View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by IUPUIScholarWorks

Towards Efficient Sequential Pattern Mining in Temporal Uncertain Databases

Jiaqi Ge[1], Yuni Xia[1], and Jian Wang[2]

[1] Department of Computer & Information Science, Indiana University Purdue University Indianapolis, Indiana, USA, 46202
[2] School of Electronic Science and Engineering, Nanjing University, Jiangsu, China, 210023
[1]{jiaqige,yxia}@cs.iupui.edu, [2]wangjnju@nju.edu.cn

Abstract. Uncertain sequence databases are widely used to model data with inaccurate or imprecise timestamps in many real world applications. In this paper, we use uniform distributions to model uncertain timestamps and adopt possible world semantics to interpret temporal uncertain database. We design an incremental approach to manage temporal uncertainty efficiently, which is integrated into the classic pattern-growth SPM algorithm to mine uncertain sequential patterns. Extensive experiments prove that our algorithm performs well in both efficiency and scalability.

Keywords: Temporal Uncertainty, Sequential Pattern Mining

1 Introduction

Sequential pattern mining (SPM) provides inter-transactional analysis for timestamped data and mines frequent patterns in sequence databases. However, it is very common that timestamps of events might be inaccurate or imprecise in real applications. And temporal uncertainty is usually caused by the following reasons:

- The exact time of an event is often unavailable. For example, in temperature monitoring sensor networks, temperatures are measured periodically. The exact time of a sudden temperature change is unknown, and it can only be inferred from raw data probabilistically.
- Temporal uncertainty arises when data are collected in different temporal scales. For example, a handhold GPS device may update the position every 10 minutes; while a GPS on a fast-moving vehicle may report every 5 seconds. And the temporal relationship is uncertain between two events within different granularities.
- Temporal uncertainty can also be caused by aggregation operations on temporal scales. For example, an economic indicator may be aggregated from weekly or monthly data to represent high level abstracted information in this time period.

This is the author's manuscript of the article published in final edited form as:
Ge, J., Xia, Y., & Wang, J. (2015). Towards Efficient Sequential Pattern Mining in Temporal Uncertain Databases. In Advances in Knowledge Discovery and Data Mining (pp. 268-279). Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-18032-8_21

– Temporal uncertainty is also used to protect privacy and confidentiality. Precise time information in monitoring data usually is not released if there is a potential to identify individuals. Therefore, uncertainty is introduced to original time points, which is unquantifiable and unknown by the data user.

A time series $T = \{t, (t+1), \ldots, (t+n)\}$ that bounds a set of consecutive timestamps is used to model an uncertain event time in probabilistic temporal databases, where it assumes that all events are defined within the same discrete time domain. However, this model becomes inaccurate and inconvenient when data are actually collected in different time scales. Instead, we use uniform distributions to represent uncertain timestamps in our model, which do no rely on any discrete time domain.

It is very important to carefully manage temporal uncertainty in SPM problems; otherwise, the mined patterns might be inaccurate. Possible world semantics is widely used to interpret probabilistic databases; however, it also brings efficiency and scalability challenges to uncertain SPM problems. Therefore, in this paper, we propose an efficient SPM algorithm in temporal uncertain sequence databases. Our main contributions are listed as follows:
(1) We model uncertain timestamps by uniform distributions. And we use possible world semantics to interpret this type of temporal uncertainty.
(2) We develop a novel approach to manage temporal uncertainty in the process of mining uncertain sequential patterns by a pattern-growth algorithm.
(3) We conduct extensive experiments to demonstrate the efficiency and scalability of our algorithm.

2 Related Works

Data mining in uncertain databases has been an active area of research recently. A lot of traditional database and data mining techniques have been extended to be applied to uncertain data [1]. Particularly, Muzammal and Raman proposal the SPM algorithm in probabilistic database using the expected support as the measurement of pattern frequentness [10]; however, expected support has inherent weakness in mining high-quality sequential patterns[12]. Zhao et al. measure pattern frequentness using possible world semantics and propose a pattern-growth uncertain SPM algorithm [14, 15]. Miliaraki et al use approximation with probabilistic guarantee to improve the efficiency of uncertain SPM problem [9]. A dynamic programming approach is used to mine probabilistic spatial-temporal frequent sequential patterns [8]. However, these methods are all designed for sequence databases with accurate timestamps.

