of (n-1)-dimensional sequence, where n is the number of dimensions and s_i is a (n-1)-dimensional sequence for (1< i <r).

Example 1: Given a web multidimensional sequence database DB in Table 1, the DB has three visit records of

Table 1: A web multidimensional sequence database

Cid	Cust-grp	City	Age-grp	Web-seq
1	business	Chicago	middle	<<<ab>$_1$<dc>$_1$>$_2$<<df >$_1$<bc>$_1$>$_2$>$_3$
2	profession	Chicago	retired	<<<ab>$_1$>$_2$<<c>$_1$<be>$_1$>$_2$<<dg>$_1$<hf>$_1$ >$_2$>$_3$
3	business	Chicago	middle	<<<bc>$_1$<ac>$_1$>$_2$<<fh>$_1$<ab>$_1$>$_2$>$_3$

customer. MDB which is composed of 2,3,4 column, contains three dimensions, Cust-grp, City and Age-grp. Web-seq <<<ab>$_1$<dc>$_1$>$_2$ <<df >$_1$<bc>$_1$>$_2$>$_3$ represents a customer visits web pages a and b in the first session and visits pages d and c sequentially in the second session on the first day; then he/she visits web pages d and f in the first session and visits pages b and c successively in the second session on the second day. The subscripts in web-seq denote the dimension numbers and dimensions 1, 2 and 3 indicate visited pages, sessions and days, respectively. <<<ab>$_1$<dc>$_1$>$_2$<<df>$_1$<bc>$_1$>$_2$>$_3$ is a 3-dimensional sequence, whose element <<ab>$_1$<dc>$_1$>2 is a 2-dimensional sequence and <ab>$_1$ is a 1-dimensional sequence. Given min-sup = 2 (*,Chicago,*, <ab>$_1$) is a web multi-dimensional sequential pattern in DB. Since there are two tuples 1 and 2 support the pattern in DB.

Definition 1: (prefix MD-pattern) Given a MD-pattern ρ and a frequent multi-dimensional value x, a new MD-pattern ρ' can be formed by appending x to the corresponding dimension in ρ; ρ is the prefix MD-pattern of ρ'.

Example 2: If a MD-pattern is (business,*,*) and the frequent multi-dimensional value is Chicago, we can obtain the new MD-pattern (business,Chicago,*) by appending Chicago to the corresponding dimension in (business,*,*) (business,*,*) is a prefix MD-pattern.

Definition 2: (ρ-index) ρ-index is a set of (p_t, pos) pairs, where p_t is a pointer to the multi-dimensional tuple t that contains MD-pattern ρ and pos in t is the occurring position of the last frequent multi-dimensional value x in ρ and represents the dimension number of multidimensional value x.

Example 3: As shown in Fig. 1 (business,*,*)-index is a set of (p_t, pos) pairs, where p_t are pointers in index set to the multi-dimensional tuples 1 and 3 that contains MD-pattern (business,*,*) and the first pos in tuples 1 and 3 is the occurring position of the last frequent multi-dimensional value business in MD-pattern (business,*,*) and represents the dimension number of multidimensional

