Exploring multi-dimensional sequential patterns across multi-dimensional multi-sequence databases

Guoyan Huang, Na Zuo, Lan Bai, Jiadong Ren

1College of Information Science and Engineering, Yanshan University
2Hebei Vocational College of Foreign Languages
Qinhuangdao, P.R.China, 066004
hgy@ysu.edu.cn, zuona622@163.com, bailanwq@163.com, jdren@ysu.edu.cn

Abstract

Existing multi-dimensional sequential pattern mining methods only discover multi-dimensional sequential patterns in databases involving one sequential dimension. Since multi-dimensional sequential patterns may exist in databases containing more than one sequential dimension, in this paper, we present algorithm PSeq-MIDim for mining multi-dimensional sequential patterns from multiple sequential dimensions with multi-dimensional information, which makes up multi-dimensional multi-sequence databases. PSeq-MIDim applies PSeq to mine sequential patterns from multiple sequential dimensions, then forms the corresponding projected multi-dimensional database for each sequential pattern, and uses MIDim to mine multi-dimensional patterns within projected databases. PSeq performs sequential pattern mining from one sequential dimension, and then propagates the mined sequential patterns to other sequential dimension. Furthermore, the mined sequential patterns are represented as a lattice structure to provide guidelines for mining sequential patterns across multiple sequential dimensions. MIDim, which scans projected database only one time, makes the best of prefix-index technique for focused searching and finds multi-dimensional patterns rapidly. The experimental results show that PSeq-MIDim is efficient to find multi-dimensional sequential patterns from multi-dimensional multi-sequence databases.

Keywords: multi-dimensional sequential pattern, sequential pattern mining, lattice structure, prefix-index

1. Introduction

Sequential pattern mining [1-4], which aims at discovery frequent subsequences that appear in a mass of sequences, has attracted a considerable amount of research efforts recently, including customer purchase behavior analysis, web access patterns, medical diagnosis, and so on.

Many studies have been done in the field. M.Y. Lin presented METISP [5] to find time constraint sequential patterns rapidly in large databases by effective time-indexing and a pattern-growth strategy. H Liu et al came up with TD-Seq [6], which exploits a top-down transposition-based searching strategy as well as a new support counting method, for mining sequential patterns from high-dimensional sequence databases. T Hong et al put forward a novel incremental mining algorithm for maintaining sequential patterns, which is based on the concept of pre-large sequences to reduce the need for rescanning original databases [7]. But these sequential pattern mining approaches only mine sequential patterns in sequence database, which are not sufficient. Since, in real life, one would like to discover multi-dimensional sequential patterns across sequence database adding interesting multi-dimensional information, such as the customer’s group, age and region. These patterns will be more consistent with business needs and more useful and interesting. For example, web access sequences of an on-line store on Taobao or eBay merging the relevance multidimensional information forms a multi-dimensional sequence database, mining patterns from which can help the marketing manager to analyze customer purchase behavior and to embellish web design so as to meet consumer's access pattern. Thus it is necessary to introduce multi-dimensional information to enrich sequential patterns.

Much work has been devoted to mine multi-dimensional sequential pattern [8-14]. H Pinto and J Han et al firstly proposed an efficient method Seq-Dim for mining multi-dimensional sequential patterns [8]. The Seq-Dim mines sequential patterns in database by using PrefixSpan algorithm [9] at first, follows to find multi-dimensional patterns associated with mined sequential patterns by using
Exploring multi-dimensional sequential patterns across multi-dimensional multi-sequence databases
Guoyan Huang, Na Zuo, Lan Bai, Jiadong Ren


For further study, P.Songram et al put forward closed multi-dimensional sequential pattern mining [12], which is an integration of closed sequential pattern mining and closed itemset pattern mining. Y.Jin and L.W.Zuo came up with TDCL-SB [13] for incremental discovery of multi-dimensional sequential patterns, which integrates sequential pattern mining and itemset pattern mining with a uniform data structure and makes the mining process more efficient. However, these previous multi-dimensional sequential pattern mining algorithms only discover multi-dimensional sequential patterns in databases involving one sequential dimension.