Dyreson and Snodgrass introduced probabilistic temporal databases which models uncertain timestamp by a set of time points with equal probabilities [4]. Zhang et al. proposed a pattern recognition algorithm in temporal uncertain streams[13]; while pattern queries in temporal uncertain sequences is studied in [16]. However, our work distinguishes from the above in that we use uniform distributions to represent uncertain timestamps, which is more flexible in modeling data collected from different scales. Meanwhile, the above works focused on

sid	eid	T	I
1	1	[100,103]	{A,C}
1	2	[102,105]	B
2	1	[160,163]	A
2	2	[162,164]	B
2	3	[163,166]	B
2	4	[167,168]	C

Fig. 1. An example of uncertain database

sid	eid	t	I
1	1	102.5	{A,C}
1	2	103.9	B
2	1	163	A
2	2	162	B
2	3	165	B
2	4	166	C

Fig. 2. An example of a possible world

matching patterns in one sequence, while the SPM problem is more complicated because it mines patterns from a large number of uncertain sequences so that their techniques cannot be directly employed.

3 Problem statement

3.1 Temporal Uncertain Sequence Database

A temporal uncertain sequence database contains a collection of uncertain sequences, and an uncertain sequence is a set of temporal uncertain events. A temporal uncertain event is represented by $e = \langle sid, eid, T, I\rangle$. Here sid is the sequence-id, eid is the event-id and $\langle sid, eid\rangle$ identifies a unique event. Note that events are not guaranteed to be ordered by their eids. An uncertain timestamp T is modeled by a uniform distribution $T \sim U(t^-, t^+)$, where $[t^-, t^+]$ is the range of T. I is an itemset that describes the content of event e.

Fig. 1 shows an example of temporal uncertain database. A sequence is a list of events that are associated with the same sid and an event identified by $sid = i$ and $eid = j$ is denoted by e_{ij}. For example, $e_{11} = \langle 1, 1, \{[100, 103]\}, \{A, C\}\rangle$ indicates that event $\{A, C\}$ occurs at time T, where $T \sim U(100, 103)$ is uniformly distributed within 100 and 103.

3.2 Temporal Possible Worlds

We use possible world semantics to interpret temporal uncertain databases. Temporal possible worlds of an uncertain database D are generated by instantiating all possible values of each uncertain timestamp. Fig. 2 shows an example a temporal possible worlds that are instantiated from the uncertain database in Fig. 1, in which the time point of an event is randomly drawn from the corresponding uncertain timestamps.

It is widely assumed that uncertain sequences in D are mutually independent, which is known as the *tuple-level independence* [7, 1] in probabilistic databases. Meanwhile, event time are also assumed to be independent of each other [3, 5,

6, 14], which can be justified by the fact that events are often observed independently in real applications. Thus, the probability density function (pdf) of the possible words can be computed by Equation (1).

$$f(w) = \prod_{i=1}^{m} f(d_i) = \prod_{i=1}^{m} \prod_{j=1}^{n_i} f(T_{ij} = t_{ij}) \tag{1}$$

Where d_i is a sequence in the database D and e_{ij} is an event in d_i. $m = |D|$ is the number of sequences in D and $n_i = |d_i|$ is the number of events in d_i. Let $T_{ij} \sim U(t_{ij}^-, t_{ij}^+)$ be the uncertain time of event e_{ij}, then its pdf $f(T_{ij} = t_{ij})$ is shown in Equation (2).

$$f(T_{ij} = t_{ij}) = \begin{cases} \frac{1}{t_{ij}^+ - t_{ij}^-} & , t \in [t_{ij}^-, t_{ij}^+] \\ 0 & , \text{otherwise} \end{cases} \tag{2}$$

3.3 Uncertain Sequential Pattern Mining Problem

A sequential pattern $\alpha = \langle X_1 \cdots X_n \rangle$ is *supported* by a sequence $\beta = \langle Y_1 \cdots Y_m \rangle$, denoted by $\alpha \preceq \beta$, if and only if there exists integers $\{k_1, \ldots, k_n\}$ so that we have $X_i.I \subseteq Y_{k_i}.I$ $(\forall i \in [1, n])$ and $l \leq Y_{k_{i+1}}.t - Y_{k_i}.t \leq h$ $(\forall i \in [1, n-1])$. Here $l = mingap$ is the minimal time gap constraint between two adjacent events of α and $h = maxgap$ is the maximal time gap constraint.

In deterministic databases, a sequential pattern s is frequent if and only if it satisfies $sup(s) \geq t_s$, where $sup(s)$ is the total number of sequences that support s and t_s is the user-defined minimal threshold. However, In an uncertain database D, the frequentness of s is probabilistic and it can be computed by Equation (3).

$$P(sup(s) \geq t_s) = \int\limits_{sup(s|w) \geq t_s} f(w) \mathrm{d}w \tag{3}$$

Where w is a possible world in which s is frequent and $f(w)$ is the pdf of w.

