Data & Knowledge Engineering xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Mining time-interval univariate uncertain sequential patterns

Ying-Ho Liu *

Department of Information Management, National Dong Hwa University, No. 1, Sec. 2, Da Hsueh Road, Hualien 97401, Taiwan, ROC

ARTICLE INFO

Article history:
Received 25 April 2014
Received in revised form 5 July 2015
Accepted 29 July 2015
Available online xxxx

Keywords:
Data mining
Mining methods and algorithms
Sequential pattern mining
Uncertain data
Univariate uncertain data
Time-interval U2-sequential pattern

ABSTRACT

In this study, we propose two algorithms to discover *time-interval univariate uncertain* (*U2*) *-sequential patterns* from a set of *univariate uncertain* (*U2*)*-sequences*. A U2-sequence is a sequence that contains transactions of *univariate uncertain data*, where each attribute in a transaction is associated with a quantitative interval and a probability density function indicating the possibility that each value exists in the interval. Many sources record U2-sequences, such as atmospheric pollution sensors and network monitoring systems. Mining sequential patterns from these U2-sequences is important for understanding the intrinsic characteristics of the U2-sequences. The proposed two algorithms are based on the candidate generate-and-test methodology and pattern growth methodology, respectively. We performed a series of experiments to evaluate them in terms of runtime and memory consumption. The experimental results show that different algorithms excel when applied to different conditions. In general, the algorithm based on the pattern growth methodology is the better choice.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Sequential pattern mining plays an important role in the frequent pattern mining area, which discovers frequent subsequences from a sequence database. In a sequence database, a sequence records a series of actions or values generated by an entity. For example, a sequence may record the respective items bought by a customer in the course of several transactions completed at different points in time.

Fig. 1 shows such a database, where each row represents a sequence. Each sequence, identified by a sequence identifier (SID), consists of several transactions made by a customer. For instance, the customer of sequence S_1 made three transactions, purchasing milk and bread, a DVD player, and a blanket and a towel. Each transaction in a sequence is also called an *element*. If we set the *minimum support* as 2, which means a subsequence is *frequent* if this subsequence appears in at least two sequences, the subsequence ⟨(milk) (towel)⟩ is frequent because it appears in two sequences (S_1 and S_2). Therefore, ⟨(milk) (towel)⟩ is a *sequential pattern*, which indicates customers often buy a towel after buying milk.

Many methods have been proposed to handle sequential pattern mining [2,4,6–8,13,15–20,26,31–34,37,38,40]. However, all these methods deal with *precise sequence databases*, where a transaction records several specific items that *actually* appear, such as milk and bread appearing in the first transaction of S_1 in Fig. 1. In contrast, many data sources record *uncertain data* in the real world. Uncertain data refers to the content of a transaction that is not accurately recorded; for example, where an item like milk exists in a transaction with an *existential probability* indicating the possibility that the item exists in the transaction. This type of uncertain data is referred to as *itemset uncertain data* [9,10]. Similarly, yet another type of uncertain data is *univariate uncertain data* [23], where each attribute in a

* Tel./fax: +886 3 8633116.
E-mail address: daxliu@mail.ndhu.edu.tw.

http://dx.doi.org/10.1016/j.datak.2015.07.012
0169-023X/© 2015 Elsevier B.V. All rights reserved.

Please cite this article as: Y.-H. Liu, Mining time-interval univariate uncertain sequential patterns, Data Knowl. Eng. (2015), http://dx.doi.org/10.1016/j.datak.2015.07.012

SID	Sequence
S_1	⟨ (milk, bread) (DVD player) (blanket, towel) ⟩
S_2	⟨ (digital camera) (milk, rice) (soap, towel, toothpaste) ⟩
S_3	⟨ (bread, egg, coffee) (trash can, bath mat, towel) ⟩

Fig. 1. A precise sequence database.

transaction is associated with a quantitative interval and a probability density function indicating the possibility that each value exists in the interval. For example, a low sensitivity sensor used to record atmospheric pollution may record a quantitative interval, instead of a precise value, to indicate the amounts of suspended particulates at 06:00 every day. A probability density function is then explicitly or implicitly assigned to the interval. Another example is a network monitoring system that records a quantitative interval for network traffic flow every hour and assigns a probability density function to the traffic volume. Such data may also be perceived as a kind of sequence database if we treat a set of daily transactions as a sequence, or treat the transactions recorded by each sensor as a sequence. Fig. 2 presents a *univariate uncertain sequence database* (*U2-sequence database*). Suppose a sensor records the values corresponding to measurements of the atmospheric pollution three times in a day and that leads to three transactions (elements) in a sequence. Each transaction contains two attributes, A_1 and A_2. Therefore, each element has two quantitative intervals for the two attributes, respectively. In addition, each element also records the time at which it occurs. For instance, the first element of S_1, [A_1:[12, 15], A_2:[30, 75]]:1, represents the respective intervals of the two attributes and the occurring time (at 1 o'clock). We call such a sequence a *univariate uncertain sequence* (*U2-sequence*). Without loss of generality, we set the probability density function over each interval as a uniform distribution.