Multi-dimensional sequential patterns may exist in databases containing more than one sequential dimension. Consider an online shopping environment, where customers may have three activities (i.e., visiting web pages, buying goods, and credit payment) via terminal, these activities closely occur together and each activity is referred to one sequential dimension in this paper. These sequential dimensions with customer category or other relevance multi-dimensional information can form a multi-dimensional multi-sequence database, mining patterns from such database captures the cross-relationship among multiple sequential dimensions and discovers classified patterns, which can yield significant information and reveal more knowledge.

To mine multi-dimensional sequential patterns from multi-dimensional multi-sequence databases effectively, we proposes PSeq-MIDim (Propagated sequential pattern mining method- Memory Indexing for mining multi-dimensional pattern). This algorithm explores PSeq to mine sequential patterns from multiple sequential dimensions at first; and then forms the corresponding projected multi-dimensional database for each sequential pattern; finally MIDim is used to mine multi-dimensional sequential patterns within projected databases. In Sections 2, we define the problem. Section 3 presents PSeq-MIDim algorithm. The experimental results are shown in section 4. Section 5 concludes the paper finally.

2. Problem definitions

Assume that each sequential dimension has its own set of items. The problem of mining multi-dimensional sequential patterns is that given a multi-dimensional multi-sequence database, we aim at discovering multi-dimensional sequential patterns that consists of co-occurred events among sequential dimensions and the relevance multi-dimensional information. Table 1 shows a multi-dimensional multi-sequence database, where the corresponding events of two sequential dimensions in one tuple closely occur together. For example, in tuple 1, event (a) and event (1,2) closely occur together, likewise event (b,c) and event (2,3), event (b,c,d) and event (6), event (e) and event (4,5).

<table>
<thead>
<tr>
<th>Cid</th>
<th>Cust-grp</th>
<th>City</th>
<th>Age-grp</th>
<th>web access sequences</th>
<th>Purchase records</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>business</td>
<td>Boston</td>
<td>middle</td>
<td>&lt;(a,b,c)(b,c,d,e)&gt;</td>
<td>&lt;(1,2)(3,6)(4,5)&gt;</td>
</tr>
<tr>
<td>2</td>
<td>professional</td>
<td>Chicago</td>
<td>young</td>
<td>&lt;(a,b)(b,c)(c,e)&gt;</td>
<td>&lt;(1,3)(2,4)(8)&gt;</td>
</tr>
<tr>
<td>3</td>
<td>business</td>
<td>Chicago</td>
<td>middle</td>
<td>&lt;(a,e)(h)(g,j)&gt;</td>
<td>&lt;(1,6)(5)(9,10)&gt;</td>
</tr>
<tr>
<td>4</td>
<td>education</td>
<td>New York</td>
<td>retired</td>
<td>&lt;(a,b,f)(d,b,c,e)&gt;</td>
<td>&lt;(1,2,5)(7)(2,3)(4,5,6)&gt;</td>
</tr>
</tbody>
</table>

A multi-dimensional multi-sequence database (MMSDB) is of schema (RID, A1, A2, ..., Am, S1, S2, ..., Sn), where RID is a primary key, (A1, A2, ..., Am) is a set of dimensions and (S1, S2, ..., Sn) is a set of sequential dimensions. Let * be a meta-symbol which does not belong to any domain of A1, A2, ..., Am, p=(a1, a2, ..., am, s1, s2, ..., sn) is a multi-dimensional sequence (where a1 \in (A1 U {*}) (1 \leq i \leq m)) and s1(1 \leq i \leq n) is a sequence of the jth sequential dimension), whose support is the count of sequences containing p in MMSDB, denoted as sup(p). If sup(p) exceeds a given threshold min-sup, p is a multi-dimensional sequential pattern and (a1, a2, ..., am) is a multi-dimensional pattern (abbreviated as MD-pattern).