The SPM problem in temporal uncertain databases can be defined as follows. *Given a minimal support t_s, a minimal frequentness probability threshold t_p, a minimal time gap l and a maximal time gap h, find every probabilistic frequent sequential pattern s in a temporal uncertain database, which has $P(sup(s) \geq t_s) \geq t_p$.*

4 Solution

4.1 Frequentness Probability

Suppose $D = \{d_1, \ldots, d_n\}$ is a temporal uncertain database and s is a sequential pattern. Because $d_1, \ldots, d_n$ in D are mutually independent, the probabilistic support of s in D, denoted by $sup(s)$, can be computed by Equation (4).

$$sup(s) = \sum_{i=1}^{n} sup(s|d_i) \tag{4}$$

Where $sup(s|d_i)$ $(\forall i \in [1, n])$ is a Bernoulli random variable, whose success probability is $P(sup(s|d_i) = 1) = P(s \preceq d_i)$.

$sup(s)$ is a Poisson-Binomial random variable, since it is the sum of n independent but non-identical Bernoulli random variables. And the probability mass function (pmf) of $sup(s)$ is $P(sup(s) = i) = p_i$, where p_i is the probability that the support of s in D equals to i $(i \in [1, |D|])$. Here we adopt the Fast Fourier Transform (FFT) technology in [14] to compute the pmf of $sup(s)$ in $O(nlogn)$ time . Thereafter, the *frequentness probability* of s is computed by Equation (5).

$$P(sup(s) \geq t_s) = \sum_{i \geq t_s} p_i = 1 - \sum_{i < t_s} p_i \tag{5}$$

Where, t_s is the minimal support threshold and $p_i = P(sup(s) = i)$ $(i \in [1, n])$ is the probability that the support of s in D equals to i. Given the minimal frequentness probability threshold t_p, s is probabilistically frequent if and only if $P(sup(s) \geq t_s) \geq t_p$.

4.2 Support Probability

We first define the *minimal possible occurrence* of a sequential pattern s in an uncertain sequence d.

Definition 1. *Given a sequential pattern s and an uncertain sequence d, a subset d' of d (e.g. $d' \subseteq d$) is called a minimal possible occurrence of s if and only if (1) $P(s \preceq d') > 0$; (2) $\forall d'' \subset d', P(s \preceq d'') = 0$.*

For example, in Fig. 1, $\{e_{21}, e_{22}\}$ and $\{e_{21}, e_{23}\}$ are two minimal possible occurrences of the sequential pattern $\langle \mathrm{A}, \mathrm{B} \rangle$ in the sequence s_2; while $\{e_{21}, e_{22}, e_{23}\}$ is not a minimal occurrence of $\langle \mathrm{A}, \mathrm{B} \rangle$. Then the *support probability* $P(s \preceq d)$ can be computed by Equation (6), since event timestamps are independent.

$$P(s \preceq d) = \sum_{i=1}^{N} P(s \preceq o_{s_i}) \tag{6}$$

Here o_{s_i} $(i = 1, \ldots, N)$ are N minimal possible occurrences of s, and the computation of $P(s \preceq o_{s_i})$ is discussed in section 4.3.

4.3 Probability of satisfying time constraints

Let $o_s = \{e_{k_1}, \ldots, e_{k_n}\}$ be a minimal possible occurrence of sequential pattern $s = \langle s_1, \ldots, s_n \rangle$. Suppose T_i is the uncertain time of the event e_{k_i}, then $P(s \preceq o_s)$, denoted by $P(\langle T_1 \cdots T_n \rangle)$, is the probability that $T_1, \cdots, T_n$ satisfy time

constraints $l \leq T_{i+1} - T_i \leq h$, $\forall i \in [1, n)$. Here l is the minimal time gap between two adjacent timestamps and h is the maximal time gap.

A naive approach of computing $P(\langle T_1 \cdots T_n \rangle)$ is to use the *chain rule*, which is shown in Equation (7).

$$\begin{aligned} P(\langle T_1 \cdots T_n \rangle) &= \int_{l \leq t_i - t_{i-1} \leq h} \cdots \int f(T_{k_1} = t_1, \ldots, T_{k_n} = t_n) \mathrm{d}t_1 \cdots \mathrm{d}t_n \\ &= \int_{l \leq t_i - t_{i-1} \leq h} \cdots \int f(t_n | t_1 \cdots t_{n-1}) \cdots f(t_2 | t_1) f(t_1) \mathrm{d}t_1 \cdots \mathrm{d}t_n \end{aligned} \tag{7}$$

However, this method is usually too complex in practice. Therefore, we design a new approach to compute $P(\langle T_1 \cdots T_n \rangle)$ efficiently.

