Sequence Mining for Similar Mental Concepts

M. Gholizadeh, M. M. Pedram, J. Shanbehzadeh

Abstract— **Sequence mining is one of very important fields in data mining studies in recent decade. In fact, sequence mining recognizes subsequences repeated in a temporal database. All proposed sequence mining algorithms focus only on the items with support higher than specified threshold. Considering items with similar mental concepts can lead to some general and more compact sequences in database which might not be distinguished before when the support of individual items were less than threshold. In this paper an algorithm is proposed to find sequences containing more general concepts by considering mental similarity between the items. In this work, fuzzy ontology is used to describe the similar mental concepts.**

Index Terms— **Sequence mining, Subsequence, Similar mental concept, Fuzzy ontology.**

I. Introduction

Sequences are an important type of data which occur frequently in many scientific, medical, security, business and other applications. For example, DNA sequences encode the genetic makeup of humans and all other species; and protein sequences describe the amino acid composition of proteins and encode the structure and function of proteins. Moreover, sequences can be used to capture how individual humans behave through various temporal activity histories such as weblogs histories and customer purchase ones. In general there are various methods to extract information and patterns from data bases, such as Time series, association rule mining and data mining.

Time series is the collection of observations sorted by time. In this case, the patterns and information are derived by studying the time items occurred [1, 2]. Association rule mining algorithms usually have two stages: (1) finding a collection of repeated items, (2) selecting suitable rules from highly repeated collections. In the first stage, the highly repeated collections are produced by use of methods like Apriori algorithm and via counting the number of items repeats. Then, the patterns and information are generated from the collections produced in the previous stage [3-5]. Sequence mining is the recognition of repeated subsequences in a set of sequential data. In such a method, the input data includes a list of transactions associated with their occurrence time. Moreover, each transaction contains a set of items. Sequential pattern is a set of sequentially happening items. The major purpose of sequence mining is to search and find all the sequential patterns having support values greater than or equal to minimum support value defined by the user [6-8]. Figure 1 demonstrates a schematic view of such a category.

In this paper a new method for sequence mining is proposed in which similar mental concepts are described by a common item set, therefore, more general sequences can be found with higher support values.

The remaining text of the paper is organized as follows. In Section 2, we define the sequence mining and introduce two standard sequence mining methods. Then, PrefixSpan algorithm is described in Section 3. Section 4 studies the algorithm of sequence mining considering similar mental concepts. An illustrate example is presented and investigated in Section 5. Finally, a brief conclusion is introduced in section 6.

II. Sequence mining

The main aim of sequence mining is to search and find all the sequential patterns with support values greater than or equal to a minimum support criteria (declared by the user). The following sentence is an example of sequential patterns: “Customers who have purchased a printer, are reasonably probable to purchase printer ink, too”. In this example, the purchase of printer and printer ink can represent a sequence. As can be seen in Figure 1, there are Apriori and PrefixSpan algorithms for finding repeated or frequent sequences.

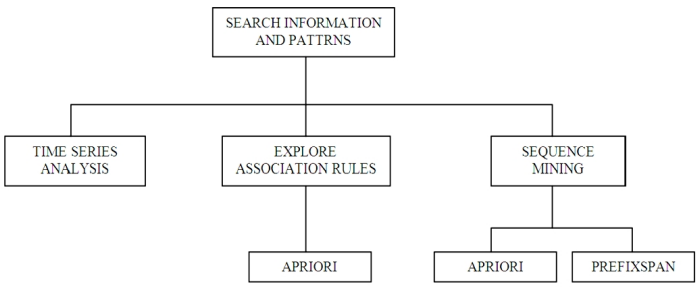

Fig 1: Frequent pattern mining studies

Manuscript received January 12, 2010.

M. Gholizadeh is MS student in the Computer Engineering Department at Islamic Azad University Science and Research Branch, Tehran, Iran (e-mail: mhdgholizadeh@ gmail.com).

M. M. Pedram, is with the Computer Engineering Department, Faculty of Engineering, Tarbiat Moallem University, Karaj/Tehran, Iran (e-mail: pedram@tmu.ac.ir).

