## 國立東華大學
### 補助國內大學校院博士班研究生出席國際會議報告

<table>
<thead>
<tr>
<th>報告人姓名</th>
<th>陳奕鈞</th>
</tr>
</thead>
<tbody>
<tr>
<td>就讀學校系所及年級</td>
<td>資訊工程學系 博士班 三年級</td>
</tr>
<tr>
<td>會議期間</td>
<td>2010 7/5~7/8</td>
</tr>
<tr>
<td>會議地點</td>
<td>Lakehead University, Thunder Bay, Canada</td>
</tr>
<tr>
<td>會議名稱</td>
<td>(中文) 數位資訊管理國際研討會 2010</td>
</tr>
<tr>
<td></td>
<td>(英文) International Conference on Digital Information Management (ICDIM2010)</td>
</tr>
<tr>
<td>發表論文題目</td>
<td>(中文) 一個有效率的投影資料庫方法探勘循序關聯法則</td>
</tr>
<tr>
<td></td>
<td>(英文) An Efficient Projected Database Method for Mining Sequential Association Rules</td>
</tr>
</tbody>
</table>

報告內容應包括下列各項:

### 一、參加會議經過

2010 International Conference on Digital Information Management 是由 IEEE Technology Management Council 所贊助的學術研討會。為期四天，是有關數位資訊管理領域的研討會。會議研討主題包括網絡內容、資訊檢索、資料探勘、E-Learning、人工智慧、行動資料管理、和資訊安全等數十個領域。此會議共匯集 80 個世界各地的專家學者共同討論與發表最新的研究成果。本次會議議程的安排和規劃主要分為三部分：7/5 當天安排了 Tutorial，主題是有關電子醫療記錄和健康護理服務，第二和第三部分為論文發表和 keynote 演講，這二部份是安排在 7/6 和 7/7 兩天舉行，論文發表的場次共為 3 個會議廳、16 個場次舉行。最後一天為 Concluding Session，由與會人士一起討論這次會議的心得，及建議改進之處，並且介紹下一年度 ICDIM 舉辦地點，La Trobe University, Melbourne, Australia。

### 二、與會心得

筆者口頭報告的發表場次在 7/7 上午第一個 session 的第二位報告，簡報過程中，獲得相當多的指教與意見，這些保貴的建議，都將做為筆者將本篇論文修改版本轉投稿至期刊的珍貴資源與修改方向。ICDIM 2010 此次約有二百篇論文投稿，經過嚴格篩選後，本次會議共發表 80 篇學術論文，我們的論文能夠錄取，算是實屬難得的經驗。

### 三、建議

感謝國科會與學校補助參與此次會議研討會往返 Canada，Thunder Bay 的機票費與會議之註冊費，此行成果非常豐碩，視野也增進不少。台灣應多鼓勵研究生參與國際學術活動，在研究成果上能有更具有國際觀的表現與價值。提供研究生邁向學術殿堂的訓練，而強化臨場發表之自信心。此外，學校亦可積極爭取國際會議的舉辦，培養學生多元文化的視野。透過多元的交流活動，增進東華大學的國際曝光率，提高台灣在國際的能見度。

### 四、攜回資料名稱及內容

1. 2010 International Conference on Digital Information Management, Program schedule
2. 2010 International Conference on Digital Information Management, Proceeding

### 五、其他

附錄-論文全文
An Efficient Projected Database Method for Mining Sequential Association Rules

Yi-Chun Chen
Department of Computer Science and Information Engineering
National Dong Hwa University
Hualien, 974 Taiwan, R.O.C.
divien@gmail.com

Guanling Lee
Department of Computer Science and Information Engineering
National Dong Hwa University
Hualien, 974 Taiwan, R.O.C.
guanling@mail.ndhu.edu.tw

Abstract—The mining of sequential patterns has been studied for several years, however, to our best knowledge, no study has considered the mining of sequential association rules despite such rules also providing valuable knowledge about many real applications. The sequential association rule represent that a set of items usually occur after a specific order sequence. In this paper, the concept of sequential association rule is proposed and an efficient algorithm, the SAR (Sequential Association Rules) algorithm, is proposed to discover these hidden knowledge. A set of experiments is also performed to show that the benefit of our approach.

