Mining Unexpected Sequential Patterns and Implication Rules

Haoyuan Li, Anne Laurent, Pascal Poncelet

Mining Unexpected Sequential Patterns and Implication Rules

Dong (Haoyuan) Li
LGI2P, École des Mines d'Alès, France

Anne Laurent
LIRMM, Université Montpellier II, France

Pascal Poncelet
LIRMM, Université Montpellier II, France

ABSTRACT

As common criteria in data mining methods, the frequency-based interestingness measures provide a statistical view of the correlation in the data, such as sequential patterns. However, when we consider domain knowledge within the mining process, the unexpected information that contradicts existing knowledge on the data has never less importance than the regularly frequent information. For this purpose, we present the approach USER for mining unexpected sequential rules in sequence databases. We propose a belief-driven formalization of the unexpectedness contained in sequential data, with which we propose 3 forms of unexpected sequences. We further propose the notion of unexpected sequential patterns and implication rules for determining the structures and implications of the unexpectedness. The experimental results on various types of data sets show the usefulness and effectiveness of our approach.

KEYWORDS

Sequence Data Mining, Interestingness Measure, Belief, Unexpected Sequence, Unexpected Sequential Pattern, Unexpected Sequential Implication Rule.

INTRODUCTION

Most real world applications process the data stored in sequence format, where the elements in data are sequentially ordered with temporal or spatial relation. For instances, in a customer retail database, a sequence can be all purchases of a customer ordered by the time of transaction; in a Web access log file, a sequence can be all of those resources accessed during a user session ordered by the time of request; in a telecommunication network monitoring database, a sequence can be all events during a period ordered by the time of occurrence; in a DNA segment, a sequence is a succession of nucleotide subunits with spatial order, etc. In order to discover the knowledge hidden in such sequential data, sequence data mining techniques (Dong & Pei, 2007;

Han & Kamber, 2006) have been highly developed and widely applied in many application domains.

As one of the most important models of sequence data mining, the sequential pattern proposed by Agrawal and Srikant (1995) provides a statistical frequency based view of the correlations between the elements in sequential data. The problem of mining sequential patterns can be formally described as follows.

Given a set of binary-valued attributes $R = \{i_1,i_2,\ldots,i_n\}$, an attribute is an *item*. An *itemset*, denoted as $I = (i_1 i_2 \ldots i_m)$, is an unordered collection of items. A *sequence* is an ordered list of itemsets, denoted as $s = \langle I_1 I_2 \ldots I_k \rangle$. A *sequence database*, denoted as *D*, is a large set of sequences. Given two sequences $s = \langle I_1 I_2 \ldots I_m \rangle$ and $s' = \langle I'_1 I'_2 \ldots I'_n \rangle$, if there exist integers $1 \le i_1 \le i_2 \le \ldots \le i_m \le n$ such that $I_1 \subseteq I'_{i1}$, $I_2 \subseteq I'_{i2}$, …, $I_m \subseteq I'_{im}$, then the sequence *s* is a *subsequence* of the sequence *s'* and the sequence *s'* is a *super sequence* of the sequence *s*, denoted as $s \sqsubseteq s'$, and we say that the sequence *s is included in* the sequence *s'*, or the sequence *s' supports* the sequence *s*. If a sequence *s* is not included in any other sequences, then the sequence *s* is a *maximal sequence*. The *support* (or the *frequency*) of a sequence *s* in a sequence database *D*, denoted as $\sigma(s, D)$, is the fraction of the total number of sequences in the database *D* that support *s*. Given a minimal frequency threshold *minimum support* specified by user, denoted as σ_{min}, a sequence *s* is *frequent* if $\sigma(s, D) \ge \sigma_{min}$. A *sequential pattern* is a frequent maximal sequence, so that the problem of *mining sequential patterns* is to find all frequent maximal sequences in a sequence database.

Example 1. Let D be a customer retail database, with the minimum support σmin = 0.5, we may find the sequential pattern s = ⟨(Sci-Fi-Novel)(Action-Film Sci-Fi-Film)(Rock-Music)⟩ where σ (s, D) = 0.6, which can be interpreted as "60% of customers purchase a Sci-Fi novel, then purchase action and Sci-Fi films later, and then purchase a rock music CD". ■

Example 2. Let *D* be a Web access log database, with the minimum support σ_{min} = 0.5, we may find the sequential pattern $s = \langle (login)(msglist)(msgread)(msgread)(logout) \rangle$ where $\sigma(s, D)$ = 0.8, which can then be interpreted as "*80% of users visit the login page, then visit the message list page, then read messages, and at last logout*". ■

Up to now, a great deal of research work focuses on effectively mining sequential patterns (Ayres et al, 2002; Li et al, 2007; Masseglia et al, 1998; Pei et al, 2004; Srikant & Agrawal, 1996; Zaki, 2001) and the variances (Garofalakis et al, 1999; Lo et al, 2007; Mannila et al, 1997; Wang & Han, 2004; Yan et al, 2003). With sequential pattern mining, we can extract the sequences that reflect the most general behaviors within the context of sequential data, which can be further interpreted as domain knowledge for different purposes. However, although sequential patterns are essential for behavior recognition, when we consider domain knowledge within the mining process, the unexpected sequences that contradict existing knowledge on the data have never less importance than the frequent sequences. On the other hand, such unexpected sequences do not mean that they cannot be frequent, so that there exist following problems in discovering the unexpectedness in data with the frequency-based interestingness measures.

First, the redundancy problem of frequency-based data mining methods undermines many real world applications where the exponential pattern or sequence sets generated by mining processes make the post analysis extremely hard. Hence, the identification of unexpected information might be impossible when the support of such unexpected sequences, within the context of sequence

data mining, is very low such that the unexpectedness may be hidden in millions of sequential patterns.

Example 3. Let us consider the instance illustrated in Example 1. Assume that in the database *D*, there exist 6% of customers who purchase a Sci-Fi novel then action and Sci-Fi films, purchase a classical music CD instead of a rock music CD. This behavior is unexpected to the frequent behavior described in Example 1 and can be interesting for product promotion. In fact, with sequential pattern mining, we are able to find such a behavior only if the minimum support threshold is no greater than 0.06. However, with σ_{min} = 0.06, the result sequence set of all sequential patterns *s* such that $\sigma(s, D) \geq 0.06$ might be very large and that makes it impossible to identify the above behavior. ■

Secondly, if an unexpected sequence is "incomplete" in comparison with an expected sequence, it is impossible to determine the former with classical sequential pattern mining: according to the definition of sequential pattern, the former is included in the latter so that the former will not appear in the result sequence set while the latter is frequent. The following example illustrates this issue (notice that we do not strictly indicate the difference between *sequence* and *sequential pattern*, however, we use the term *sequence* for a full sequence contained in the database, and the term *sequential pattern* for a potentially frequent part of a full sequence contained in the database).

Example 4. Considering again the Web access log database *D* illustrated in Example 2, let the sequential pattern s_0 = ⟨(*login*)(*msglist*)(*logout*)⟩ be an expected access sequence with respect to the workflow of the service, where we do not require the access of the resource "*msgread*" in the workflow since there can be no new unread messages for a user. Assume that the sequence *s* = ⟨(*login*)(*logout*)⟩ is unexpected to the workflow s_0 and it is caused by failing to list all messages of a user. Let s_1 = ⟨(*login*)(*msglist*)(*msgread*)(*msgread*)(*logout*)⟩ and s_2 = ⟨(*login*)(*option*)(*password*)(*logout*)⟩ be two sequential patterns (in order to simplify the example, s_1 and s_2 are not included in a same sequence), then we have $\sigma(s, D) \geq \sigma(s_0, D) \geq \sigma(s_1, D)$ and $\sigma(s, D) \geq \sigma(s_2, D)$. Assume that s_1 and s_2 are the only sequential patterns other than s_0 that include *s*, then we can conclude the existence of the unexpected sequence *s* if and only if $\sigma(s, D) > \sigma(s_1, D) + \sigma(s_2, D)$. Nevertheless, if *s* is unknown, then we have to examine the support values of all possible combinations of subsequences of s_0, s_1 and s_2 for seeking the unexpected information, and the computation and identification tasks will become extremely hard. ■

The complex constraint based approaches like SPIRIT proposed by Garofalakis et al (1999) can find the unexpected sequences, however the premise is that we must know the composition of an unexpected sequence before the extraction, and an important drawback is that we cannot find all sequences representing the behavior. The closed sequential pattern mining (Yan et al, 2003) may tell the existence of the unexpected one by computing the difference of the support values of all sequences that include the unexpected sequence, however, only if we have already known what the unexpected sequences are, we have to seek the unexpectedness in the result set of all possible combinations of candidate unexpected sequences.