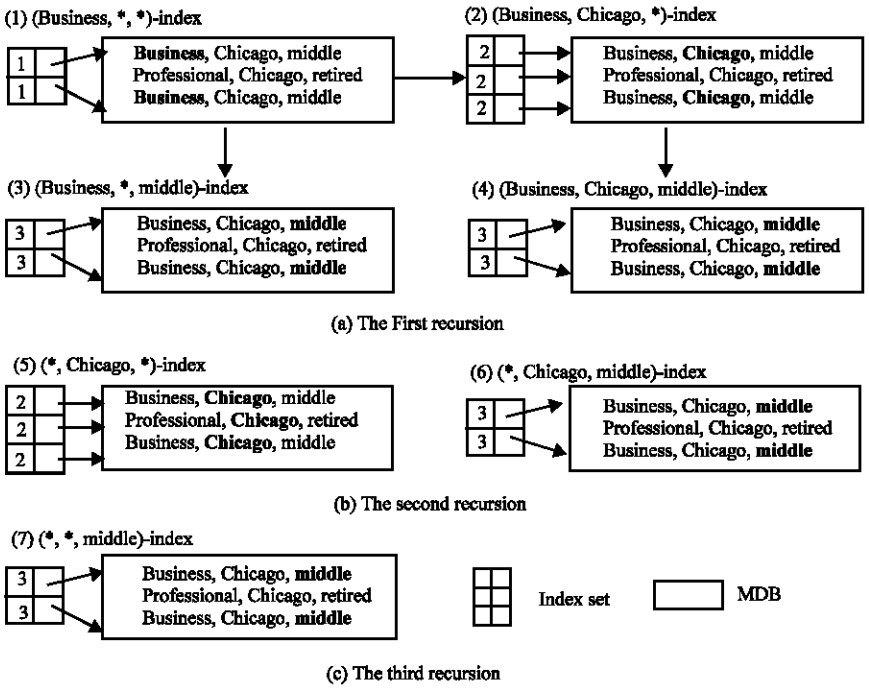

Fig. 1: Find the MD-patterns with MIDim algorithm

value business in MDB. Since business belongs to the first dimension Cust-grp in MDB, we denote the dimension number of business is 1.

EXTSEQ-MIDIM ALGORITHM

ExtSeq-MIDim algorithm mines sequential patterns by PrefixMDSpan in multidimensional sequence data at first and then builds up projected multidimensional database for each sequential pattern. Second, we scan the projected database once and find 1-frequent MD-pattern, then construct index for each MD-pattern. By a prefix-index method and the pattern-growth theory, we use MIDim to mine MD-patterns in projected database. Finally, by combing the sequential pattern and the corresponding MD-pattern, multi-dimensional sequential pattern can be obtained.

PrefixMDSpan: pattern-growth principle for mining sequential patterns: Take the three web-seqs in Table 1 for example. There are two different representations of web-seqs: the standard format and the simplified format, as shown in Table 2. The standard format is used to define web-seqs, the simplified format is used to design PrefixMDSpan algorithm. So, the standard format of web-seqs needs to be transformed into the simplified format before mining web-seq database. This conversion is based on the definition of the scope relation DS (S, k, j) between any two items S (k) and S (j) in sequence S.

Table 2: A sequence database represented in both standard and simplified formats

Cid	The standard format (web-seq)	The simplified format (web-seq)
1	<<<ab>$_1$<dc>$_1$>$_2$<<df >$_1$<bc >$_1$>$_2$>$_3$	<(a1)(b1)(d2)(c1)(d3)(f1)(b2)(c1) >$_3$
2	<<<ab>$_1$>$_2$<<c>$_1$<be>$_1$>$_2$<< dg>$_1$<hf>$_1$>$_2$>$_3$	<(a1)(b1)(c3)(b2)(e1)(d3)(g1)(h2) (f1)>$_3$
3	<<<bc>$_1$<ac>$_1$>$_2$<<fh>$_1$<ab>$_1$ >$_2$>$_3$	<(b1)(c1)(a2)(c1)(f3)(h1)(a2) (b1)>$_3$

Given an n-dimensional sequence S, for any two items S (k) and S (j) in S (k and j denote the ith position in web-seq), let s be the I-dimensional element of S that contains S (k) and S (j) in its different elements and no u-dimensional element of S, where u<I. Then, the dimensional scope of S (k) and S (j) is DS (S, k, j) = I. Take <<<ab>$_1$<dc>$_1$>$_2$<<df >$_1$<bc>$_1$>$_2$>$_3$ as an example, DS (*S*,2,3) = 2.