J. Shanbehzadeh is with the Computer Engineering Department, Faculty of Engineering, Tarbiat Moallem University, Karaj/Tehran, Iran (e-mail: shanbehzadeh@gmail.com).

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

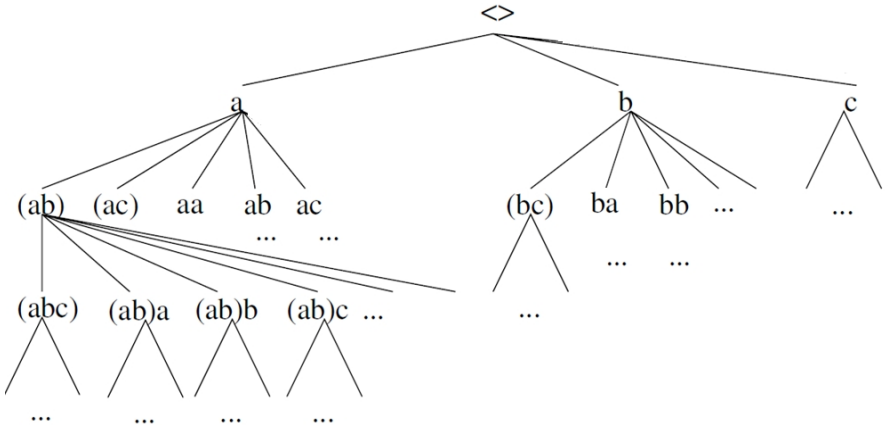

Fig 2: The sequence enumeration tree on the set of items {a, b, c}

Comparing the Apriori algorithm with the PrefixSpan algorithm, the latter operates faster, so it has been selected as the base algorithm in this article.

III. PrefixSpan Algorithm

At the first we introduce the concepts of prefix and suffix which are essential in PrefixSpan.

➢ Prefix:

Suppose all the items within an element are listed alphabetically. For a given sequence $\alpha=e_1e_2\ldots e_n$, where each e_i($1 \le i \le n$) is an element, a sequence $\beta=e'_1e'_2\ldots e'_m$ ($m \le n$) is called a prefix of α if (1) $e'_i=e_i$ for $i \le m-1$; (2) $e'_m \subseteq e_m$ and (3) all items in $(e_m-e'_m)$ are alphabetically after those in e'_m.[7]

For example, consider sequence s=a(abc)(ac)d(cf). Sequences a, aa, a(ab) and a(abc) are prefixes of s, but neither ab nor a(bc) is a prefix.

➢ Suffix

Consider a sequence $\alpha=e_1e_2\ldots e_n$ where each $e_i(1 \le i \le n)$ is an element. Let $\beta=e'_1e'_2\ldots e'_m$ ($m \le n$) be a subsequence of α. Sequence $\gamma=e''_pe_{p+1}\ldots e_n$ is the suffix of α with respect to prefix β, denoted as $\gamma=\alpha/\beta$, if

1. $p=i_m$ is the suffix of α with respect to prefix β, denoted as $\gamma=\alpha/\beta$, if $p=i_m$ such that there exist $1 \le i_1 \le \ldots \le i_m \le n$ such that there exist $e'_j \subseteq e_{ij}$ $(1 \le j \le m)$ and i_m is minimized. In other words, $e_1\ldots e_l$ is the shortest prefix of α which contains $e'_1e'_2\ldots e'_{m-1}e'_m$ as a subsequence; and
2. e''_p is the set of items in e_p-e'_m that are alphabetically after all items in e'_m.

If e''_p is not empty, the suffix is also denoted as (- items in e''_p) $e_{p+1}\ldots e_n$. Note that if β is not a subsequence of α, the suffix of α with respect to β is empty [7].

For example, consider sequence s=a(abc)(ac)d(cf), (abc)(ac)d(cf) is the suffix with respect to a, (c)(ac)d(cf) is the suffix with respect to ab, and (ac)d(cf) is the suffix with respect to a(abc).

The depth first search method is applied on the tree of the Fig. 2, as it is apparent from the method name. In this tree, sub-trees related to each node indicate all sequence patterns which are prefixes of the node. This tree is called as sequence enumeration tree.