In this chapter, we propose a novel approach USER (Mining unexpected sequential rules) for finding unexpected sequential rules in large sequence databases. Furthermore, when we consider

the unexpectedness in sequential data, we are interested not only in the internal structures, but also in the premises and consequences represented as rules on the discovered unexpected sequences. Such rules are important to a lot of real world applications, especially to the early prediction of critical events or behaviors in the domains such as telecommunication network monitoring, credit card fraud detection, financial risk investigation, DNA segment analysis, and so on. Notice that our goal is not to find infrequent rules from sequence databases, but to find the rules disclosing the information that contradicts existing knowledge.

The rest of this chapter is organized as follows. Section 2 the related work on unexpected pattern and sequence mining. Section 3 presents the discovery of unexpected sequences and sequential implication rules for determining the unexpectedness in sequence databases. Section 4 shows the results of the experimental evaluation of our approach on real data and synthetic data for testing the effectiveness and the scalability. Finally, we discuss our further research direction and we conclude in Section 5.

RELATED WORK

In this chapter, we propose a subjective measure for sequence data mining. McGarry (2005) systematically investigated the interestingness measures for data mining, which are classified into two categories: the objective measures based on the statistical frequency or properties of discovered patterns, and the subjective measures based on the domain knowledge or the class of users. Silberschatz and Tuzhilin (1995) studied the subjective measures, in particular the unexpectedness and actionability. The term *unexpectedness* stands for the newly discovered (sequential) patterns that are surprising to users. For example, if most of the customers who purchase Sci-Fi movies purchase rock music, then the customers who purchase Sci-Fi movies but purchase classical music are unexpected. The term *actionability* stands for reacting to the discovered (sequential) patterns to user's advantage. For example, for the customers who purchase Sci-Fi movies without purchasing any kind of music, it is actionable to improve the promotion of rock music, even though it is unexpected. Therefore, in many cases, the unexpectedness and actionability exist at the same time, however, clearly, some actionable (sequential) patterns can be expected and some unexpected (sequential) patterns can also be non-actionable (Silberschatz & Tuzhilin, 1995).

Silberschatz and Tuzhilin (1995) further introduced two types of beliefs, *hard belief* and *soft belief*, for addressing unexpectedness. According to authors' proposition, the hard belief is a belief that cannot be changed by new evidences in data, and any contradiction of such a belief implies data error. For example, in the Web access log analysis, the error "404 Not Found" can be considered as a contradiction of a head belief: "the resources visited by users must be available"; however, the soft belief corresponds to the constraints on data that are measured by a degree, which can be modified with new evidences in data that contradict such a belief and interestingness of new evidences is measured by the change of the degree. For example, when more and more users visit the Web site at night, the degree of the belief "users access the Web site at day time" will be changed. The computation of the degree can be handled by various methods, such as the Bayesian approach and the conditional probability.

With the unexpectedness measure, Padmanabhan and Tuzhilin (1998) propose a belief-driven approach for finding unexpected association rules. In that approach, a belief is given from association rule, and the unexpectedness is stated by semantic contradictions of patterns. Given a belief $X \rightarrow Y$, a rule $A \rightarrow B$ is unexpected if: (1) the patterns B and Y semantically contradict each

other; (2) the support and confidence of the rule $A \bigcup X \rightarrow B$ hold in the data; (3) the support and confidence of the rule $A \bigcup X \rightarrow Y$ do not hold in the data. The discovery process is performed within the framework of the *a priori* algorithm.

Spiliopoulou (1999) proposed an approach for mining unexpectedness with sequence rules transformed from frequent sequences. The sequence rule is built by dividing a sequence into two adjacent parts, which are determined by the *support*, *confidence* and *improvement* from association rule mining. A belief on sequences is constrained by the frequency of the two parts of a rule, so that if a sequence respects a sequence rule but the frequency constraints are broken, then this sequence is unexpected. Although this work considers the unexpected sequences and rules, it is however very different to our problem in the measure and the notion of unexpectedness contained in data.

The outlier mining focuses on finding infrequent patterns in data with objective measures of interestingness, which are mostly distance-based (Angiulli & Pizzuti, 2002, 2005; Jin et al, 2001; Knorr & Ng, 1998; Ramaswamy et al, 2000). The study of outlier mining in sequential data is very limited. To the best of our knowledge, the approach proposed by Sun et al (2006) is currently the unique one. In our approach, the unexpectedness is stated by the semantics and temporal occurrences, instead of the statistical frequency or distance. Moreover, we concentrate on finding sequential implication rules of unexpectedness, which is not covered by outlier mining. For these meanings, we consider the unexpectedness within the context of domain knowledge and the aspect "*valid*" within the contact of the classical notions of support and confidence.

MINING IMPLICATION RULES IN UNEXPECTED SEQUENCES

Let $s_\alpha \rightarrow s_\beta$ be a *sequential rule* of sequences, where s_α, s_β are two sequences. Let τ be the constraint on the number of itemsets, or the *occurrence distance*, between the sequences s_α and s_β. Let η be the constraint on the semantics of sequences that $s_\beta \neq_{sem} s_\gamma$, where s_γ is a sequence that semantically opposite to the sequence s_β. A *belief* is considered as a sequential rule and the constraints τ and η on the rule. A sequence s is *unexpected* if s contradicts a belief.

We concentrate on finding the premises that possess the unexpectedness in sequences and the consequences engendered in the sequential data. In this section, we present the discovery of unexpected sequences and sequential implication rules for determining the unexpectedness in sequence databases.

Belief Base

In order to construct the belief base for mining unexpected sequences, let us first introduce some additional notions on sequential data.

The *length* of a sequence s is the number of itemsets contained in the sequence, denoted as $|s|$. An *empty sequence* is denoted as ϕ, where $|\phi| = 0$. The *concatenation* of sequences is denoted as the form $s_1 \cdot s_2$, so that we have $|s_1 \cdot s_2| = |s_1| + |s_2|$. We denote $[s$ the first itemset of the sequence s, and $s]$ the last itemset of the sequence s. For two sequences s and s' such that $s \sqsubseteq s'$, we note $s \sqsubseteq^{[} s'$ if we have $[s \subseteq [s'$, note $s \sqsubseteq^{]} s'$ if we have $s] \subseteq s']$, and note $s \sqsubseteq^{[\cdot]} s'$ if we have $[s \subseteq [s'$ and $s] \subseteq$

s']. We denote $s \sqsubseteq_c s'$ that the sequence s is a *consecutive subsequence* of the sequence s'. For example, we have $\langle(a)(b)(c)\rangle \sqsubseteq_c \langle(b)(\underline{a})(\underline{ab})(\underline{c})(d)\rangle$, but $\langle(a)(b)(c)\rangle \not\sqsubseteq_c \langle(\underline{a})(\underline{b})(ab)(\underline{c})(d)\rangle$.

Given sequences s, s_1 and s_2 such that $s_1 \cdot s_2 \sqsubseteq s$, the *occurrence relation*, denoted as $\mapsto^{\tau}$, is a relation r between the occurrences of s_1 and s_2 in s, where $\tau = [min..max]$ ($min, max \in \mathbb{N}$ and $min \leq max$) is the constraint on the occurrence distance between s_1 and s_2. Let $|s| \vDash [min..max]$ (or $|s| \vDash \tau$) denote that the length of the sequence s satisfies the constraint $[min..max]$, that is, $min \leq |s| \leq max$, then the relation $s_1 \mapsto^{\tau} s_2$ represents

$$s_1 \cdot s_2 \sqsubseteq s' \Rightarrow \left(s_1 \cdot s \cdot s_2 \sqsubseteq_c s'\right) \wedge \left(|s| \vDash \tau\right).$$

When max is not specified (or cannot be specified, like $max = \infty$), we note max as $*$, that is, $\tau = [min..*]$. In the particular cases, for $min = max = 0$, we note $s_1 \mapsto^{[0..0]} s_2$ as $s_1 \mapsto s_2$; for $min = 0$ and $max = *$, we note $s_1 \mapsto^{[0..*]} s_2$ as $s_1 \mapsto^{*} s_2$. Given a sequence s and an occurrence relation r, we note $s \vDash r$ if the sequence s satisfies the relation r.