Keywords-component: sequential association rule; projected database method; data mining

I. INTRODUCTION

Over the past decade, sequential pattern mining is an important data mining problem with many applications, such as the behavior of customer purchase in the market prediction, alignment of DNA sequences and disease diagnoses and treatments. The basic idea of sequential pattern mining was first introduced in [2]. Given a set of data sequences, an apriori-based algorithm was proposed to discover all the subsequences whose supports are larger than a user-defined threshold. Subsequently, many approaches have been proposed to improve the efficiency of mining process, such as FreeSpan [3], PrefixSpan [4], SPADE [6], and SPAM [1], etc. In [3], by constructing projected databases, FreeSpan algorithm was proposed to find sequential patterns. And in [4], PrefixSpan algorithm was proposed to improve the performance of FreeSpan algorithm. It also used the projection scheme except that the projection is based on frequent prefixes. It is because that any frequent subsequence can be mined out according to the frequent prefixes. [6] and [1] introduced the vertical format and lattice concept methods, SPADE and SPAM algorithm, respectively. The SPADE algorithm was introduced to divide the candidate sequences into distinct groups such that each group can be completely stored in the main memory. By using a lexicographic sequence tree and bitmap representation, SPAM algorithm can speed up mining process of sequential patterns. Then, constraint-based sequential pattern mining based on constructing projected databases have been studied in [5]. By characterizing constraints for sequential pattern mining, it can find highly interesting sequential patterns to user. Therefore, it introduced a PG algorithm which is based on a prefix-monotone property to generate these patterns.

Exploring frequent sequential pattern can indeed extract useful information. In traditional sequential patterns mining, it usually mined out the specific order sequences, because the support of these sequences is higher than that of generic sequences. However, in real application, it is worth noting the conclusions of items which occurs after mining the specific order sequences. For example, in disease analysis, obesity (BMI ≥ 25) usually results in hypertriglyceremia. Moreover, obesity and hypertriglyceremia usually cause a higher incidence of complications, such as hypertension, diabetes mellitus and hyperuricemia. There is no a specific occurring order of the three complications. Therefore, in this application, <obesity, hypertriglyceremia> forms a sequential pattern and {hypertension, diabetes mellitus, hyperuricemia} is the itemset which occurs frequently after the sequence <obesity, hypertriglyceremia>.

The contribution of our work is that it introduces a new concept of sequential association rules. A sequential pattern \(s\) with positive conclusion \(A\) is a sequential association rule, \(s \rightarrow A\), where \(s\) is a sequential pattern and \(A\) is a frequent itemset following \(s\). Moreover, an efficient algorithm, SAR algorithm, which is based on PrefixSpan algorithm is proposed to generate them. Experimental results show the effectiveness of our approach in mining sequential association rules.

The remainder of the paper is organized as follows. Section 2 defines the problem and Section 3 discusses the main concepts of our algorithm. Experimental results and analyses are presented in section 4 and conclusions and future work is discussed in section 5.

II. PROBLEM DEFINITION

In this section, we formally define the problem of mining sequential association rule and some notations. Let \(I = \{i_1, ..., i_n\}\) be a set of items and itemset \(A\) is a subset of \(I\). A
sequence $s$ is an ordered list of items which is denoted as $s = \langle i_1, \ldots, i_j, \ldots, i_r \rangle$, where $i_1, i_2$, and $i_r$ are items belonging to $I$, i.e., $i_1, i_2, \ldots, i_r \in I$ and $1 \leq \alpha, \beta, \gamma \leq n$. $|s|$ and $|A|$ denote the lengths of $s$ and $A$, respectively. A sequence database $S$ is a set of sequence transactions, $<\text{Sequence}_\text{id}, \text{Sequence}>$, and the list of sequence is ordered by increasing time. A sequence transaction $S_I$ is said to contain a sequence $s$ if and only if $s$ is a subsequence of $S_I$. $\tau(s)$ is the collection of $\text{Sequence}_\text{id}$ which contains $s$. The number of $\text{Sequence}_\text{id}$ contained in $s$ is called the support of $s$, denoted as $\text{sup}(s)$. Given a user-defined support threshold $\text{min_sup} > 0$, a sequence $s$ is a sequential pattern in $S$ if and only if $\text{sup}(s) \geq \text{min_sup} \times |S|$. 

### Table I. A Sequence Database

<table>
<thead>
<tr>
<th>Sequence_id</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a b c d e</td>
</tr>
<tr>
<td>2</td>
<td>a d e b c</td>
</tr>
<tr>
<td>3</td>
<td>e a b d c</td>
</tr>
<tr>
<td>4</td>
<td>a c b f e</td>
</tr>
</tbody>
</table>

Example 2.1 Refer to the sequence database listed in Table I. Given $\text{min_sup} = 0.5$, the set of items in the sequence database is $\{a, b, c, d, e, f\}$. Take sequence $<a,c>$ as an example, $\tau(<a,c>) = \{1, 2, 3, 4\}$ and $\text{sup}(<a,c>) = 4 \geq \text{min_sup} \times 4 = 2$. Sequence $<a,c>$ is a sequential pattern.