Input: A sequence database S, and the threshold
Output: The complete set of sequential patterns.
Method: Call *PrefixSpan*$(\emptyset, 0, S)$.

Subroutine *PrefixSpan*$(\alpha, \mathbf{p}, S|_\alpha)$
The parameters are (1) α is a sequential pattern; (2) $\mathbf{p}$ is the i-length of α; and (3) $S|_\alpha$ is the α-projected database if $\alpha \neq \emptyset$, otherwise, it is the sequence database S.

Method:

1. Scan $S|_\alpha$ once, find each frequent item b such that
   a) b can be assembled to the last element of α to form a sequential pattern; or
   b) b can be appended to α to form a sequential pattern.
2. For each frequent item b, append it to α to form a sequential pattern α', and output α';
3. For each α', construct α'-projected database $S|_{\alpha'}$, and call *PrefixSpan*$(\alpha', \mathbf{p}+1, S|_{\alpha'})$.

Fig 3: The PrefixSpan Algorithm [7]

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

➢ Projected database:

Let α be a sequential pattern in a sequence database S.

The α -projected database, denoted as $S|_\alpha$, is the collection of suffixes of sequences in S with respect to prefix α. Based on the above discussion, the algorithm of PrefixSpan is presented in Figure 3.

IV. Sequence Mining for Similar Mental Concept

The mental similarity between items has not been considered in the sequence mining algorithms. While using these similarities, we can achieve more general sequences. In other words, not only we review the number of a specific item repeats, but also the items with mental similarity are gathered in a collection and put under a general concept. In this case, rather than studying the items one by one, the number of repeats of the general concepts is calculated, which results in sequences with upper support and besides, more general sequences. For this purpose, in addition to the data collection that shows the transactions, there should be another collection which represents the items similarities. Ontology can be used to show the similar mental concepts. Ontology is a method for representing knowledge in an understandable form for both human and machine and provides the ability to share the information between different programs. All the concepts in the desired range, associated with their hierarchical structure and the existing relations between concepts are defined in ontology. In fuzzy ontology we also can model and represent the uncertainty of real world [10, 11].

The proposed algorithm receives two sets as inputs. The first data set is a collection including identification number, time, number of items and the number of items happening. The second data set describes the similarity between each item and each general concept by a membership degree, i.e., the fuzzy ontology database. Then, the PrefixSpan algorithm is implemented on the new data set and the final results are sequences with more general concepts.

Nomenclator

A_i: i-th general concept,

a_j: j-th item which has mental similarity with the i-th general concept,

C_k: Identification number,

t_m: Transaction date,

$n_{aj}(t_m)$: Number of item a_j in the transaction with date t_m,

Similarity (A_i, a_j): The measure describing the similarity of item a_j and the general concept A_i ,

Count(A_i, t_m, C_k): Number of times that concept A_i occurred by the identification number C_k at the time t_m.

Algorithm:

- Inputs:
  1. The data set including identification number, time, number of items and the number of items happening
  2. The data set containing a list of similar mental concepts by which their similarity is determined via fuzzy ontology.
- Outputs:

  General sequences that indicate the items regularity and priority.
- Steps:
  1. Receive the first and second data sets and build the new one as follows:
     a. The identification number (C_k) and the transaction date (t_m) get no change,
     b. The items a_j are replaced with the concepts A_i ,
     c. The number of occurrence of the concept A_i is calculated as:

$$Count(A_i, t_m, C_k) = Count(A_i, t_m, C_k) + Similarity(A_i, a_j) \times n_{a_j}(t_m) \quad (1)$$

  2. Implement the PrefixSpan algorithm on the new data set.
  3. Return the derived general sequences in step 2.
  4. End.

V. Illustrated Example

As an example, consider the transactional data set shown by table 1. Table 2 shows the fuzzy ontology, in which general concepts as well as items are shown. In fact, the similarity degree for item a_j and general concept A_i is shown by the table.

The original data set (table 1) is transformed into table 3 by (1), in which general concepts are used.