Example 5. Given an occurrence relation $r = \langle(a)\rangle \mapsto^{[1..2]} \langle(c)\rangle$, we have $\langle(a)\_(c)\rangle \nvDash r$, $\langle(a)(b)(c)\rangle \vDash r$, $\langle(a)(b)(b)(c)\rangle \vDash r$, $\langle(a)(be)(b)(c)\rangle \vDash r$, and $\langle(a)(\underline{b})(\underline{b})(\underline{b})(c)\rangle \nvDash r$. ■

Given a sequential rule $s_\alpha \rightarrow s_\beta$, the semantic constraint $s_\beta \neq_{sem} s_\gamma$ requires that the occurrence of the sequence s_β should not be replaced by the occurrence of the sequence s_γ, since s_β and s_γ are semantically opposite to each other. That is, with this meaning, since the rule $s_\alpha \rightarrow s_\beta$ can be interpreted as the implication $s_\alpha \sqsubseteq s \Rightarrow s_\alpha s_\beta \sqsubseteq s$, according to $s_\beta \neq_{sem} s_\gamma$ we have the implication $s_\alpha \sqsubseteq s \Rightarrow s_\alpha s_\gamma \not\sqsubseteq s$. Moreover, considering the semantic constraint η together with the occurrence constraint τ, we have the following relation:

$$s_\alpha \sqsubseteq s \Rightarrow \left(s_\alpha \cdot s' \cdot s_\beta \sqsubseteq_c s\right) \wedge \left(s_\alpha \cdot s' \cdot s_\gamma \not\sqsubseteq_c s\right) \wedge \left(|s'| \vDash \tau\right).$$

From these constraints, we define the belief on user behaviors as follows.

Definition 1. A *belief* on sequences consists of a sequential rule $s_\alpha \rightarrow s_\beta$, an occurrence constraint $\tau = [min..max]$ ($min, max \in \mathbb{N}$ and $min \leq max$), and a semantic constraint $\eta : s_\beta \neq_{sem} s_\gamma$ on the rule, denoted as $b = [s_\alpha; s_\beta; s_\gamma; min..max]$, such that for any sequence s satisfies the belief b, denoted as $s \vDash b$, we have that $s_\alpha \sqsubseteq s$ implies $s_\alpha s' \cdot s_\beta \sqsubseteq_c s$ and $s_\alpha s' \cdot s_\gamma \not\sqsubseteq_c s$, where $|s'| \vDash \tau$. ■

Beliefs can be generated from existing domain knowledge on common behaviors of the data, or from the predefined workflows. Let us examine the Example 3 and 4 for illustrating how beliefs work.

Example 6. Let us consider Example 3. According to customer purchase behaviors, we first create the sequential rule $\langle(Sci\text{-}Fi\text{-}Novel)(Action\text{-}Film\ Sci\text{-}Fi\text{-}Film)\rangle \rightarrow \langle(Rock\text{-}Music)\rangle$, which indicates that the purchases of a Sci-Fi novel then action and Sci-Fi films later imply the purchase of a rock music CD. If we just expect that a purchase of rock music CD should be performed after

the precedent purchases, then the following belief can be established for describing this requirement:

$$[\langle(\textit{Sci-Fi-Novel})(\textit{Action-Film Sci-Fi-Film})\rangle; \langle(\textit{Rock-Music})\rangle; \phi; 0..*],$$

where the position of the sequence s_γ is empty since at this moment we are not yet taking the semantic opposition into account.

Now we consider the classical music to be semantically opposite to the rock music, then we have the semantic constraint as $\langle(\textit{Rock-Music})\rangle \neq_{sem} \langle(\textit{Classical-Music})\rangle$, then the above belief can be rewritten as follows:

$$[\langle(\textit{Sci-Fi-Novel})(\textit{Action-Film Sci-Fi-Film})\rangle; \langle(\textit{Rock-Music})\rangle; \langle(\textit{Classical-Music})\rangle; 0..*].$$

Moreover, if the customer transaction records show that most of customers purchase a rock music CD in a short delay after purchasing a Sci-Fi novel then action and Sci-Fi films, for example in the next 3 to 5 purchases, then the second belief can be further rewritten as:

$$[\langle(\textit{Sci-Fi-Novel})(\textit{Action-Film Sci-Fi-Film})\rangle; \langle(\textit{Rock-Music})\rangle; \langle(\textit{Classical-Music})\rangle; 3..5].$$ ■

Example 7. Considering Example 4, the user access sequence $\langle(\textit{login})(\textit{msglist})(\textit{logout})\rangle$ is expected to be frequent. According to the workflow of the Web site, the following rules can be generated: $\langle(\textit{login})(\textit{msglist})\rangle \rightarrow \langle(\textit{logout})\rangle$ and $\langle(\textit{login})\rangle \rightarrow \langle(\textit{logout})\rangle$, respectively with the occurrence constraints $[0..*]$ and $[1..*]$, since the access of *logout* should not just be after the access of *login*. Hence, we have the following beliefs without semantic constraint:

$$[\langle(\textit{login})(\textit{msglist})\rangle; \langle(\textit{logout})\rangle; \phi; 0..*] \text{ and } [\langle(\textit{login})\rangle; \langle(\textit{logout})\rangle; \phi; 1..*].$$

In order to constrain the relation between *login* and *logout*, the above two beliefs can be rewritten as:

$$[\langle(\textit{login})\rangle; \langle(\textit{msglist})\rangle; \langle(\textit{logout})\rangle; 0..0],$$

where *logout* is semantically opposite to *msglist* according to the access of *login*. Other user behaviors can also be represented by beliefs. The following belief,

$$[\langle(\textit{login})(\textit{msglist})\rangle; \langle(\textit{msgread})\rangle; \langle(\textit{logout})\rangle; 0..5]$$

depicts that we expect that users will not logout to the system too early, for example, after at least 5 visits of other resources. ■

In our approach, we consider the *consistent belief* in the semantics, that is, for any belief $b = [s_\alpha; s_\beta; s_\gamma; min..max]$, we have $s_\gamma \not\sqsubseteq s_\beta$. For example, considering a belief $b = [\langle(a)\rangle; \langle(b)(c)\rangle; \langle(c)\rangle; 0..3]$, although we cannot assert that the sequence $\langle(b)(c)\rangle$ is not semantically opposite to the sequence $\langle(c)\rangle$ (such as (b) stands for "*not*" and (c) stands for "*good*" within the context of text mining), such a belief is rather ambiguous in the semantics: since $\langle(c)\rangle \sqsubseteq \langle(b)(c)\rangle$, we can say that $\langle(c)\rangle$ is more general than $\langle(b)(c)\rangle$, which means that *the unexpectedness is more general than the expectedness* in a sequence. In this case, any unexpected sequence is always expected. Notice that our goal is to find the unexpectedness, but not the expectedness, so that the inverse, that *the expectedness is more general than the unexpectedness*, is allowed in a sequence. Obviously, as to be consistent, the semantics of two beliefs in the same belief base must not contradict each other. For instance, the beliefs $[s_1; s_2; s_3; min..max]$ and $[s_1; s_3; s_2; min..max]$ semantically contradict each other, and must be take into account during the construction of a belief base.

A *belief base*, denoted as *B*, is a set of consistent beliefs, which are stored in a prefix represented *belief tree*, denoted as *T*, defined below.

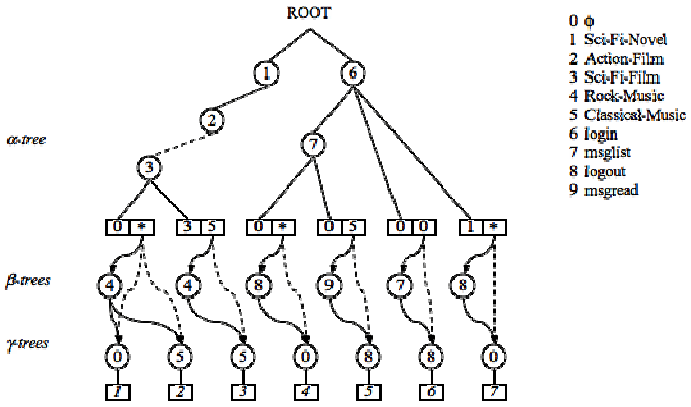

Figure 1. The belief tree for Example 6 and 7.