Basic case. We first consider the basic case of two uncertain timestamps $X \sim U(x^-, x^+)$ and $Y \sim U(y^-, y^+)$. Given time constraints $mingap = l$, $maxgap = h$, $P(\langle XY \rangle)$ can be computed in Equation (8).

$$P(\langle XY \rangle) = \int_{max(y^-, x^- + l)}^{min(y^+, x^+ + h)} \int_{max(x^-, y - h)}^{min(x^+, y - l)} \frac{1}{(x^+ - x^-)(y^+ - y^-)} \mathrm{d}x \mathrm{d}y \tag{8}$$

Equation (8) is decomposed into p deterministic cases, if $[y^-, y^+]$ is divided into p disjoint subintervals by the endpoints $\{x^+ + l, x^- + l, x^+ + h, x^- + h\}$ as $[y^-, y^+] = \bigcup [y_i^-, y_i^+], \forall i \in [1, p]$. Here $Y_k \sim U[y_k^-, y_k^+]$ is a uniformly distributed random variable, and $P(\langle XY \rangle)$ can be computed by Equation (9).

$$P(\langle XY \rangle) = \sum_{k=1}^{p} P(\langle XY_k \rangle) P(Y = Y_k) \tag{9}$$

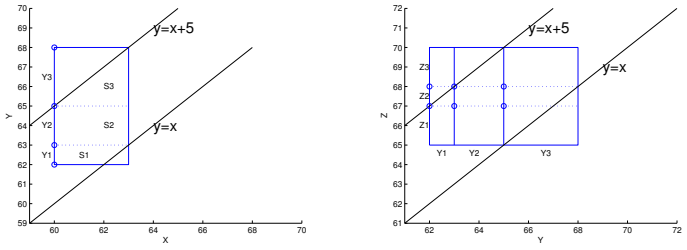
Where $P(Y = Y_k) = (y_k^+ - y_k^-)/(y^+ - y^-)$. We use a geographic method to compute $P(\langle XY_k \rangle)$ in $O(1)$ time, which is shown in Equation (10).

$$P(\langle XY_k \rangle) = \frac{S_k}{A_k} = \frac{(1/2) * (L_1 + L_2) * H}{(y_k^+ - y_k^l)(x^+ - x^-)} \tag{10}$$

Where, A_k is the area of the rectangle defined by the 2-dimensional uniform distribution of X and Y_k, and S_k is the area within A_k which satisfies the time constraints. Here $H = y_k^+ - y_k^-$, L_1 and L_2 are computed as follows.

$$\begin{aligned} L_1 &= max(0, L_1'), L_1' = min(y_k^- - l, x^+) - max(y_k^- - h, x^-) \\ L_2 &= max(0, L_2'), L_2' = min(y_k^+ - l, x^+) - max(y_k^+ - h, x^-) \end{aligned}$$

Fig. 3(a) shows an example of computing $P(\langle XY \rangle)$ with $l = 0$ and $h = 5$, where $X \sim \mathrm{U}[60, 63]$ and $Y \sim \mathrm{U}[62, 68]$. There two endpoints $\{63, 65\}$ within the range of Y, which divide $[62, 68]$ into three disjoint subintervals as $[62, 68] = [62, 63] \cup [63, 65] \cup [65, 68]$.



(a) compute $P(\langle XY \rangle)$ (b) compute $P(\langle XYZ \rangle)$

Fig. 3. An example of compute the probability of satisfying time constraints

Let $Y_1 \sim \mathrm{U}[62, 63]$, $Y_2 \sim \mathrm{U}[63, 65]$ and $Y_3 \sim \mathrm{U}[65, 68]$, then we have

$$\begin{aligned} &P(Y = Y_1) = \frac{1}{6} \quad P(\langle XY_1 \rangle) = \frac{S_1}{A_1} = \frac{2.5}{3} \quad P(\langle XY_1 \rangle \cap Y_1) = \frac{2.5}{18} \\ &P(Y = Y_2) = \frac{2}{6} \quad P(\langle XY_2 \rangle) = \frac{S_2}{A_2} = \frac{6}{6} \quad P(\langle XY_2 \rangle \cap Y_2) = \frac{1}{3} \\ &P(Y = Y_1) = \frac{3}{6} \quad P(\langle XY_3 \rangle) = \frac{S_3}{A_3} = \frac{4.5}{9} \quad P(\langle XY_3 \rangle \cap Y_3) = \frac{1}{4} \end{aligned} \tag{11}$$

Thereafter, $P(\langle XY \rangle) = \sum_{i=1}^{3} P(\langle XY_i \rangle \cap Y_i) = 0.72$.