(S1, S2, ..., Sn) is a set of sequential dimensions. sj(1 \leq j \leq n), which is a sequence, is expressed by <X1,j, X2,j, ..., Xl,j>, where Xk,j is the kth element of sequence sj, and l is the number of elements of sj. Therefore,
a sequence across multiple sequential dimensions is represent as \( M = (s_1,s_2,\ldots,s_n) \) and is further denoted as \( M = (X_{11},X_{12},\ldots,X_{1l}, \ldots, X_{n1},X_{n2},\ldots,X_{nl}) \), where the same position’s element of all sequential dimensions occurs within the same time window.

**Definition 1. (Containing relation)** Suppose that we have two sequences across multiple sequential dimensions \( M = (X_{11},X_{12},\ldots,X_{1l}, \ldots, X_{n1},X_{n2},\ldots,X_{nl}) \) and \( N = (Y_{11},Y_{12},\ldots,Y_{l1}, \ldots, Y_{n1},Y_{n2},\ldots,Y_{nl}) \), where \( 1 \leq b \leq n \). \( M \) is contained by \( N \), denoted as \( M \subseteq N \), if and only if there exists an same integer list \( K(M,N) \) in each sequential dimension, denoted as \( k_1,k_2,\ldots,k_r \), such that 1 \( \leq k_1 \leq k_2 \leq \ldots \leq k_r \leq n \) and \( X_{ij} \subseteq Y_{ij} \), where \( 1 \leq i \leq l, 1 \leq j \leq r \).

**Example 1:** Assume that \( M = (\langle a \rangle, \langle b,c \rangle, \langle 2 \rangle, \langle 6 \rangle) \) and \( N = (\langle a \rangle, \langle b,c,d \rangle, \langle e \rangle, \langle 1,2 \rangle, \langle 2,3 \rangle, \langle 6 \rangle, \langle 4 \rangle, \langle 5 \rangle) \). It can be verified that \( M \) is contained by \( N \) since there exist integer list \( K(M,N) = \{1,3\} \) in each sequential dimension such that \( 1 \leq s_1 \leq 3 \) and \( a \subseteq (a), (b,c) \subseteq (b,c,d), (2) \subseteq (1,2) \) and \( (6) \subseteq (6) \).

### 3. PSeq-MIDim algorithm

The schema of a MMSDB is \((R_1,A_1,A_2,\ldots,A_m,S_1,S_2,\ldots,S_n)\), and every tuple \( t = (x_1,x_2,\ldots,x_m,s_1,s_2,\ldots,s_n) \) in MMSDB consists of two parts: sequence across multiple sequential dimensions \((s_1,s_2,\ldots,s_n)\) and multi-dimensional information \((x_1,x_2,\ldots,x_m)\). Therefore, sequential patterns and MD-patterns can be mined respectively.

PSeq-MIDim algorithm mines sequential patterns from multiple sequential dimensions by PSeq at first, which utilizes previous sequential pattern mining algorithms to discover sequential patterns in a starting dimension, and uses a lattice structure to store these patterns, then propagates the mined sequential patterns to other sequential dimension. After that, for each sequential pattern, we collect all multi-dimensional tuples in database containing the sequential pattern, and build up the corresponding projected multi-dimensional database. Then, 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.

We give our algorithm PSeq in section 3.1 and introduce previous proposed method MIDim in section 3.2 respectively.

### 3.1. PSeq: propagating sequential patterns among multiple sequential dimensions

If we treat a sequence across multiple sequential dimensions above as a single sequence, for example, \( \langle a \rangle, \langle b,c,d \rangle, \langle e \rangle, \langle 1,2 \rangle, \langle 2,3 \rangle, \langle 6 \rangle, \langle 4 \rangle, \langle 5 \rangle \rangle \) in customer 1 are joined as one sequence \( \langle a \rangle, \langle b,c,d \rangle, \langle e \rangle, \langle 1,2 \rangle, \langle 2,3 \rangle, \langle 6 \rangle, \langle 4 \rangle, \langle 5 \rangle \rangle \). Then, existing sequential pattern mining algorithms can be utilized to mine sequential patterns. Clearly, with sequences from multiple sequential dimensions, a sequence is the same integer list \( K(M,N) \) in each sequential dimension that \( 1 \leq s \leq 3 \) and \( a \subseteq (a), (b,c) \subseteq (b,c,d), (2) \subseteq (1,2) \) and \( (6) \subseteq (6) \).