1. The belief tree *T* consists of one root node and three groups of sub-trees (*α-tree*, *β-trees* and *γ-trees*) as the children of the root, where each group respectively represents the s_α, s_β, and s_γ parts of each belief $b \in B$.
2. Only one *α-tree* is contained in a belief base, which consists of two kinds of nodes: *i-node* and *τ-node*, where *i-node* constitutes the prefix-tree representation of a s_α sequence and *τ-node* consists of two fields for holding the *min* and *max* values. The prefix tree representation of sequences is detailed in the approach PSP proposed by Masseglia et al (1998).
3. Each *i-node* is connected by a *set-edge* or a *seq-edge* for representing sequences.
4. A *β-tree* is a prefix tree representation of sequences, and is connected with a *τ-node* in the *α-tree* by the *τ-link*, a special edge inter-sub-trees.
5. A *γ-tree* consists of *i-nodes* for storing sequences and *u-nodes* identifying unique unexpectedness IDs, and is connected with a leaf of a *β-tree* by the *η-link*, another special edge inter-sub-trees.
6. Each *η-link* has a copy connected to its parent *τ-node* for the needs of skipping s_β.

For instance, all beliefs concentrated in Example 6 and 7 are shown in Figure 1 as a belief tree.

Based on the above definition and the semantic consistence of a belief, we propose the following belief tree construction algorithm.

Algorithm 1 (*Belief tree construction*).
Input: A belief tree *T* and a belief $b = [s_\alpha; s_\beta; s_\gamma; min..max]$.
Output: The belief tree *T* with the belief *b* appended.

1. If the belief tree *T* is empty, then initialize the global belief base information (e.g., number of nodes, number of beliefs, etc.) and create the root node for the belief tree.

2. Verify the input belief *b*. If $s_\gamma \sqsubseteq s_\beta$, *min* > *max*, or *min* < 0, then reject *b* and exit the construction procedure.
3. Append s_α as prefix tree to the root node of the belief tree, where each item is an *i-node*. Any two items inter-itemsets are connected by a *seq-edge* and any two items within an itemset are connected by a *set-edge*. Append a τ*-node* with *min* and *max* to the last *i-node* of s_α.
4. Transform s_β to prefix tree representation and connect it to the newly created τ*-node* by τ*-link*.
5. Transform s_γ to prefix tree representation and connect it to the newly created leaf of the β*-tree* by η*-link* and copy this link to the parent τ*-node*, and then label the belief *b* by a unique identification.
6. Update the global belief base information and exit the construction procedure.

Given a belief tree *T* constructed from a belief base *B*, a sequence *s* can be verified in at most |*B*| traverses of the belief tree *T* with respect to each belief *b* in the belief base *B*.

UNEXPECTED SEQUENCES AND FEATURES

Given a belief *b* and a sequence *s*, if *s* satisfies the belief *b*, then *s* is an *expected sequence* with respect to the belief *b*, denoted as $s \vDash b$, and $s \nvDash b$ denotes that *s* does not verify the belief *b*; if *s* contradicts the belief *b*, then *s* is an *unexpected sequence*, denoted as $s \nVdash b$. We denote the *unexpectedness* that "contradicting the belief *b*" as $\{\nVdash b\}$. According to the occurrence constraint and the semantic constraint, we propose three forms of unexpected sequences: α*-unexpected*, β*-unexpected* and γ*-unexpected*.

Definition 2. Given a belief $b = [s_\alpha; s_\beta; s_\gamma; 0..*]$ and a sequence *s*, if $s_\alpha \sqsubseteq s$ and there does not exist s_β, s_γ such that $s_\alpha s_\beta \sqsubseteq s$ or $s_\alpha s_\gamma \sqsubseteq s$, then the sequence *s* is an α*-unexpected sequence* stated by the belief *b*, denoted as $s \nVdash_\alpha b$. The α*-unexpectedness* stated by the belief *b* is denoted as $\{\nVdash_\alpha b\}$. ■

A belief with the occurrence constraint $\tau = [0..*]$ states that s_β should occur after the occurrence of s_α in a sequence *s*. Hence, the sequence *s* contradicts the constraint $\tau = [0..*]$ if and only if $s_\alpha \sqsubseteq s$ and $s_\alpha s_\beta \not\sqsubseteq s$. Notice that for not confusing the unexpected sequences caused by the occurrence constraint or the semantic constraint, s_γ should not occur after the occurrence of s_α in an α-unexpected sequence.

Example 8. Let us consider the beliefs listed in Example 6, where the two beliefs

$b_1 = [\langle(Sci\text{-}Fi\text{-}Novel)(Action\text{-}Film\ Sci\text{-}Fi\text{-}Film)\rangle; \langle(Rock\text{-}Music)\rangle; \phi; 0..*]$

and

$b_2 = [\langle(Sci\text{-}Fi\text{-}Novel)(Action\text{-}Film\ Sci\text{-}Fi\text{-}Film)\rangle; \langle(Rock\text{-}Music)\rangle; \langle(Classical\text{-}Music)\rangle; 0..*]$

determine α-unexpected sequences. The belief b_1 depicts that a purchase of rock music CD is expected after the purchases of a Sci-Fi novel then action and Sci-Fi films later. The belief b_2 further requires that the purchase of a classical music CD should not occur. Therefore, given the sequence

$s = \langle$(*Sci-Fi-Novel*)(*Printer*)(*Action-Film Sci-Fi-Film*)(*Classical-Music*)(*PS3-Station*)$\rangle$,

we have $s \nvDash_\alpha b_1$ but $s \models b_2$, that is, s is not an α-unexpected sequence with respect to the belief b_2. ■

Now let us consider the recognition of α-unexpected sequences with respect to a belief base, i.e., a set of beliefs. For instance, given a belief base consists of 3 beliefs

$$b_1 = [\langle(a)\rangle; \langle(b)\rangle; \phi; 0..*],$$
$$b_2 = [\langle(a)\rangle; \langle(c)\rangle; \phi; 0..*],$$
$$b_3 = [\langle(a)\rangle; \langle(d)\rangle; \phi; 0..*],$$

and a set of sequences

$$s_1 = \langle(a)(b)\rangle,$$
$$s_2 = \langle(a)(c)\rangle,$$
$$s_3 = \langle(a)(d)\rangle,$$
$$s_4 = \langle(a)(e)\rangle,$$

we have the following relations:

$$s_1 \nvDash_\alpha b_2,\ s_1 \nvDash_\alpha b_3;$$
$$s_2 \nvDash_\alpha b_1,\ s_2 \nvDash_\alpha b_3;$$
$$s_3 \nvDash_\alpha b_1,\ s_3 \nvDash_\alpha b_2;$$

and

$$s_4 \nvDash_\alpha b_1,\ s_4 \nvDash_\alpha b_2,\ s_4 \nvDash_\alpha b_3.$$

Clearly, the beliefs b_1, b_2, and b_3 depicts that $\langle(b)\rangle$, $\langle(c)\rangle$, or $\langle(d)\rangle$ should occur after the occurrence of $\langle(a)\rangle$, thus, in this meaning, only the sequence s_4 is unexpected. However, according to Definition 2, all of the 4 sequences are α-unexpected. In order to avoid this redundancy problem, we further define the notion of α-unexpected sequence within the context of belief base as below.