General case. Given uniformly distributed uncertain timestamps $T_1, \ldots, T_n$, suppose the range of T_n is divided into p sub-partitions as $[t_n^-, t_n^+] = \bigcup_{i=1}^{p} [t_{n_i}^-, t_{n_i}^+]$, and $T_{n_i} \sim U(t_{n_i}^-, t_{n_i}^+)$ is a uniform distributed random variable, then we can compute $P(\langle T_1 \cdots T_n \rangle)$ by Equation (12).

$$\begin{aligned} P(\langle T_1, \ldots, T_n \rangle) &= \sum_{i=1}^{p} P(\langle T_1, \ldots, T_{n_i} \rangle) * P(T_n = T_{n_i}) \\ &= \sum_{i=1}^{p} P(\langle T_1, \ldots, T_{n_i} \rangle \cap T_{n_i}) \end{aligned} \tag{12}$$

Where $P(\langle T_1 \cdots T_{n_i} \rangle)$ can be computed by Equation (13).

$$\begin{aligned} P(\langle T_1, \ldots, T_{n_i} \rangle) &= \sum_{j=1}^{q} P(\langle T_1, \ldots, T_{(n-1)_j} \rangle) P(\langle T_{(n-1)_j} T_{n_i} \rangle) P(T_{(n-1)_j}) \\ &= \sum_{j=1}^{q} P(\langle T_{(n-1)_j} T_{n_i} \rangle) P(\langle T_1, \ldots, T_{(n-1)_j} \rangle \cap T_{(n-1)_j}) \end{aligned} \tag{13}$$

Let $s' = \langle s_1, \ldots, s_{n-1} \rangle$ be a sequential pattern by removing the last element of $s = \langle s_1, \ldots, s_n \rangle$. In SPM process, we have already computed $P(\langle T_1, \ldots, T_{(n-1)_j} \rangle \cap$

$T_{(n-1)_j}$) in searching s'. Thus, we can save and reuse these values when we search pattern s, in order to avoid repeated computation.

Given another uncertain time $Z \sim$ U[65, 70], Fig. 3(b) shows the process of computing $P(\langle XYZ\rangle)$ by reusing previous computational results. First, we compute potential end points by the ranges of Y_1, Y_2 and Y_3 as follows.

$$z_{11} = y_1^- + l = 62, z_{12} = y_1^+ + l = 63, z_{13} = y_1^- + r = 67, z_{14} = y_1^+ + h = 68$$
$$z_{21} = y_2^- + l = 63, z_{22} = y_2^+ + l = 65, z_{23} = y_2^- + r = 68, z_{24} = y_2^+ + h = 70$$
$$z_{31} = y_3^- + l = 65, z_{32} = y_3^+ + l = 68, z_{33} = y_3^- + r = 70, z_{34} = y_3^+ + h = 73$$

Therefore, the range of Z is divided into three disjoint sub-partitions as $[65, 70] = [65, 67] \cup [67, 68] \cup [68, 70]$. Let $Z_1 \sim$ U[65, 67], $Z_2 \sim$ U[67, 68] and $Z_3 \sim [68, 70]$. Here we take the computation of $P(\langle XYZ_1\rangle)$ in Equation (14) as an example.

$$P(\langle XYZ_1\rangle) = \sum_{i=1}^{3} P(\langle XY_i\rangle \cap Y_i)P(\langle Y_iZ_1\rangle) \tag{14}$$

Where $P(\langle XY_i\rangle \cap Y_i)$ is already computed in Equation (11). Referring to Equation (10), we can compute $P(\langle Y_1Z_1\rangle) = 1$, $P(\langle Y_2Z_1\rangle) = 1$, $P(\langle Y_3Z_1\rangle) = 1/3$. Thereafter, we have $P(\langle XYZ_1\rangle) = 1 * \frac{1}{6} * \frac{2.5}{3} + 1 * \frac{2}{6} * \frac{6}{6} + \frac{1}{3} * \frac{3}{6} * \frac{4.5}{9} = 0.5555$.

Similarly, we can compute $P(\langle XYZ_2\rangle) = 0.6111$ and $P(\langle XYZ_3\rangle) = 0.3333$. Therefore, we arrive to the final result $P(\langle XYZ\rangle) = 0.4 * 0.5555 + 0.2 * 0.6111 + 0.4 * 0.3333 = 0.4777$.