In the past researches on sequential patterns mining, the mining results are usually a set of specific order sequences. However, it is worth noting another specific type of sequential patterns. For example, refer to Table I, we know that $<a,b>$, $<a,b,c>$ and $<a,b,d>$ are sequential patterns, but $<a,b,c,d>$ or $<a,b,d,c>$ are not frequent in sequence database. However, the itemset $\{c,d\}$ usually occurs after the sequential pattern $<a,b>$. Therefore, in this paper, the concept of sequential association rule $(s \rightarrow A)$ is introduced. As mentioned in section 1, obesity usually results in hypertriglyceridemia. Moreover, obesity and hypertriglyceridemia usually cause a higher incidence of complications, such as hypertension, diabetes mellitus and hyperuricemia. Therefore, $\text{<obesity, hypertriglyceridemia> \rightarrow \{hypertension, diabetes mellitus, hyperuricemia\}}$ form a sequential association rule. Based on above discussion, the following definition is proposed:

### Definition 2.1 (Sequential Association Rule)

Given a sequential pattern $s$ in $S$, where $|s| \geq 2$, and an itemset $A \subseteq I$, they constitute a sequential association rule $s \rightarrow A$, if and only if the following conditions hold:

1) Let $s = \langle i_1, \ldots, i_j, \ldots, i_r \rangle$, it satisfies that $\{i_1, \ldots, i_j, \ldots, i_r\} \cap A = \emptyset$.

2) It satisfies that $\text{sup}(s \rightarrow A) \geq \text{min_sup}$ and $\text{conf}(s \rightarrow A) \geq \text{min_conf}$.

3) There does not exist $s' = \langle j_1, \ldots, j_n \rangle$ such that $ss' \geq \text{min_sup}$ and $A \subseteq \bigcup_{j \in s'} \{j_i, j_n \in I \text{ and } 1 \leq k, m \leq n \}$.

### III. SAR Algorithm

The basic concept of our approach, SAR algorithm, is based on PrefixSpan algorithm [4]. It can be decomposed into three stages: (1) it finds the frequent 1-itemsets, and for each frequent 1-itemset, a projected database is constructed. Then, 2-sequential patterns are mined from the projected database accordingly; (2) Begin from 2-sequential patterns, projected databases are constructed and used to mine out next length of sequential patterns and sequential association rules, recursively; (3) Finally, sequential association rules generated in stage 2 are compared with sequential patterns to eliminate the rules which violate condition 3 of definition 2.1.

A detailed description of the algorithm is given in Fig. 1 and 2. The algorithm takes three inputs: a sequence database, $S$, $\text{min_sup}$, and $\text{min_conf}$. Two arrays, $SP[]$ and $SAR[]$, are used to store the set of sequential patterns and sequential association rules, respectively. In line 1 of Fig. 1, the subroutine Modify_PrefixSpan is executed to find sequential patterns and sequential association rules. Continuously, by comparing $SP[ ]$ and $SAR[ ]$, the sequential association rules which violate condition 3 of definition 2.1 are removed from SAR[ ] to avoid false positive(line 2- line 9). The final output is the set of sequential association rules.

#### Algorithm

Input: A sequence DB $S$, the minimum support threshold $\text{min_sup}$, and the minimum confidence threshold $\text{min_conf}$

Output: The set of sequential association rules

Method:

1. Modify_PrefixSpan(\langle >, l, S)

2. For each sequential association rule $X$ in SAR[] do

3. { For each sequential pattern $Y$ in SP[] do

4. { If $(X=Y)$

5. Delete $X$

6. Else

7. Output $X$

8. }/\text{end for}

9. }/\text{end for}

Figure 1. The SAR algorithm.

The subroutine Modify_PrefixSpan is discussed in Fig. 2. According to Definition 2.1, by considering the length of sequential pattern, it can be decomposed into two parts (line 2 and line12). We generate all 2-sequential patterns in the first part (line2- line 11). Second, for the sequential patterns generated in the first part, by constructing projected databases of the sequential patterns, we can mine out sequential patterns.
and sequential association rules level by level, recursively (line 12–line 28). Moreover, the confidence of sequential association rule is calculated in this part and used to check whether the rule is strong (line 25). Example 3.1 is used to illustrate the main concept of our algorithm.