PrefixMDSpan first finds frequent items after scanning the transformed sequence database only once. According to the frequent items, the database is projected into several smaller databases by pattern-growth principle. All web sequential patterns are found by recursively growing longer patterns from shorter ones. Specially, PrefixMDSpan extends Prefixspan to find

frequent items in projected database by constructing matrix M, where each column corresponds to an item and each row corresponds to a dimensional scope value, each cell in the matrix records the number of sequences in the projected database containing the item and the dimensional scope of this item and the last element of prefix.

MIDim: Memory indexing for mining multi-dimensional patterns: After mining sequential patterns from multidimensional sequence data by PrefixMDSpan, we begin to construct projected multidimensional database for each sequential pattern and use our MIDim algorithm to mine MD-patterns in projected database.

MIDim algorithm scans only one time over projected database and reads all multi-dimensional tuples into memory during the whole mining process, then applies prefix-index and projected MDB for the MD-pattern mining. By the pattern-growth theory, this algorithm discovers all MD-patterns with larger size recursively. The MIDim algorithm is outlined as follows.

The MIDim algorithm discovers all MD-patterns by the following steps:

- **Step 1:** Read MDB into memory, scan projection database once and find frequent multi-dimensional values, output the matching prefix MD-pattern for each frequent multidimensional value, then construct indexing for each prefix MD-pattern
- **Step 2:** Use index set and the projection database to seek locally frequent multi-dimensional values with respect to current prefix MD-pattern. Append frequent multidimensional values on current prefix MD-pattern to form long prefix MD-patterns, output the long prefix MD-patterns and build up indexing for them, respectively
- **Step 3:** Carry out Step 2 recursively. When there are no frequent multi-dimensional values in the index set of current prefix MD-pattern, output the pattern

Example 4: Given the database DB in Table 1, where the 2,3,4 column of DB forms MDB. Given min-sup = 2. We can see all records support sequential patterns $<b>_1$, so all multi-dimensional tuples in MDB consists of

Algorithm: MIDim
Input: projected database, min-sup
Output: All MD-patterns
Method:

1. Scan projected database into memory and find all frequent multi-dimensional values L, let F-list "x_1£–...£–x_n" (n=|L|) be a list of frequent multi-dimensional values;
2. for i=1 to n do{
3. output the matching prefix MD-pattern ρ_i of x_i;
4. then construct ρ_i-index for prefix MD-pattern ρ_i;
5. for prefix MD-pattern ρ_i do{
6. call MineIndexset(ρ_i , ρ_i-index);
7. delete index set of ρ_i;}}

Algorithm: MineIndexset(ρ, ρ-index)
Input: ρ=a prefix MD-pattern; ρ-index= an index set of ρ
Output: the set of new prefix MD-pattern
Method:

1. for each pair (p_t,pos) in ρ-index do{
2. count the occurrence for each locally multidimensional value x from position (pos+1) to |ds| in tuple t;/*|ds| is the length of multidimensional tuple t.*/}
3. for each x do{
4. if(x's occurrence times>=min-sup){
5. output the new prefix MD-pattern ρ' formed by ρ and x;
6. insert a pair (p_t,x_pos) into the index set of the new prefix MD-pattern ρ';/* x_pos is the occurrence position of x in t*/}
7. else delete x;}}
8. if(index set of the new prefix MD-pattern is empty)
9. output ρ';
10. else
11. for each new prefix MD-pattern ρ' do{
12. call MineIndexset(ρ', ρ'-index);}

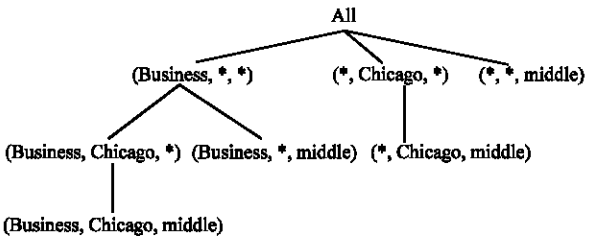

Fig. 2: Frequent MD-patterns tree

<b>$_1$- projection database, from which we mine MD-patterns. The mining process is shown in Fig. 1.