4.4 Uncertain SPM Algorithm

We integrate our uncertain management approach into the classic SPM algorithm PrefixSPan[11]. There are two major modifications to the original PrefixSpan in our uncertain SPM algorithm.

(1) We project the database by minimal possible occurrences. Suppose $o_s = \{e_{k_1}, \ldots, e_{k_n}\}$ is a minimal possible occurrence of s in sequence d. The projection of d w.r.t. o_s, denoted by $d|_{o_s}$, eliminates any event e_i in d if $P(e_i.T \geq e_{k_n}.T + mingap) = 0$. A projected database $D|_s = \{d_1|_{o_1,\ldots,o_p}, \ldots, d_t|_{o_1,\ldots,o_q}\}$ is a collection of projected sequences, where $d_i|_{o_1,\ldots,o_p} = \{d_i|_{o_1}, \ldots, d_i|_{o_p}\}$ is a set of p projected sequences of d_i w.r.t. to the minimal occurrences $o_1, \ldots, o_p$ of s in d_i. For example, in Fig. 1, if we set $mingap = 1$ and let $s = \langle$AB$\rangle$, then $D|_s = \{d_2|_{o_1,o_2}\}$, where $d_2|_{o_1} = \{e_{23}, e_{24}\}$ and $d_2|_{o_2} = \{e_{24}\}$.

(2) We save intermediate computational results for each minimal possible occurrence. Let $o_s = \{e_{k_1}, \ldots, e_{k_n}\}$ be a minimal possible occurrence of s in d and $T_i = e_{k_i}.T\ \forall i \in [1, n]$. Suppose the range $[t_n^-, t_n^+]$ of T_n is divided into k subintervals $[t_n^-, t_n^+] = \bigcup [t_{n_i}^-, t_{n_i}^+]$ ($\forall i \in [1, k]$), then we compute $p_i = P(\langle T_1, \ldots, T_{n_i}\rangle)$ by Equation (12) and save the results in the form as $T(o_s) = \{[t_{n_1}^-, t_{n_1}^+] : p_1, \ldots, [t_{n_k}^-, t_{n_k}^+] : p_k\}$. Therefore, we can reuse $T(o_s)$ in searching longer sequences.

We adopt the pattern-growth approach to search new patterns in Algorithm 1. We first mine frequent items in $D|_s$, denoted by $I = \{i_1, i_2, ..., i_n\}$. This process

ALGORITHM 1: USPM$(s, D|_s)$

Input: sequential pattern s, uncertain projected database $D|_s$,
$minsup = t_s$, $minprob = t_p$
Output: L: a set of frequent sequential patterns
Find all frequent items $I = \{i_1, i_2, ..., i_n\}$ in $D|_s$
if $D|_s = \phi$ *or* $I = \phi$ **then**
return L
end
foreach *item* $i \in I$ **do**
$s' = s + \{i\}$
for $d_i|_{o_1,...,o_n} \in D|_s$ **do**
for $d_i|_{o_j} \in d_i|_{o_1,...,o_n}$ **do**
construct a projected sequence $d_i|_{o_{s'}}$ from $d_i|_{o_j}$
compute $T(o_{s'})$ from $T(o_{s_j})$ in d_i by Equation (12) and Equation (13)
end
compute the support probability $P(s' \preceq d_i)$ by Equation (6).
end
use FFT to compute the Poisson Binomial distribution of $sup(s')$
if $P(sup(s') \geq t_s) \geq t_p$ **then**
$L = L \cup \{s'\}$;
USPM$(D|_{s'}, s')$;
end
end

is straightforward because it does not need to consider temporal uncertainty. A candidate pattern $s' = s + \{i\}$ is generated for each $i \in I$. Then, we extract all minimal possible occurrences of s' and construct their projected databases. For each minimal possible occurrence $o_{s'}$ of s', we compute and save its probability of satisfying time constraints by Equation (12) and Equation (13).

We compute the support probability of s' in each uncertain sequence by Equation (6). Thereafter, we adopt the FFT technique in [14] to compute the Poisson Binomial distribution of the overall support $sup(s')$. The frequentness probability is computed in Equation (5), by which we can determine if s' is a probabilistic frequent sequential pattern. The searching process stops until no frequent patterns are mined.

5 Evaluation

5.1 Synthetic data generation

We use the IBM market-basket data generator [2] to generate synthetic sequence datasets in different scales with the following parameters: (1) C: number of sequences; (2) T: average number of transactions/itemsets per data-sequence; (3) L: average number of items per transaction/itemset per data-sequence; (4) I: number of different items.