Definition 3. Given a belief $b = [s_\alpha; s_\beta; s_\gamma; 0..*]$ and a sequence s, let B be the belief base such that $b \in B$. Let B_α be a subset of B such that for each $b' \in B$, where $b' = [s'_\alpha; s'_\beta; s'_\gamma; 0..*]$, we have $s'_\alpha = s_\alpha$ implies $b' \in B_\alpha$. If $s_\alpha \sqsubseteq s$ and there does not exist $b' \in B$, where $b' = [s'_\alpha; s'_\beta; s'_\gamma; 0..*]$, such that $s_\alpha s'_\beta \sqsubseteq s$ or $s_\alpha s'_\gamma \sqsubseteq s$, then the sequence s is an *α-unexpected sequence* stated by the belief b of the belief base B, denoted as $s \nvDash_\alpha b_{(B)}$. Respectively, the *α-unexpectedness* stated by the belief b of the belief base B is denoted as $\{\nvDash_\alpha b_{(B)}\}$. ■

In fact, an α-unexpected sequence stated by a belief b of a belief base B is an unexpected sequence determined by all sub-trees of a same *τ-node* where *min* = 0 and *max* = $*$. In the rest of

the paper, without special notice, the notation $s \nvDash_\alpha b$ and $\{\nvDash_\alpha b\}$ denote the α-unexpected sequence and α-unexpectedness stated by a belief b within the context of a given belief base B.

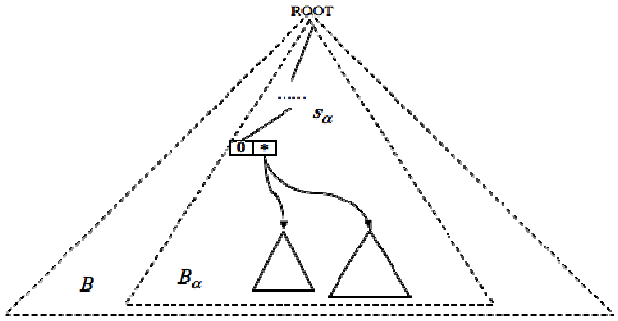

Figure 2. Determining α-unexpected sequences in a belief base.

Definition 4. Given a belief $b = [s_\alpha; s_\beta; s_\gamma; min..max]$ ($min \neq 0$ or $max \neq *$) and a sequence s, if $s_\alpha s_\beta \sqsubseteq s$ and there does not exist a sequence s' such that $|s'| \vDash [min..max]$ and $s_\alpha s' \cdot s_\beta \sqsubseteq_c s$, then the sequence s is a *β-unexpected sequence* stated by the belief b, denoted as $s \nvDash_\beta b$. The *β-unexpectedness* stated by the belief b is denoted as $\{\nvDash_\beta b\}$. ■

A β-unexpected sequence reflects that the occurrence constraint $\tau = [min..max]$ ($\tau \neq [0..*]$) on the sequential rule $s_\alpha \rightarrow s_\beta$ is broken because the occurrence of s_β in the sequence s contradicts the constraint τ.

Example 9. Let us consider the below belief proposed in Example 6:

[⟨(*Sci-Fi-Novel*)(*Action-Film Sci-Fi-Film*)⟩; ⟨(*Rock-Music*)⟩; ⟨(*Classical-Music*)⟩; 3..5].

The purchase of a rock music CD is expected within the next 3 to 5 purchases after the purchases of a Sci-Fi novel then action and Sci-Fi films later. In this case, the customers who purchase a rock music CD just in the next purchase or after many purchases of other products are unexpected to this belief and might be valuable to make new promotion strategies on related products. Notice that ⟨(*Classical-Music*)⟩ in this belief is not considered within the context of β-unexpected sequences. ■

Definition 5. Given a belief $b = [s_\alpha; s_\beta; s_\gamma; min..max]$ and a sequence s, if $s_\alpha s_\gamma \sqsubseteq s$ and there exists a sequence s' such that $|s'| \vDash [min..max]$ and $s_\alpha s' \cdot s_\gamma \sqsubseteq_c s$, then the sequence s is a *γ-unexpected sequence* stated by the belief b, denoted as $s \nvDash_\gamma b$. The *γ-unexpectedness* stated by the belief b is denoted as $\{\nvDash_\gamma b\}$. ■

A γ-unexpected sequence is concentrated on the semantics: the occurrence of s_β is replaced by its semantic opposition s_γ within the occurrence constraint $\tau = [min..max]$.

Example 10. Let us consider again the belief studied in Example 9:

[⟨(*Sci-Fi-Novel*)(*Action-Film Sci-Fi-Film*)⟩; ⟨(*Rock-Music*)⟩; ⟨(*Classical-Music*)⟩; 3..5].

The rock music can be considered as being opposite to the classical music. Therefore, the purchase of a rock music CD cannot be replaced by the purchase of a classical music CD. In this example, since the purchase of a rock music CD is expected within the next 3 to 5 purchases after the purchases of a Sci-Fi novel then action and Sci-Fi films later, the purchase of a classical music CD is not expected within the interval of the next 3 to 5 purchases. Of course, in this case, the purchase of a classical music CD is allowed within the next 3 purchases or after the next 5 purchases according the purpose of this belief. ■

A feature is the unexpected part of an unexpected sequence, which informs the internal structure of the unexpectedness.

Definition 6. Given a belief $b = [s_\alpha; s_\beta; s_\gamma; min..max]$ and an unexpected sequence s such that $s \nvDash b$, the *feature* of the unexpected sequence s is the maximum consecutive subsequence u of the sequence s such that: (1) if $s \nvDash_\alpha b$, we have $s_a \cdot u = s$, where s_a is a sequence such that $|s_a| \geq 0$; (2) if $s \nvDash_\beta b$, we have $s_a \cdot u \cdot s_c = s$, where s_a and s_c are two sequences such that $|s_a| \geq 0$, $|s_c| \geq 0$, $s_\alpha \not\sqsubseteq s_a$, $s_\alpha \sqsubseteq^{[} u$, and $s_\beta \sqsubseteq^{]} u$ (i.e., $s_\alpha s_\beta \sqsubseteq^{[\cdot]} u$); (3) if $s \nvDash_\gamma b$, we have $s_a \cdot u \cdot s_c = s$, where s_a and s_c are two sequences such that $|s_a| \geq 0$, $|s_c| \geq 0$, $s_\alpha \not\sqsubseteq s_a$, $s_\alpha \sqsubseteq^{[} u$, and $s_\gamma \sqsubseteq^{]} u$ (i.e., $s_\alpha s_\gamma \sqsubseteq^{[\cdot]} u$). The feature of an unexpected sequence s with respect to a belief b is denoted as $u \vDash (s \nvDash b)$. ■

Figure 3. Features of unexpected sequences.

Example 11. Let us consider a belief $b = [\langle(e)(f)\rangle; \langle(d)\rangle; \langle(c)\rangle; 0..3]$, Figure 3 shows the features of a set of β-unexpected or γ-unexpected sequences. ■

Based on the definitions of unexpected sequences and the structure of belief tree, we have the following algorithm for mining unexpected sequences.

Algorithm 2 (*USE: Mining unexpected sequences*).
Input: A sequence database D, a belief base B stored as a belief tree T.
Output: All unexpected sequences with respect to each belief $b \in B$.

1. Read a sequence s from the sequence database D. If all sequences in D have been processed, exit the procedure.

2. Use depth-first method for traversing the α-*tree* of the belief tree T, till to reach a τ-*node*. If no τ-*node* can be reached, reject current sequence and back to step 1 for restarting with next sequence.
3. If current τ-*node* consists of [0..∗], then use depth-first method for traversing all β-*trees* by following each τ-*link*. If no leaf-node of any β-*tree* and of any γ-*tree* can be reached, mark current sequence s as α-unexpected. If any leaf-node of any γ-*tree* can be reached, mark current sequence s as γ-unexpected. If the sequence s is unexpected, output the sequence s, the feature u, the antecedent and consequent sequences with belief identification. Continue step 2.
4. If current τ-*node* does not consist of [0..∗], then use depth-first method for traversing all nodes of all β-*trees* by following each τ-*link*. If any leaf-node of any β-*tree* can be reached and the path can be verified with respect to the complement of [*min*..*max*] contained in current τ-*node*, mark current sequence s as β-unexpected. Use depth-first method for traversing all γ-*trees* by following each η-*link* from current leaf-node of current β-*tree*. If any leaf-node of any γ-*tree* can be reached, mark current sequence s as γ-unexpected. If the sequence s is unexpected, output the sequence s, the feature u, the antecedent and consequent sequences with belief identification. Continue step 2.