Therefore, to reduce the generation of redundant patterns, and to further reduce the cost of mining sequential patterns in long sequences, PSeq is proposed. At first, some relevant definitions are given.

**Definition 2. (The position set)** Let \( M \) be a sequence in one sequential dimension. The position set of \( M \) is defined as \( P(M) = \{C_{id} : k_i, \ldots, C_{id} : k_i, \ldots, C_{id} : k_i\} \), where \( C_{id} \) is a customer id and \( k_i \) is the corresponding position list.

**Example 2:** Given a sequence in the first sequential dimension \( \langle a \rangle, \langle b \rangle, \langle c \rangle, \langle d \rangle \rangle \), it can be seen in table 1, the position set of \( \langle a \rangle, \langle b \rangle \rangle \) is \( P(S(\langle a \rangle, \langle b \rangle)) = \{C_{id} : 1,2, \ldots, C_{id} : 1,3, \ldots, C_{id} : 1,3\} \).

**Definition 3. (Propagated table)** Let \( M \) be a sequential pattern in one sequential dimension with position set \( P(S(M)) = \{C_{id} : k_i, \ldots, C_{id} : k_i, \ldots, C_{id} : k_i\} \). Assume that a sequence across multiple sequential dimensions is represent as \( N = (s_1,s_2,\ldots,s_n) \), where \( s_j = (X_{ij},x_{ij},X_{ij}) \). When propagating position set of \( M \) to sequential dimension \( s_j \), we would have a propagated table defined as \( S_{ij}[M = (X_{ij})] \) \( P(S(M)) = P(S(s_j)) \), which consists of sequences that co-occurred with \( M \).

**Example 3:** In Table 1, by propagating \( P(S(\langle a \rangle, \langle b \rangle)) \) to the next sequential dimension (Purchase records), we could have the propagated table \( Purchase \) records \( \| (\langle a \rangle, \langle b \rangle) \rangle \) shown in Table 2.
Definition 4. (Sequential patterns across multiple sequential dimensions) Given the propagated table $S||M$, which is also a sequence database, $(M, p)$ is a sequential pattern across multiple sequential dimensions if and only if $p$ is a sequential pattern in $S||M$ and the number of elements of $p$ is equal to that of $M$.

Example 4: Given Purchase records $$(a,b)$$ and min-sup=3, we can easily find that $$(1,2)$$ is a sequential pattern having the same number of elements as the propagated sequential pattern $$(a,b)$$, thus $$(a,b)$$ is a sequential pattern across two sequential dimensions.

Definition 5. (Combination operation) Let $M$ and $N$ be two sequences across two sequential dimensions. $PS(M) = \{(Cid_1 : k_1, k_2, ..., k_l), (Cid_2 : k_1, k_2, ..., k_l)\},$ and $PS(N) = \{(Cid_3 : l_1, l_2, ..., l_l), (Cid_4 : l_1, l_2, ..., l_l)\}$. The combination of $PS(M)$ and $PS(N)$ is denoted as $PS(M) \cup PS(N) = \{(Cid_1 : k_1, k_2, ..., k_l), (Cid_2 : k_1, k_2, ..., k_l), (Cid_3 : l_1, l_2, ..., l_l), (Cid_4 : l_1, l_2, ..., l_l)\}$, such that $Cid_{1,2} = Cid_1$ and $k_i < l_i$.