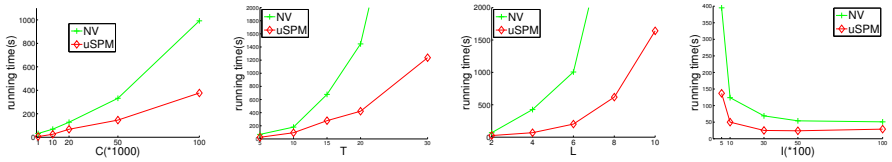


Fig. 4. Scalability of uSPM in synthetic uncertain datasets

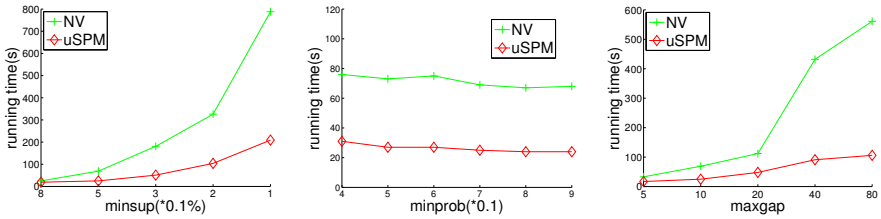


Fig. 5. Effect of parameters in synthetic uncertain datasets

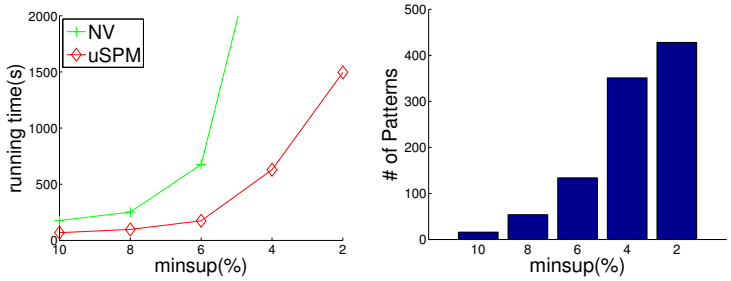
To add temporal uncertainty, we replace a point-value timestamp t in the original synthetic datasets by a uniform distribution in $[(1-r)*t, (1+r)*t]$, where r is randomly drawn from the uniform distribution $U(0,1)$. We name the generated synthetic dataset by parameters. For example, the dataset named $T4L10I1C10$ indicates that $T=4$, $L=10$, $I=1*1000$ and $C=10*1000$.

Our uncertain sequential pattern mining algorithm is called *uSPM* for short. Recall from Section 4.3 that a naive method to compute the probability of an occurrence satisfying time constraints is to directly evaluate Equation (7) using chain rule. This naive method is implemented and abbreviate as NV. We compare uSPM with NV to evaluate the performance of our algorithm. All the experiments were done in the desktop with Intel(R) Core (TM) Duo CPU @ 2.33GHz and 4GB memory.

5.2 Scalability and efficiency

In Fig. 4, we compare the running time of uSPM and NV on synthetic datasets with different scales, where we set $minsup = 0.5\%$, $minprob = 0.7$, $mingap = 1$, and $maxgap = 10$. We initially have $C = 10\,000$, $T = 4$, $I = 10\,000$ and $L = 2$. In Figigure 4(a), C varies from $1\,000$ to $100\,000$; In Fig. 4(b), T varies from 5 to 30; In Fig. 4(c), L varies from 2 to 10; and In Fig. 4(c) I varies from 500 to $10\,000$.

In Fig. 4, we observe the following phenomenons: (1) uSPM is significantly faster than NV under every setting of the parameters, which proves the effectiveness of our temporal uncertainty management approach. (2) The running time increases with the increment of C, T, L, as the increment of these parameters



(a) running time (b) number of patterns

Fig. 6. Performance of uSPM in real stock dataset

generates larger synthetic datasets. (3) The running time drops slightly with the increment of I, because there are less repeated items in sequences when I is set to a larger value.

Fig. 5 compares the running time of uSPM and NV with different of user-defined parameters in the dataset T4L2I10C10. We initially set $minsup = 0.2\%$, $minprob = 0.7$, $mingap = 1$, and $maxgap = 10$. In Fig. 5(a), $minsup$ decreases from 0.8% to 0.1%; in Fig. 5(b), $minprob$ varies from 0.4 to 0.9; and $maxgap$ varies from 5 to 80 in Fig. 5.