Unexpected Sequential Patterns and Implication Rules

According to Definition 6, given an unexpected sequence s stated by a belief b, the feature u is the part of the sequence s that causes the unexpectedness $\{\not\Vdash b\}$. With features, we can study the internal structure of the unexpectedness via the notion of unexpected sequential patterns.

Definition 7. Given a belief b and a sequence database D, let $D_{\{\not\Vdash b\}}$ be a subset of the sequence database D consisting of all sequences $s \in D$ such that $s \not\Vdash b$, and let $U_{\{\not\Vdash b\}}$ be the feature set of $u \vDash (s \not\Vdash b)$ of each unexpected sequence $s \in D_{\{\not\Vdash b\}}$. Given a user specified minimum support threshold σ_{min}, an *unexpected sequential pattern* is a maximal sequence p in the feature set $U_{\{\not\Vdash b\}}$ such that $\sigma(p, U_{\{\not\Vdash b\}}) \geq \sigma_{min}$. ■

Notice that in the feature set $U_{\{\not\Vdash b\}}$, the support value of the unexpected part (i.e., s_α in α-unexpected sequences, $s_\alpha s_\beta$ in β-unexpected sequences, and $s_\alpha s_\gamma$ in γ-unexpected sequences) is 100% with perforce. For example, for the β-unexpectedness, the support of the sequence $s_\alpha s_\beta$ in the feature set $U_{\{\not\Vdash b\}}$ is 100%, since for each feature $u \in U_{\{\not\Vdash b\}}$, we have the same structure $u = s_\alpha s' s_\beta$. Therefore, for extracting unexpected sequential patterns, we do not consider the subsequences s_α, s_β and (or) s_γ in the feature set $U_{\{\not\Vdash b\}}$. Since any existing sequential pattern mining algorithms can extract the unexpected sequential patterns, we do not repeat such a process in this chapter.

Example 12. For the sequence database shown in Figure 3, in the feature set of β-unexpected sequences, we find that the sequence $\langle(a)(a)(a)\rangle$ is an unexpected sequential pattern that its

presence gives the β-unexpectedness; in the feature set of γ-unexpected sequences, we find that the presence of the sequential pattern $\langle(b)(b)(b)\rangle$ indicates β-unexpectedness. ■

Given an unexpected sequence s and its feature u, the sequence s can be represented as $s = s_a \cdot u \cdot s_c$, where $|s_a|, |s_c| \geq 0$ (we have $|s_c| \equiv 0$ for an α-unexpected sequence). The sequences s_a and s_c are called the *antecedent sequence* and the *consequent sequence* of an unexpected sequence.

Definition 8. Given a belief b and a sequence database D, let $D^A_{\{\not\models b\}}$ be the subset of the database D that consists of the antecedent sequences s_a of each sequence $s \in D$ such that $s \not\models b$. An *antecedent rule* of the unexpectedness $\{\not\models b\}$ is a rule $a \rightarrow \{\not\models b\}$ where a is a frequent sequence in the sequence set $D^A_{\{\not\models b\}}$. ■

Antecedent rules reflect the causes of the unexpectedness contradicting a given belief b. With respect to a belief b, the support of an antecedent rule in a sequence database D, denoted as $\sigma(a \rightarrow \{\not\models b\}, D)$, is the fraction of the total number of the sequences in the sequence set $D^A_{\{\not\models b\}}$ that support the sequence a on the sequence database D, that is,

$$\sigma(a \rightarrow \{\not\models b\}, D) = \frac{\left|\left\{s \middle| \left(a \sqsubseteq s\right) \wedge \left(s \in D^A_{\{\not\models b\}}\right)\right\}\right|}{\left|D\right|}.$$

The confidence of an antecedent rule in the sequence database D, denoted as $\delta(a \rightarrow \{\not\models b\}, D)$, is the fraction of the total number of the sequences in the sequence database D that support the sequence a, that is,

$$\delta(a \rightarrow \{\not\models b\}, D) = \frac{\left|\left\{s \middle| \left(a \sqsubseteq s\right) \wedge \left(s \in D^A_{\{\not\models b\}}\right)\right\}\right|}{\left|\left\{s \middle| (a \sqsubseteq s) \wedge (s \in D)\right\}\right|}.$$

Example 13. Considering the sequence database shown in Figure 3, according to the belief $b = [\langle(e)(f)\rangle; \langle(d)\rangle; \langle(c)\rangle; 0..3]$, given a minimum support 50% and a minimum confidence 50%, we have the rule $\langle(ab)\rangle \rightarrow \{\not\models_\beta b\}$, whose support is 60% and confidence is 100%. ■

Definition 9. Given a belief b and a sequence database D, let $D^C_{\{\not\models b\}}$ be the subset of the database D that consists of the consequent sequences s_c of each sequence $s \in D$ such that $s \not\models b$. An *antecedent rule* of the unexpectedness $\{\not\models b\}$ is a rule $\{\not\models b\} \rightarrow c$ where c is a frequent sequence in the sequence set $D^C_{\{\not\models b\}}$. ■

Consequent rules reflect the causes of the unexpectedness contradicting a given belief b. With respect to a belief b, the support of a consequent rule in a sequence database D, denoted as $\sigma(\{\not\models b\} \rightarrow c, D)$, is the total number of sequences in the sequence set $D^C_{\{\not\models b\}}$ that support the sequence c on the sequence database D, that is,

$$\sigma(\{\not\Vdash b\} \to c, D) = \frac{\left|\left\{s \middle| \left(c \sqsubseteq s\right) \wedge \left(c \in D^{C}_{\{\not\Vdash b\}}\right)\right\}\right|}{\left|D\right|}.$$

The confidence of a consequent rule in the sequence database D, denoted as $\delta(\{\not\Vdash b\} \to c, D)$, is the fraction of the total number of the sequences in the sequence set $D^{C}_{\{\not\Vdash b\}}$ that support the sequence c, that is,

$$\delta(\{\not\Vdash b\} \to c, D) = \frac{\left|\left\{s \middle| \left(c \sqsubseteq s\right) \wedge \left(c \in D^{C}_{\{\not\Vdash b\}}\right)\right\}\right|}{\left|D^{C}_{\{\not\Vdash b\}}\right|}.$$

Example 14. Considering again the sequence database shown in Figure 3, according to the belief b = [⟨(*e*)(*f*)⟩; ⟨(*d*)⟩; ⟨(*c*)⟩; 0..3], given a minimum support 50% and a minimum confidence 50%, we have the rule $\{\not\Vdash_{\beta} b\} \to \langle(bc)\rangle$, whose support is 60% and confidence is 100%. ■

For globally illustrating the purpose of mining unexpected sequential rules including the antecedent rules and the consequent rules, let us study the following example.

Example 15. Considering a WebMail system, assume a log file containing 10,000 user sessions of (*Time*, *IP*, *Request*) where *Time* identifies the time range of the session, *IP* identifies the range of remote IP addresses, and *Request* identifies the resources requested such that *Request* ∈ {*Begin-Session*, *End-Session*, *Help*, *Login*, *Logout*, *Mailbox*, *Reset-Password*, …}, where *Help*, *Login*, *Logout*, *Reset-Password*, etc. note Web pages. In such a log file, each user session is a sequence. A valid user login process (the access of *Login*) should redirect the user session to the mailbox page (the access of *Mailbox*), so that a belief on such a behavior can be b = [⟨(*Login*)⟩; (*Mailbox*)⟩; ⟨(*Logout*)⟩; 0..0]. Suppose that we found 100 β-unexpected sequences.

Assume that we found that 100 sequences in the whole log file with 80 β-unexpected sequences support the antecedent sequence ⟨(*T1*, *IP1*, *Begin-Session*)⟩; 9,000 sequences in the whole log file with 20 β-unexpected sequences support the antecedent sequence ⟨(*IP2*, *Begin-Session*)⟩; 90 β-sequences support the consequent sequence ⟨(*T1*, *IP1*, *End-Session*)⟩; 15 β-unexpected sequences support the consequent sequence ⟨(*IP2*, *Reset-Password*)(*IP2*, *End-Session*)⟩; 10 β-unexpected sequences support the frequent consequent sequence ⟨(*IP2*, *Help*)(*IP2*, *End-Session*)⟩.