Example 5: Given $M = (a,b), (1,1)$, $N = (b,c), (1,2)$. It can be verified that $PS(a,b), (1,1)) = \{(Cid_1 : 1), (Cid_1 : 1)\}, PS(b,c), (1,2)) = \{(Cid_2 : 2), (Cid_2 : 2)\},$ and $PS(a,b), (1,1) \cup PS(b,c), (1,2) = \{(Cid_1 : 1,2), (Cid_2 : 1,2)\}$. Hence, $PS((a,b), (1,1)) \cup PS((b,c), (1,2)) = \{(Cid_1 : 1,2), (Cid_2 : 1,2)\}$.

Definition 6. (Atomic pattern) If the position set of a pattern $p$ is not contained in any other patterns, the pattern $p$ is called atomic pattern.

Example 6: Some patterns mined in propagated tables Purchase records $$(a)$$ and Purchase records $$(b)$$ are the same as patterns mined in propagated table Purchase records $$(a,b)$$, this is due to the position set of $$(a,b)$$ is contained in both position sets of $$(a)$$ and $$(b)$$. Consequently, sequences in Purchase records $$(a)$$ and Purchase records $$(a,b)$$ include some sequences in propagated table Purchase records $$(a,b)$$ and Purchase records $$(b)$$. Therefore, only sequential patterns with their length being one should be propagated to other domains. In other words, only position sets of the top-level nodes in lattice structures are propagated. These top-level nodes are referred to as atomic patterns.

### 3.1.1. PSeq Algorithm

Algorithm PSeq performs sequential pattern mining in one sequential dimension (referred to as the starting sequential dimension) and then propagates position sets of the mined sequential patterns to other sequential dimensions. However, to provide guidelines for mining sequential patterns across multiple sequential dimensions, the sequential patterns mined in the starting sequential dimension are represented as the lattice structure. By propagating position sets of these patterns, only those sequences with the same positions in multiple sequential dimensions are extracted, thereby the mining space is reduced in each sequential dimension. Algorithm PSeq iteratively propagates position sets of sequential patterns to the next sequential dimension until all sequential dimensions have been mined. The PSeq algorithm is given as follows.

<table>
<thead>
<tr>
<th>Algorithm: PSeq</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Sequence databases across n-sequential dimensions $S_1, S_2, ... , S_n$ and min-sup $\delta$.</td>
</tr>
<tr>
<td><strong>Output:</strong> sequential patterns across n-sequential dimensions.</td>
</tr>
<tr>
<td><strong>Method:</strong></td>
</tr>
<tr>
<td>1. Use sequential pattern mining on $S_1$;</td>
</tr>
<tr>
<td>2. Let $SP_1$ be the set of sequential patterns mined in $S_1$, and $SP_1$ is stored in lattice structure;</td>
</tr>
<tr>
<td>3. For each pattern $p$ pointed to by inter-dimension links of $Y$</td>
</tr>
<tr>
<td>4. If Support $(p, (a,b)) \geq T \text{ min}$</td>
</tr>
<tr>
<td>5. Build intra-dimension links from $p$ to $(a,b)$;</td>
</tr>
<tr>
<td>6. Build intra-dimension links from $p$ to $(a,b)$;</td>
</tr>
<tr>
<td>7. Append $(p, (a,b))$ to $SP_i$;</td>
</tr>
<tr>
<td>8. End</td>
</tr>
</tbody>
</table>

//Step 3
3. For each sequential dimension $S_i$, $i=2,3,\ldots,n$

4. For each $P \in S_{i-1}$

//Step 1

5. If $|P|=1$ // $P$ is a sequential pattern with its length being one

6. Build propagation table $S_i|P$, and seek frequent items in $S_i|P$ with min-sup $\delta$

7. Let $FI$ be the set of frequent items in $S_i|P$

8. For each $Q \in FI$

9. Append $(P,Q)$ to $S_{i-1}$, and Let $PS(P) = PS(P) \cap PS(Q)$

10. End

//Step 2

11. If $|P|=1$ // $P$ is a sequential pattern in each sequential dimension having only one element

12. Let $X$ and $Y$ be two patterns pointed to by intra-dimension links of $P$

13. For each pattern $\alpha$ pointed to by inter-dimension links of $X$

14. For each pattern $\beta$ pointed to by inter-dimension links of $Y$

15. If $\text{Support}(P,(\alpha)(\beta)) \geq \delta$

16. Then $\text{Build inter-dimension links from } P \text{ to } (\alpha)(\beta)$;

17. $\text{Build intra-dimension links from } (\alpha)(\beta) \text{ to } (\alpha) \text{ and } (\beta)$;