Table 1. Transactions of some customers

Customer identification number	Purchase time	item	Number
100100002	95/07/22	Tea	1
100100002	95/07/23	Cream	3
100100003	95/07/22	Butter	3
100100003	95/07/23	Coffee	1
100100002	95/07/27	Fruit juice	2
100100003	95/07/29	Fruit juice	1

Table2. Fuzzy ontology

item (a_j)	General Concept (A_i)	
	Hot drink	**Fat dairy**
Tea	1	0
Coffee	1	0
Cream	0	0.9
Butter	0	0.9
Fruit juice	0.1	0

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table3. Transactions of the customers with general concepts

Customer identification number	Purchase time	General Concepts	Number
100100002	95/07/22	Hot drink	1
100100002	95/07/23	Fat dairy	2.7
100100003	95/07/22	Fat dairy	2.7
100100003	95/07/23	Hot drink	1
100100002	95/07/27	Hot drink	0.2
100100003	95/07/29	Hot drink	0.1

The PrefixSpan algorithm was applied on the data set shown by table 3, with the minimum support equal to 1. The results are listed in table 4, in which each item with its count is shown.

For comparison, the result of applying the PrefixSpan algorithm on table 2 is shown by table 5. It is clear that the sequences shown in table 4 are more general with higher support values.

It is worth mentioning, if the minimum support was set to 1, there would be no output sequence in table 5, while table 4 would be gained with no change.

Table 4. Output sequences found by proposed method

Sequences	Support values
(Hot drink : 1)	2
(Fat dairy : 2.7)	2
(Hot drink : 1 , Fat dairy : 2.7)	2
(Hot drink : 0.2)	1
(Hot drink : 0.1)	1

Table 5. Output sequences found by applying the PrefixSpan algorithm on table 1

Sequences	Support values
(Cream : 3)	1
(Cream : 3 , Tea : 1)	1
(Tea : 1)	1
(Coffee : 3)	1
(Coffee : 3 , Butter : 2)	1
(Coffee : 3 , Butter : 2)(Fruit juice : 1)	1
(Butter : 2)	1
(Fruit juice : 1)	1
(Fruit juice : 2)	1
(Cream : 3) (Fruit juice : 2)	1
(Cream : 3, Tea : 1) (Fruit juice : 2)	1
(Tea : 1) (Fruit juice : 2)	1
(Coffee : 3) (Fruit juice : 1)	1
(Fruit juice : 1)	1

VI. Conclusion

In this paper, we introduce a new algorithm for sequence mining. This algorithm works based on the similar mental concepts and uses the PrefixSpan algorithm and gives more general results as output sequences. Moreover, the proposed method is able to find the sequences which may be hidden when no mental similarity is considered.

VII. Reference

[1] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, “Fast subsequence matching in time-series databases”, Proceedings of the ACM SIGMOD International Conference on Management of Data, Minneapolis, Minnesota, 1994.

[2] B. LeBaron, A.S. Weigend, “A bootstrap evaluation of the effect of data splitting on financial time series”, IEEE Transactions on Neural Networks 9 (1) (1998) 213–220.

[3] C.Y. Chang, M.S. Chen, C.H. Lee, “Mining general temporal association rules for items with different exhibition periods”, IEEE International Conference on Data Mining, Maebashi City, Japan, 2002.

[4] C.H. Lee, M.S. Chen, C.R. Lin, Progressive partition miner: “an efficient algorithm for mining general temporal association rules”, IEEE Transactions on Knowledge and Data Engineering 15 (4) (2003) 1004–1017.

[5] Kuok C.-M. ,Fu A., Wong M. H., “Mining Fuzzy Association Rules in Databases, ” SIGMOD Record, vol. 27,no. 1, pp. 41-46, 1998.

[6] Agrawal, R. and Srikant, R. (1995), “Mining sequential patterns, in P. S”, Yu and A. S. P. Chen, eds, ‘11th International Conference on Data Engineering (ICDE’95)’, IEEE Computer Society Press, Taipei, Taiwan, pp. 3–14.

[7] Guozhu Dong, Jian Pei , *Sequence Data Mining*, Springer Science+Business Media, LLC, 2007.

[8] X. Yan, J. Han, CloSpan: “mining closed sequential patterns in large datasets”, Proceedings of the 2003 SIAM International Conference on Data Mining (SDM’03), San Francisco, California, May, 2003.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)