MIDim =reads projected database into memory and scan the projected database once to find all frequent multi-dimensional values. F-list {business, Chicago, middle} is obtained. Since there are three multi-dimensional values in F-list, MIDim carries out three times recursive calls.

In view of the divide-and-conquer strategy, the mined frequent MD-patterns can be divided into three subsets: (1) the subset containing (business,*,*); (2) the subset containing (*,Chicago,*) but not containing (business,*,*); (3) the subset containing (*,*,middle) but not containing (business,*,*) (*,Chicago,*). The classified result is shown in Fig. 2.

EXPERIMENTS AND PERFORMANCE EVALUATION

To evaluate the ExtSeq-MIDim performance, we use different synthetic datasets and give parameters in Table 3. Experimental dataset is generated by IBM Data Generator through adding dimensional information. Dimensional information is generated randomly so that values are distributed evenly in every dimension.

All the experiments are performed on a PC with Pentium (R) Dual-core 2.60 GHz CPU and 2.00 GB main memory, in the environment of Windows XP. We implement the programs in C++.

Execution time of ExtSeq-MIDim on different synthetic datasets: Here, the experiments are to test the running times for different thresholds and different dimensions over three different synthetic datasets. We set 10 dimensions in the first experiment and set the minimum support 20% in the second experiment and set the cardinality of each dimension 10 in the two experiments. The results are in Fig. 3a and b which show the efficiency of ExtSeq-MIDim over various synthetic datasets. When the support threshold falls or the number of dimensions rises, the number as well the length of MD-patterns increase obviously, meanwhile, the runtime increases gradually.Since ExtSeq-MIDim utilizes efficient PrefixMDSpan to mine sequential patterns and only scans projected database one time and alters the relevant

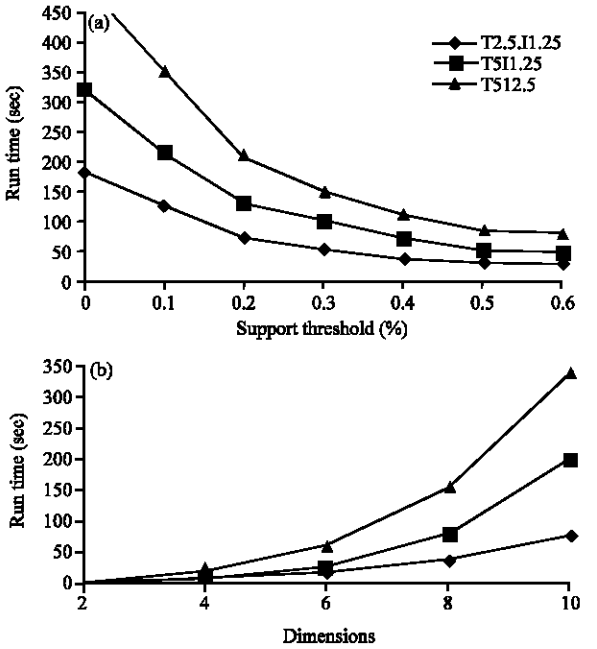

Fig. 3(a-b): Execution time of ExtSeq-Dim on different synthetic datasets. (a) Runtime over different thresholds and (b) Runtime over different dimensions

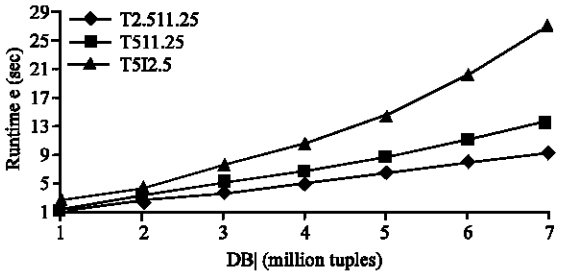

Fig. 4: Scalability of ExtSeq-MIDim on different synthetic datasets

Table 3: Parameters used in the experiment

Parameter	Description	Value
|D|	Number of customers	25K
|C|	Average number of visit records per Customer	10
|T|	Average number of items per visit record	2.5,5
|S|	Average length of maximal potentially large sequences	4
|I|	Average size of Itemsets in maximal potentially large sequences	1.25,2.5

indexing when mining MD-patterns, ExtSeq-MIDim is efficient to find web multi-dimensional sequential patterns.