U2-sequence databases can also be constructed intentionally. For instance, it is believed that the prices of different stocks may be correlated. To observe this kind of correlation, the maximum and minimum prices of each selected stock in a day can be used to form a quantitative interval. A transaction is formed by recording the quantitative intervals of the selected stocks, each of which is treated as an attribute. The probability density function associated with each interval is assigned according to the detailed daily stock prices. Such a transaction represents the price variations of the stocks in a day. The transactions recorded in a fixed time span, e.g., a week or a month, form a U2-sequence and each transaction accompanied by its occurring time. For example, a U2-sequence ⟨[A_1: [599, 601], A_2: [28,34]]:1; [A_1: [592, 598], A_2: [26,29]]:2; [A_1: [596, 605], A_2: [21,25]]:3; [A_1: [593, 595], A_2: [23,26]]:4; [A_1: [596, 598], A_2: [28, 31]]:5⟩ shows the price variations of two selected stocks, i.e., A_1 and A_2, in a week (five working days). Many existing studies only consider closing prices, which ignore variations of stock prices. Instead, univariate uncertain setting preserves complete stock price information. Therefore, mining results of higher quality are expected.

Sequential patterns in a U2-sequence database reveal the intrinsic regularity. One method to retrieve sequential pattern from a U2-sequence database is to treat average of an attribute's quantitative interval as the value of this attribute. Each attribute's average acts as an item and traditional sequential pattern mining approaches could retrieve sequential patterns from this transformed database. However, using average discards information provided by intervals; for example, both intervals [12, 16] and [6, 22] have 14 as average, however, the possible ranges of two intervals are quite different. Using average may distort the real intrinsic regularity of a U2-sequence database. Quantitative intervals should be considered to form sequential patterns. In addition, occurring time of each element provides additional useful information. Only retrieving sequential patterns without temporal relation limits practicality of patterns.

Therefore, we propose using quantitative intervals to form sequential patterns. In Fig. 2, a sequential pattern [[A_1:[12, 15], A_2:[30,75]]; [A_1:[12, 16], A_2:[30, 42]]] can be discovered if minimum support is set as 1, which means that the sensor typically records [A_1:[12, 15], A_2:[30,75]] and then [A_1:[12, 16], A_2:[30, 42]] in a day. We also consider occurring time of each element to derive sequential patterns with temporal relation. For instance, [[A_1:[12, 15], A_2:[30,75]]; I_t; [A_1:[12, 16], A_2:[30, 42]]], I_t: $20 < t \leq 23$, means that the sensor typically records [A_1:[12, 15], A_2:[30,75]], then [A_1:[12, 16], A_2:[30, 42]] in 20–23 h. Exploiting the temporal relation also helps the user to utilize the intrinsic information of the database for decision making purposes, such as predictions of future behavior.

In this study, we propose two mining algorithms for mining U2-sequence databases. The key contributions of this study are as follows: first, to the best of our knowledge, this study proposes the first approaches designed to mine U2-sequence databases. Second, this study explores the time interval relationship in a sequence, which is seldom addressed in the literature. Third, we propose two algorithms based on the candidate generate-and-test methodology and the pattern growth methodology, respectively. Fourth, comprehensive experiments are conducted to compare the performance of the proposed algorithms and three compared algorithms.

SID	U2-sequence
S_1	⟨ [A_1:[12, 15], A_2:[30, 75]]:1; [A_1:[14, 16], A_2:[36, 75]]:12; [A_1:[12, 16], A_2:[30, 42]]:23 ⟩
S_2	⟨ [A_1:[12, 14], A_2:[30, 42]]:1; [A_1:[15, 16], A_2:[36, 75]]:12; [A_1:[14, 16], A_2:[30, 36]]:23 ⟩
S_3	⟨ [A_1:[12, 15], A_2:[30, 75]]:1; [A_1:[15, 16], A_2:[30, 75]]:12; [A_1:[12, 15], A_2:[30, 42]]:23 ⟩

Fig. 2. A univariate uncertain sequence database.

Please cite this article as: Y.-H. Liu, Mining time-interval univariate uncertain sequential patterns, Data Knowl. Eng. (2015), http://dx.doi.org/10.1016/j.datak.2015.07.012

Download English Version:

https://daneshyari.com/en/article/10321188

Download Persian Version:

https://daneshyari.com/article/10321188

Daneshyari.com