In Fig. 5, we observe that: (1) The running time of uSPM increase with the decrement of $minsup$; however, the performance is relatively stable to the variations of $minprob$. The probabilistic support of a sequential pattern is bounded to its expected value (*Chernoff bound*) so that the frequentness of a large number of patterns become deterministic. This explains why the running time of uSPM does not fluctuate significantly in Fig. 5(b). (3) The running time of uSPM increases when we set a larger value to $maxgap$. This is intuitive because a larger $maxgap$ indicates a less strict constraint of sequential patterns.

We also apply uSPM to a real world stock market dataset. The prices for 882 stocks are extracted from Shanghai Stock Exchange Center in 16 weeks from 12-03-2012 to 03-24-2013. Each stock corresponds to a sequence. We define three events such as price going up (+), going down (−) and no change (0). An uncertain event is aggregated from consecutive events. For example, if price goes up at time 1, 2 and 3, then we aggregate them to form an uncertain event ([1,3], +).

Here we set $minprob = 0.7$, $mingap = 1$ and $maxgap = 5$. Fig. 6(a) shows that the running time of uSPM in the stock dataset increases with the decrement of $minsup$. As we only define three distinct events, there are many repeated items in sequences; however, uSPM still significantly outperforms NV in this dataset. In Fig. 6(b), we can see that the number of frequent sequential patterns in the stock dataset increases significantly when we decrease the value of $minsup$ from 10% to 2%. And a mined pattern $\langle +, -, +, - \rangle$ from this dataset reveals that stock prices are fluctuated in general during the time when data are collected, which is consistent with intuitive observations.

6 Conclusion

In this paper, we study the problem of mining probabilistic frequent sequential patterns in databases with temporal uncertainty. We design an incremental approach to manage temporal uncertainty efficiently and integrate it into classic pattern-growth SPM algorithm. The experimental results prove that our algorithm is efficient and scalable.

References

1. C. C. Aggarwal and P. S. Yu. A survey of uncertain data algorithms and applications. *IEEE Trans. on Knowl. and Data Eng.*, 21(5):609–623, May 2009.
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, pages 487–499, 1994.
3. T. Bernecker, H.-P. Kriegel, M. Renz, F. Verhein, and A. Zuefle. Probabilistic frequent itemset mining in uncertain databases. In *SIGKDD*, pages 119–128, 2009.
4. C.E.Dyreson and R.T.Snodgrass. Supporting valid-time indeterminacy. In *TODS*, 1998.
5. C. Chui and B. Kao. A decremental approach for mining frequent itemsets from uncertain data. In *PAKDD*, pages 64–75, 2008.
6. C. Chui, B. Kao, and E. Hung. Mining frequent itemsets from uncertain data. In *PAKDD*, pages 47–58, 2007.
7. J. Jestes, G. Cormode, F. Li, and K. Yi. Semantics of ranking queries for probabilistic data. *IEEE Transactions on Knowledge and Data Engineering*, 23(12):1903–1917, 2011.
8. Y. Li, J. Bailey, L. Kulik, and J. Pei. Mining probabilistic frequent spatio-temporal sequential patterns with gap constraints from uncertain databases. In *ICDM*, pages 448–457, 2013.
9. I. Miliaraki, K. Berberich, R. Gemulla, and S. Zoupanos. Mind the gap: Large-scale frequent sequence mining. In *SIGKDD*, pages 797–808, 2013.
10. M. Muzammal and R. Raman. Mining sequential patterns from probabilistic databases. In *PAKDD*, pages 210–221, 2011.
11. J. Pei, J. Han, B. Mortazavi-asl, H. Pinto, Q. Chen, U. Dayal, and M. chun Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *ICDE*, pages 215–224, 2001.
12. Y. Tong, L. Chen, Y. Cheng, and P. S. Yu. Mining frequent itemsets over uncertain databases. In *Proceeding of the VLDB Endowment*, volume 5, pages 1650–1661, 2012.
13. H. Zhang, Y. Diao, and N. Immerman. Recognizing patterns in streams with imprecise timestamps. *Proc. VLDB Endow.*, 3(1-2):244–255, 2010.
14. Z. Zhao, D. Yan, and W. Ng. Mining probabilistically frequent sequential patterns in uncertain databases. In *EDBT*, pages 74–85, 2012.
15. Z. Zhao, D. Yan, and W. Ng. Mining probabilistically frequent sequential patterns in large uncertain databases. In *IEEE Transactions on Knowledge and Data Engineering*, volume 26, pages 1171–1184, 2013.
16. Y. Zhou, C. Ma, Q. Guo, L. Shou, and G. Chen. Sequence pattern matching over time-series data with temporal uncertainty. In *EDBT*, pages 205–216, 2014.