**Subroutine Modify_PrefixSpan**\( (a, l, S_a) \)

\( a \): a sequential pattern

\( l \): the length of \( a \)

\( S_a \): the \( a \)-projected DB

Method:

1. \( s = < > \)
2. If \( (l < 2) \)
3. } Scan \( S_a \)
4. } Find the set \( F_a \) of frequent items
5. For each frequent item \( b \in F_a \) do
6. \( s = s \cup < ab > \)
7. For each \( {a'} \in s \) do
8. } Construct \( {a'} \)-projected DB \( S_a \)
9. } Modify_PrefixSpan\( (a', l+1, S_a) \)
10. } // end for
11. } // end if
12. Else
13. } Scan \( S_a \)
14. } Find the set \( F_a \) of frequent items
15. } Find the set \( FI \) of frequent itemsets \( A \)
16. For each frequent item \( b \in F_a \) do
17. } \( s = s \cup < ab > \)
18. } \( SP[ ] \) \( \leftarrow s \)
19. } // end for
20. For each \( {a'} \in s \) do
21. } Construct \( {a'} \)-projected DB \( S_a \)
22. } Modify_PrefixSpan\( (a', l+1, S_a) \)
23. } // end for
24. For each frequent itemset \( A \in FI \) do
25. } if \( \text{conf}(a \rightarrow A) \geq \text{min}_\text{conf} \)
26. } \( \text{SAR} \ [ ] \) \( \leftarrow aA \)
27. } // end for
28. } // end else

![Figure 2. Modify_PrefixSpan subroutine.](image)

**Example 3.1** Refer to Table I. Let \( \text{min}_\text{sup} = 0.5 \) and \( \text{min}_\text{conf} = 0.5 \), sequential association rules can be found in the following steps.

1) Scan \( S \) once to find all frequent 1-sequential patterns, \(<a>: 4, <b>: 4, <c>: 4, <d>: 3, <e>: 4, <f>: 2\)

2) According to the six prefixes, the corresponding projected databases are constructed and frequent 2-sequential patterns are generated. For example, sequences in \( S \) containing \(<a>\) form the \( a \)-projected database. By scanning \( a \)-projected database, the 2-sequential patterns having prefix \(<a>\) can be mined, \(<a, b>: 4, <a, c>: 4, <a, d>: 3, <a, e>: 3, <a, f>: 2\).

3) Subsequently, for the sequential patterns generated in the step 2, by constructing projected databases of the sequential patterns, we can mine out sequential patterns and sequential association rules level by level, recursively. For example, by using 2-sequential patterns having prefix \(<a, b>: , <a, c>: , <a, d>: , <a, e>: , <a, f>: , and <a, f>:\), five projected database are constructed. Mining \(<a, b>:\)-projected database returns four 3-sequential patterns, \(<a, b, c>: 3, <a, b, d>: 2, <a, b, f>: 2, and <a, b, e>: 2,\) and two frequent 2-itemsets, \({c, d}: 2\) and \({e, f}: 2\). Because \( \text{conf}(<a, b \rightarrow {c, d}>)=2/4=0.5 \geq \text{min}_\text{conf} \) and \( \text{conf}(<a, b \rightarrow {e, f}>)=0.5 \geq \text{min}_\text{conf} \), these two sequential association rules are strong. Therefore, the rules \(<a, b \rightarrow {c, d}>\) and \(<a, b \rightarrow {e, f}>\) are stored into the array SAR[ ]. Similarly, we can find sequential patterns and sequential association rules on prefix \(<b>: , <c>: , <d>: , <e>: , and <f>:\). The constructing projected databases and mining process are illustrated in Fig. 3.

4) Eventually, we check that \(<a, b, c, e>: and <a, c, f, e>: \) are sequential patterns. Therefore, \(<a, b \rightarrow {c, d}>\) and \(<a, c \rightarrow {e, f}>\) are not the valid sequential association rules. The final result is \(<a \ b \rightarrow {e, f}>\) and this phase is shown in Fig. 4.

IV. EXPERIMENTAL RESULTS

<table>
<thead>
<tr>
<th>Table II. Parameter Setting of the Synthetic Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Notation</strong></td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>min_sup</td>
</tr>
<tr>
<td>min_conf</td>
</tr>
</tbody>
</table>

**A. Simulation Platform**

This section describes a set of simulations performed to assess the benefit of our approach. The test data were generated by an IBM synthetic data generator. The parameter settings for generating the test dataset and evaluating the approach are listed in Table II. In this paper, due to the limit of space, only the results on default dataset C10T1 are reported. In this dataset, the number of distinct items is 1000, and the number of sequences is 10000. C10 represents the average number of transactions of each customer is 10 and T1 means that each transaction contains only one item. The simulations are implemented in C++ and all the experiments are performed in an Intel Pentium 3.20GHz with 3GB of memory, running Windows 7.