According to the above assumes, we have: the antecedent rule ⟨(*T1*, *IP1*, *Begin-Session*)⟩ → $\{\not\Vdash_{\beta} b\}$ with support 80/10,000 = 0.8% and confidence 80/100 = 80%; the antecedent rule ⟨(*IP2*, *Begin-Session*)⟩ → $\{\not\Vdash_{\beta} b\}$ with support 10/10,000 = 0.1% and confidence 10/9,000 ≅ 0.1%; the consequent rule $\{\not\Vdash_{\beta} b\}$ → ⟨(*T1*, *IP1*, *End-Session*)⟩ with support 90/10,000 = 0.09% and confidence 90/100 = 90%; the consequent rule $\{\not\Vdash_{\beta} b\}$ → ⟨(*IP2*, *Reset-Password*)(*IP2*, *End-Session*)⟩ with support 15/10,000 = 0.15% and confidence 15/100 = 15%; the consequent rule $\{\not\Vdash_{\beta} b\}$ → ⟨(*IP2*, *Help*)(*IP2*, *End-Session*)⟩ with support 10/10,000 = 0.1% and confidence 10/100 = 10%.

Obviously, we can interpret the antecedent rule ⟨(*T1*, *IP1*, *Begin-Session*)⟩ → $\{\not\Vdash_{\beta} b\}$ and the consequent rule $\{\not\Vdash_{\beta} b\}$ → ⟨(*T1*, *IP1*, *End-Session*)⟩ as that the connections from IP range 1 at

time range 1 can be considered as critical event since the confidences of these two rules are strong, however, the antecedent rule $\langle(IP2, Begin\text{-}Session)\rangle \rightarrow \{\not\models_\beta b\}$ can be safely ignored not only because the very low confidence, but also the consequent rules $\{\not\models_\beta b\} \rightarrow \langle(IP2, Reset\text{-}Password)(IP2, End\text{-}Session)\rangle$ and $\{\not\models_\beta b\} \rightarrow \langle(IP2, Help)(IP2, End\text{-}Session)\rangle$ show that the connections from IP2 do not contain strong behaviors that can be interpreted as critical events. ■

Based on the above propositions, Algorithm 3 shows the procedure of mining unexpected antecedent and consequent rules in a sequence database, with user defined minimum support and confidence threshold values.

Algorithm 3 (*USR: Mining unexpected sequential rules*).

Input: A sequence database D, a belief base B stored as a belief tree T, minimum support σ_{min}, minimum confidence δ_{min}.

Output: All antecedent and consequent rules stated by each belief $b \in B$, with respect to the minimum support σ_{min} and minimum confidence δ_{min}.

1. Call the procedure *USE* for extract the antecedent sequence set $D^A_{\{\not\models b\}}$ and the consequent set $D^C_{\{\not\models b\}}$ stated by each belief $b \in B$.
2. For each antecedent sequence s_a in the antecedent sequence set $D^A_{\{\not\models b\}}$, find all sequential patterns $a \in D^A_{\{\not\models b\}}$ such that the support $\sigma(a, D) \geq \sigma_{min}$. If the fraction of $\sigma(a, D^A_{\{\not\models b\}})/\sigma(a, D) \geq \delta_{min}$, output the rule $a \rightarrow \{\not\models b\}$.
3. For each consequent sequence s_c in the consequent sequence set $D^C_{\{\not\models b\}}$, find all sequential patterns $c \in D^C_{\{\not\models b\}}$ such that the support $\sigma(c, D) \geq \sigma_{min}$. If the support $\sigma(c, D^C_{\{\not\models b\}}) \geq \delta_{min}$, output the rule $\{\not\models b\} \rightarrow c$.

Notice that we separate the process of mining unexpected sequential implication rules into two standalone sub-routines: we first compute the support value of the premise a or the consequence c, then we compute the confidence of the rules, in order to obtain the best performance and flexibility.

EXPERIMENTAL EVALUATION

To evaluate the effectiveness and scalability of our approach, we have performed two groups of experiments. The first group of experiments is performed on large log files of two real Web servers, with the belief base defined by domain experts. The second group of experiments is performed on various dense synthetic data files generated by the IBM Quest Synthetic Data Generator[1], where we use a set of random generated beliefs as the belief bases. All experiments have been performed on a Sun Fire V880 system with 8 1.2GHz UltraSPARC III processors and 32GB main memory running Solaris 10 operating system.

[1] http://www.almaden.ibm.com/cs/quest/

Experiments on Web Access Logs

We performed a group of experiments on two large log files containing the access records of two Web servers during a period of 3 months. The first log file, labeled as LOGBBS, corresponds to a PHP based discussion forum Web site of an online game provider; the second log file, labeled as LOGWWW, corresponds to a Web site that hosts personal home pages of researchers and teaching staffs. We split each log file into three 1-month period files, i.e., LOGBBS-{1,2,3} and LOGWWW-{1,2,3}. Table 1 details the number of sequences, distinct items, and the average length of the sequences contained in the Web access logs.

Access Log	Sessions	Distinct Items	Average Length
LOGBBS-1	27,294	38,678	12.8934
LOGBBS-2	47,868	42,052	20.3905
LOGBBS-3	28,146	33,890	8.5762
LOGWWW-1	6,534	8,436	6.3276
LOGWWW-2	11,304	49,242	7.3905
LOGWWW-3	28,400	50,312	9.5762

Table 1. Web access logs in experiments.

In order to compare our approach with the sequential pattern mining, we first apply the sequential pattern mining algorithm to find the frequent behaviors from LOGBBS-{1,2,3} and LOGWWW-{1,2,3} with different minimum support thresholds, shown in Figure 4 (a) and (b); Figure 4 (c) and (d) show the number of unexpected sequential implication rules discovered by *USER*. Post analysis of the experimental results shows the effectiveness of our approach.

The result set of our approach is much less than the extremely large sequence set generated by sequential pattern mining, where the many discovered frequent sequences are similar in the data sets LOGBBS-{1,2,3}. One important reason is that the accesses of the Web server of LOGBBS-{1,2,3} are very regular and the most frequent behaviors are similar. Moreover, with the minimum confidence 20%, totally 15 antecedent rules and 2 consequent rules are finally recognized as representing new navigation behaviors of users, however, such behaviors have low support values (< 1%) and cannot be discovered by frequency based approaches in our experiments, since according to Figure 4 (a), with the minimum support 2%, more than 1000 frequent sequences are extracted.

The experiments on the data sets LOGWWW-{1,2,3} show that the comparison of the result size is similar to the experiments data sets LOGBBS-{1,2,3}. An important note is that the antecedent rules discovered by *USER* with the minimum confidence 10%, totally 12 rules show the relevant information of Web security problems. However, in the data sets LOGWWW-{1,2,3}, only 1 consequent rule shows a weak connection of Web security.

Experiments on Synthetic Data

The scalability of the *USER* approach has been tested first with a fixed belief number of 20 by increasing the size of sequence database from 10,000 sequences to 500,000 sequences, and then

with a fixed sequence database size of 100,000 sequences by increasing the number of beliefs from 5 to 25.