Scale-up: To verify the scalability of ExtSeq-MIDim, we increase the number of |DB| from 1000 to 7000 K with different synthetic datasets and set the minimum support 25% in the test, the dimensionality and cardinality of each dimension are set to 5 and 10. Figure 4 shows ExtSeq-

MIDim is scalable with the number of |DB| on various synthetic datasets, so ExtSeq-MIDim has good scalability.

CONCLUSIONS

Present study, we propose a new algorithm ExtSeq-MIDim for mining web frequent multi-dimensional sequential patterns. The algorithm contains two steps: the sequential pattern mining and the multidimensional pattern mining. In the sequential pattern mining process, ExtSeq-MIDim employs PrefixMDSpan to mine sequential patterns from multidimensional sequence data. PrefixMDSpan uses pattern-growth principle in sequential pattern mining and leads to its high execution efficiency. During the multidimensional pattern mining process, we apply MIDim to mine MD-patterns in projected multidimensional database. MIDim, firstly scans projected MDB once to load it into memory, finds all frequent multidimensional values and outputs matched MD-pattern; then this algorithm builds index set of each prefix MD-pattern, finds local frequent multi-dimensional values from index set and grows discovered patterns; at last the algorithm recursively constructs index set of the detected pattern and discovers all MD-patterns. The performance study shows that ExtSeq-MIDim has good scalability and is efficient in finding web multi-dimensional sequential patterns over web multidimensional sequence database.

ACKNOWLEDGMENTS

Present study is supported by the Natural Science Foundation of Hebei Province P.R. China No. F2009000477. We also feel grateful for the helpful comments and suggestions of the experts.

REFERENCES

Agrawal, R. and R. Srikant, 1995. Mining sequential patterns. Proceedings of the 11th International Conference on Data Engineering, March 6-10, Taipei, Taiwan, pp: 3-14.

Chang, J.H., 2011. Mining weighted sequential patterns in a sequence database with a time-interval weight. Knowledge-Based Syst., 24: 1-9.

Lin, M. and S. Lee, 2005. Fast discovery of sequential patterns through memory indexing and database partitioning. J. Inform. Sci. Eng., 21: 109-128.

Lin, M.Y., S.C. Hsueh and C.W. Chang, 2008a. Fast discovery of sequential patterns in large databases using effective time-indexing. Inform. Sci., 178: 4228-4245.

Lin, M.Y., S.C. Hsueh and C.W. Chang, 2008b. Mining closed sequential patterns with time constraints. J. Inform. Sci. Eng., 24: 33-46.

Pei, J., J. Han and W. Wang, 2007. Constraint-based sequential pattern mining: The pattern-growth methods. J. Intell. Inform. Syst., 28: 133-160.

Pei, J., J. Han, B. Mortazavi-Asl, J. Wang and H. Pinto *et al*., 2004. Mining sequential patterns by pattern-growth: The prefixspan approch. IEEE Trans. Knowledge Data Eng., 16: 1424-1440.

Pinto, H., J. Han, J. Pei, K. Wang, Q. Chen and U. Dayal, 2001. Multi-dimensional sequential pattern mining. Proceedings of the 10th International Conference on Information and Knowledge Management, Oct. 5-10, Atlanta, GA., pp: 81-88.

Srikant, R. and R. Agrawal, 1996. Mining sequential patterns: Generalizations and performance improvements. Proc. Int. Conf. Extend. Database Technol., 1057: 3-17.

Yu, C.C. and Y.L. Chen, 2005. Mining sequential patterns from multidimensional sequence data. IEEE Trans. Knowledge Data Eng., 17: 136-140.