**B. Efficiency Analysis**

In the first experiment, we test the execution time of scalability with support threshold are shown in Fig. 5. The result show that when the support threshold is high, few sequential patterns are mined and the length of patterns is short. Comparatively, as the support threshold decreases, there are
### Figure 3. An example of SAR algorithm

<table>
<thead>
<tr>
<th>Sequence_id</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a b c d f e</td>
</tr>
<tr>
<td>2</td>
<td>a d e b c</td>
</tr>
<tr>
<td>3</td>
<td>e a b d c</td>
</tr>
<tr>
<td>4</td>
<td>a c b f e</td>
</tr>
</tbody>
</table>

#### A sequence database

**Frequent 1-sequence**
- `a`: 4
- `b`: 4
- `c`: 4
- `d`: 3
- `e`: 4
- `f`: 2

**<a b>-projected DB**
- 1: c d f e
- 2: c
- 3: d c
- 4: f e

**<a c>-projected DB**
- 1: d f e
- 4: b f e

**<a f>-projected DB**
- 1: e
- 4: e

#### b-projected DB

<table>
<thead>
<tr>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

**Frequent 2-sequence**
- `<a b>`: 4
- `<a c>`: 4
- `<a d>`: 3
- `<a e>`: 3
- `<a f>`: 2

#### <a b>-projected DB
- 1: d f e
- 2: <a b d>: 2
- 3: <a b f>: 2

**<a b c>-projected DB**
- 1: d f e
- 3: c d
- 4: f e

**<a b d>-projected DB**
- 1: <a b d>: 2
- 3: c

**<a b f>-projected DB**
- 1: <a b f>: 2
- 4: e

**<a b f e>-projected DB**
- 1: <a b f e>: 2

#### 3-sequence

**<a b c>-projected DB**
- 1: d f e
- 3: c d
- 4: f e

**Frequent 3-sequence**
- `<a b c>`: 3
- `<a b d>`: 2
- `<a b f>`: 2
- `<a b e>`: 2

**Frequent 2-itemset of prefix `<a b>`**
- `{c d}`: 2
- `{e f}`: 2

Figure 4. The final results of SAR algorithm

### Figure 4. The final results of SAR algorithm

**SP[ ]**
- `<a b>`: 4
- `<a c>`: 4
- `<a d>`: 3
- `<a e>`: 3
- `<a f>`: 2
- `<a b c>`: 3
- `<a b d>`: 2
- `<a b f>`: 2
- `<a b e>`: 2
- `<a b f e>`: 2
- `<a c f>`: 2
- `<a c e>`: 2
- `<a c f e>`: 2

**SAR[ ]**
- `<a b>` → `{c d}`: 2
- `<a b>` → `{e f}`: 2
- `<a c>` → `{e f}`: 2

**output**
- `<a b>` → `{c d}`: 2
many sequential patterns and the length of patterns is long. Therefore, the execution time increases as the support threshold is decreased because of a large increase in the number of sequential patterns in the result. The number of sequential association rules generated in different support thresholds is shown in Fig. 6. When min_sup is 0.5%, few sequential association rules are mined. This is because that there are many sequential patterns as support threshold is low. Therefore, only longer sequences can satisfy the properties of sequential association rules in a low min_sup. As support threshold increases from 0.5% to 1.5%, the number of sequential association rules increase. This is because that few sequential patterns are mined and the length of patterns is short. However, according to the property of apriori, the support of short length of sequences is larger than that of long length of sequences. As a result, the probability of a short sequential pattern to form a sequential association rule is higher than that of a long sequential pattern. As the support threshold is between 1.5% and 3%, the number of sequential association rules decreased. The main reason for the decrease is that the minimum support is high enough, the restriction of sequential association rules become more tightly.

In summary, the experimental results described above demonstrate that our SAR algorithm offers the scalability in efficiency and reports sequential association rules with different support thresholds.

![Figure 5. Execution time of our algorithm with different min_sup.](image)

![Figure 6. Number of sequential association rules.](image)

V. CONCLUSION AND FUTURE WORK

In this paper, a new concept of sequential association rules is introduced. Mining sequential association rules provide valuable knowledge which cannot get from sequential patterns mining and it can be applied in many applications, such as disease analysis and market decision. Moreover, SAR algorithm is proposed for mining sequential association rules. The experiments show that the proposed algorithm offers execution time of scalability with support threshold and reports sequential association rules.

There remain some problems that are worth studying in the future. First, investigating more efficient algorithms to improve the scalability of mining sequential association rules is important. Furthermore, extending to find negative sequential association rules will also be worthwhile.

REFERENCES