18. Append $((\alpha)(\beta))$ to $S_{i-1}$;

19. End

20. If $|P|>1$ // $P$ is a sequential pattern in each sequential dimension having more than one element

21. Let $X$ and $Y$ be two patterns pointed to by intra-dimension links of $P$

22. For each pattern $\alpha$ pointed to by inter-dimension links of $X$

23. For each pattern $\beta$ pointed to by inter-dimension links of $Y$

24. If $\text{Support}(P,(\alpha)(\beta)) \geq \delta$

25. Then $\text{Build inter-dimension links from } P \text{ to } (\alpha)(\beta)$;

26. $\text{Build intra-dimension links from } (\alpha)(\beta) \text{ to } (\alpha) \text{ and } (\beta)$;

27. Append $((\alpha)(\beta))$ to $S_{i-1}$;

28. End

29. Output $= S_P$

30. End

---

**Example 7:** Suppose that the starting sequential dimension is set to web access sequences in Table 1. Those mined sequential patterns are represented as a lattice structure shown in Fig. 1, where each node represents a sequential pattern and the linkages of nodes (standing for intra-dimension links) represent itemset relationships. Furthermore, those nodes with the same number of elements are further arranged level-by-level. Explicitly, it can be seen in Fig. 1 for the nodes with their number of elements is 1, these nodes are put level-by-level in increasing order of length of sequences.

![Figure 1. An example of lattice structures for sequential patterns in a starting sequential dimension](image)

The detailed steps for propagating the mined sequential patterns to other sequential dimension are described as follows:

**Step 1:** Obtain atomic patterns across two sequential dimensions through propagated table:

Sequential patterns mined in web access sequences have been represented as a lattice structure in Fig. 1. From the propagated table of each mined atomic pattern (i.e., the top-level nodes), we could derive atomic patterns in the second sequential dimension. Then the corresponding atomic patterns across two sequential dimensions are generated by propagating the position sets of atomic patterns in the first sequential dimension to the second sequential dimension. Specifically, in Fig. 2, for each atomic pattern in the first sequential dimension, there is an inter-dimension link representing that these two patterns are able to form sequential patterns across two sequential dimensions. Consequently, we have $((a),(1))$, $((b),(2))$, $((b),(3))$, $((c),(2))$, which are obviously atomic patterns.

**Step 2:** Obtain sequential patterns across two sequential dimensions through union operation, where sequential patterns in each sequential dimension have only one element.

Let $P = ((b,c))$, a sequential pattern with $|P| = 1$ in the first sequential dimension of Table 1. Through the intra-dimension links, atomic patterns that are components of $P$ (i.e., $(b)$ and $(c)$) can be found. In Fig. 3 following inter-dimension links of $(b)$ and $(c)$, we could obtain the atomic patterns in the second sequential dimension (i.e., $(2)$ and $(3)$). Consequently, two possible unions of $P$ are generated ($(i.e., (b),(2)) \cup (c),(2)) = ((b,c),(2))$ and $(b),(3)) \cup (c),(2)) = ((b,c),(2,3))$. Then the support values of these patterns are examined by checking their position sets. Given min-sup=3, the support values of $((b,c),(2))$ and $((b,c),(2,3))$ are 3 and 2, respectively, so $((b,c),(2))$ is a frequent sequence across two sequential dimensions. Thus, the
lattice structure in the second sequential dimension contains \(<(2)\>\) and inter-dimension links are built between lattice structures in the first sequential dimension and that in the second sequential dimension.

**Figure 2.** An example of generating atomic patterns in the second sequential dimension

**Figure 3.** An example of generating sequential patterns whose number of elements is 1 in the second sequential dimension

**Figure 4.** An example of generating sequential patterns with their number of elements larger than 1 in the second sequential dimension

Step 3: Obtain sequential patterns across two sequential dimensions through combination operation, where sequential patterns in each sequential dimension have more than one element:

Assume that pattern \(P \in SP_1\) and \(|P|>1\). Similar to Step 2, we can obtain the components of \(P\), \(X\) and \(Y\), by traversing intra-dimension links among lattice structure in the first sequential dimension, and the sequential patterns across two sequential dimensions, which are pointed to by their inter-dimension links, can be determined. Suppose pattern \(P=<(a)(b,c)>\) in Fig.4. The intra-dimension and inter-dimension links yield \(<(a)\>,<(1)\>\) and \(<(b, c)\>,<(2)\>\). In light of Definition 5, we have \(PS(<(a)\>,<(1)\>)\)+\(PS(<(b, c)\>,<(2)\>)\). Therefore, \(P'=<(a)(b,c)\>,<(1)(2)\>\) is generated.

The above steps allow sequential patterns across \((k+1)\)-sequential dimensions to be derived from sequential patterns across \(k\)-sequential dimensions. Algorithm PSeq iteratively repeats the above three steps until all sequential dimensions are propagated.

3.2. MIDim: Memory indexing for mining multi-dimensional pattern

After mining sequential patterns from multiple sequential dimensions by PSeq, we begin to construct projected multi-dimensional database for each sequential pattern and use MIDim algorithm to
mine MD-patterns in projected database. In the beginning, we will give the definitions about prefix MD-pattern and ρ-index, based upon which MIDim is built.

Definition 7. (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 8: If a MD-pattern is (business,*,*) and the frequent multi-dimensional value is Chicago, we obtain new MD-pattern (business,Chicago,*) by appending Chicago to the corresponding dimension in (business,*,*), (business,*,*) is a prefix MD-pattern.

Definition 8. (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 multi-dimensional value x.

Example 9: As shown in Fig 5-(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 multi-dimensional value business in multi-dimensional database(MDB). Since business belongs to the first dimension Cust-grp in MDB, we denote the dimension number of business is 1.

3.2.1. MIDim Algorithm

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-patterns mining. By the pattern-growth theory, this algorithm discovers all MD-patterns with larger size recursively. The MIDim algorithm is outlined as follows.

Algorithm: MIDim
Input: projected database, min-sup
Output: All MD-patterns
Method:
1. Scan projected database into memory and find all frequent multidimensional values L, let F-list “x₁—...—xₙ (n=|L|) be a list of frequent multidimensional values;
2. For i=1 to n do{
3. Output the matching prefix MD-pattern ρᵢ of xᵢ;
4. Construct ρᵢ-index for prefix MD-pattern ρᵢ;
5. For prefix MD-pattern ρᵢ do{
6. Call mineIndexset(ρᵢ, ρᵢ-index);
7. Delete index set of ρᵢ;}}

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 multi-dimensional value x from position(pos+1) to |ds| in tuple t/*|ds| is the length of multi-dimensional 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. Scan each tuple t in ρ-index and 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. Call mineIndexset(ρ’, ρ’-index); }
}

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 value.
Step(2). Output the matching prefix MD-pattern for each frequent multidimensional value, and then construct indexing for each prefix MD-pattern.
Step(3). 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(4). Carry out Step(2) and Step(3) recursively until discovering all MD-patterns.
Example 10: Given the database MMSDB in Table 1, where the 2,3,4 column of MMSDB forms MDB. Given min-sup=2. We can see all records support sequential patterns \((<a>,<1>)\), so all multi-dimensional tuples in MDB consists of \((<a>,<1>)\)-projected database, from which we mine MD-patterns. The mining process is shown in Fig. 5.

![Diagram](image1)

**Figure 5.** Find the MD-patterns with MIDim algorithm

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

In view of the divide-and-conquer strategy, the mined frequent MD-patterns can be divided into four subsets: (1) the subset containing \((business,*,*)\); (2) the subset containing \((*,Chicago,*)\) but not containing \((business,*,*)\); (3) the subset containing \((*,*,middle)\) but not containing \((business,*,*)\), \((*,Chicago,*)\); (3) the subset containing \((*,*,retired)\) but not containing \((business,*,*)\), \((*,Chicago,*)\), \((*,*,middle)\). The classified result is shown in Fig. 6.

4. Experiments and performance evaluation

To evaluate the performance of algorithm PSeq-MIDim, we use different synthetic datasets in experiments, 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.

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Number of sequential dimensions</td>
</tr>
<tr>
<td>D</td>
<td>Number of sequences</td>
</tr>
<tr>
<td>C</td>
<td>Average number of elements within a sequence</td>
</tr>
<tr>
<td>T</td>
<td>Average number of items within an element</td>
</tr>
<tr>
<td>I</td>
<td>Total number of different items</td>
</tr>
</tbody>
</table>

Several experiments have been conducted to evaluate the performance of PSeq-MIDim. In the first experiment, we test the running times for different supports from 0.5% to 5.5% over three different synthetic datasets. The dimensionality and cardinality of multi-dimensional databases are set to 5 and 10. The result is in Fig. 7. With the smaller minimum support, the number of multi-dimensional sequential patterns will be larger; thereby the execution time of PSeq-MIDim increases. With a larger minimum support, the number of multi-dimensional sequential patterns in databases will be smaller, thus the runtime of PSeq-MIDim reduces.
Next, we compare the performances for different number of sequential dimensions over three different synthetic datasets, where the variable parameter is the dimensionality and cardinality of each dimension are set to 5 and 10. We test the minimum support 0.3%, changing from 2 to 5. In this test, the execution time of PSeq-MIDim also increases, since propagating patterns to other sequential dimensions needs to consume time.

In this section, we evaluate the scalability of PSeq-MIDim over different number of sequence. We choose three different synthetic dataset M2C3T4I200, M2C4T5I300, and M2C5T6I400. We set the minimum support 1%, and we set the cardinality of each dimension 10. As can be seen in Fig.9, with the number of sequences increasing, the runtime over different synthetic datasets rises, since the mined patterns as well as the size of database increase. Fig.9 also shows that PSeq-MIDim is scalable with the number of sequences on various synthetic datasets, so PSeq-MIDim has good scalability.

The fourth experiment is to observe the scalability of PSeq-MIDim over different number of dimensions. We choose three different synthetic dataset M2D2kC3T4I200, M2D2kC4T5I300, and M2D2kC5T6I400. Min-sup= 1% is set, and the cardinality of each dimension 10 is also set. We can find from Fig.10 that as the number of dimensionality grows, the execution time of PSeq-MIDim goes up over different synthetic datasets; this is due to the mined MD-patterns increase.

5. Conclusions

In this paper, we propose a new algorithm PSeq-MIDim for mining frequent multi-dimensional sequential patterns from multi-dimensional multi-sequence databases. The algorithm contains two steps: the sequential pattern mining and the multidimensional pattern mining. In the sequential pattern mining process, PSeq-MIDim employs PSeq to mine sequential patterns from multiple sequential dimensions. PSeq first mines sequential patterns in a starting sequential dimension, and then uses a lattice structure to store these sequential patterns. In light of the lattice structure, algorithm PSeq is able to propagate position sets of only atomic patterns to next sequential dimensions for mining sequential patterns in a
level-by-level manner. During the multidimensional pattern mining process, we apply MIDim to mine MD-patterns in the corresponding 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 it builds index set of each prefix MD-pattern, finds local frequent multi-dimensional values from index set and grows discovered patterns; at last it recursively constructs index set of the detected pattern and discovers all MD-patterns. The performance study shows that PSeq-MIDim has good scalability and is efficient in finding multi-dimensional sequential patterns over multi-dimensional multi-sequence databases.

6. Acknowledgements

This work 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.

7. References