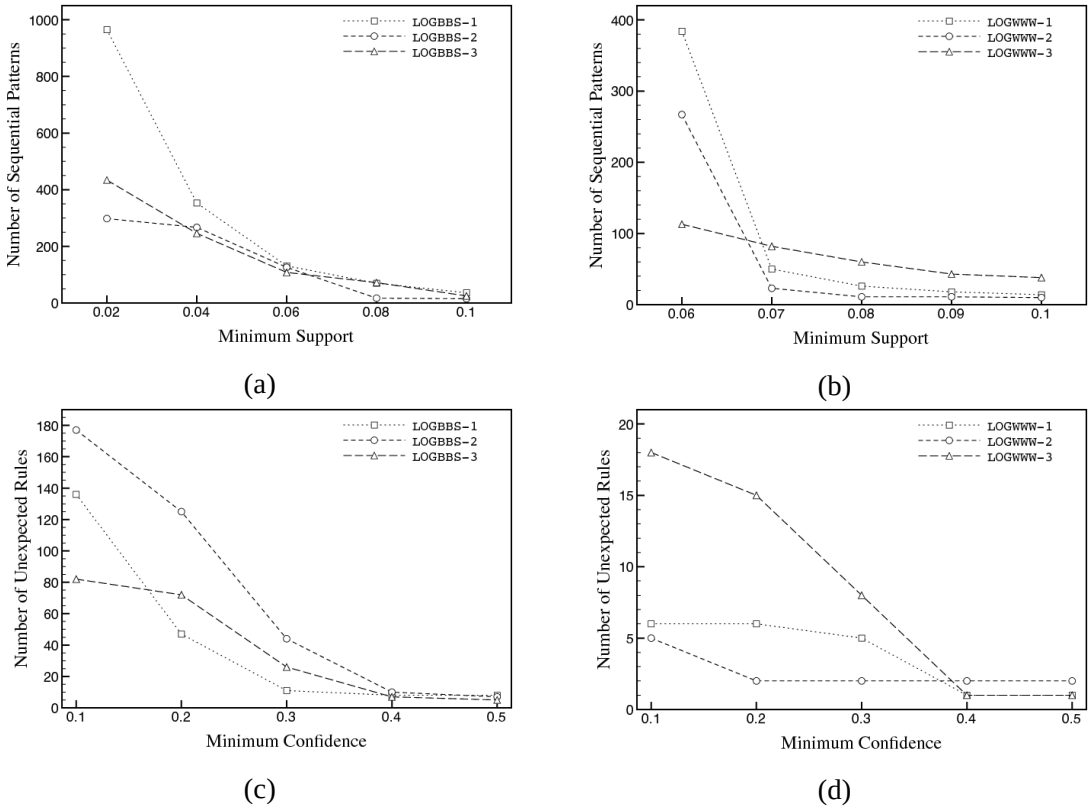

Figure 4. (a) Sequential patterns in data sets `LOGBBS-{1,2,3}`. (b) Unexpected sequential implication rules in data sets `LOGBBS-{1,2,3}`. (c) Sequential patterns in data sets `LOGWWW-{1,2,3}`. (b) Unexpected sequential implication rules in data sets `LOGWWW-{1,2,3}`.

Figure 5(a) shows that, when the belief number is fixed, the number of all unexpected sequences increases linearly with the increasing of the size of sequence database. Because the data sets generated by the IBM Quest Synthetic Data Generator contain repeated blocks, the unexpected sequences with respect to the same 20 beliefs are repeated. Therefore, Figure 5(b) shows that, when the belief number is fixed to 20, the run time of the extraction of all unexpected sequences increases linearly with the increasing of the size of sequence database.

Figure 5(c) shows that, when the size of sequence database is fixed, the number of all unexpected sequences extracted increases, but not linearly, when the number of beliefs increases. This is a previewed result since the number of unexpected sequences depends on the structure of beliefs. In this test the last 10 beliefs address much less unexpected sequences than others. Figure 5(d) shows the increment of run time of the extraction of all unexpected sequences illustrated in Figure 5(c), and from which we can find that the increasing rate of extracting time depends on the number of unexpected sequences. In our implementation of the USER approach, to predict and process a non-matched sequence is much faster than to predict and process a matched sequence.

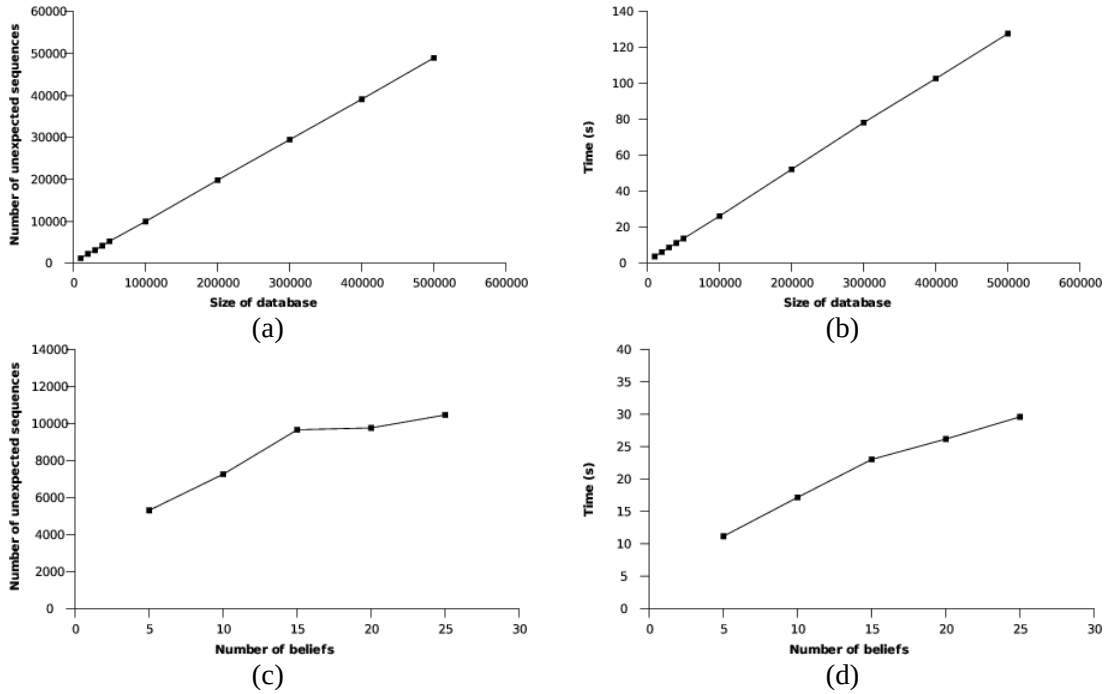

Figure 5. (a) Number of all unexpected sequences stated by 20 beliefs. (b) Run time for extracting all unexpected sequences stated by 20 beliefs. (c) Number of all unexpected sequences in 100,000 sequences. (d) Run time for extracting all unexpected sequences from 100,000 sequences.

CONCLUSIONS

In this chapter, we introduce a belief-driven approach *USER* for mining unexpected sequential patterns and implication rules in sequence databases. We first formalize the belief base and propose 3 forms of unexpected of sequences, and then we propose the notions and discoveries of the unexpected sequential patterns and implication rules, including antecedent rules and consequent rules for measuring the unexpected behaviors in sequence data.

The approach *USER* is evaluated with different types of Web access logs and synthetic data. Our experimental results show that: (1) our approach permits to extract unexpected sequential patterns and implication rules with low support value; (2) our approach is capable to find unexpected sequences that are included in expected sequences; (3) the unexpected sequences depend on the belief base and the characteristics of the sequence database.

Our approach can be extended with an application of soft beliefs. For example, in a data set, we know that 90% of customers purchase a Sci-Fi novel and then action and Sci-Fi films later, so it is possible to create a soft belief like “the purchase of a Sci-Fi novel implies the purchase of action and Sci-Fi films”, and its degree can be defined by a soft measure function μ (0.9). If in another data set, there are only 10% of customers who confirm this belief, then the change of degree can be computed by the a soft measure function ψ (0.9, 0.1). We are also interested in mining unexpected sequences and sequential rules with the notion of hierarchies and soft hierarchies.

REFERENCES

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. In *ICDE* (pp. 3-14).

Angiulli, F., & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. In *PKDD* (pp. 15-26).

Angiulli, F., & Pizzuti, C. (2005). Outlier mining in large high-dimensional data sets. *IEEE Trans. Knowl. Data Eng.*, 17(2), 203-215.

Ayres, J., Flannick, J., Gehrke, J., & Yiu, T. (2002). Sequential PAttern Mining using a bitmap representation. In *KDD* (pp. 429-435).

Dong, G., & Pei, J. (2007). *Sequence Data Mining (Advances in Database Systems)*: Springer.

Garofalakis, M. N., Rastogi, R., & Shim, K. (1999). SPIRIT: Sequential pattern mining with regular expression constraints. In *VLDB* (pp. 223-234).

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques* (2nd ed.): Morgan Kaufmann Publishers.

Jin, W., Tung, A. K. H., & Han, J. (2001). Mining top-n local outliers in large databases. In *KDD* (pp. 293-298).

Knorr, E. M., & Ng, R. T. (1998). Algorithms for mining distance-based outliers in large datasets. In *VLDB* (pp. 392-403).

Li, D. H., Laurent, A., & Teisseire, M. (2007). On transversal hypergraph enumeration in mining sequential patterns. In *IDEAS* (pp. 303-307).

Lo, D., Khoo, S.-C., & Liu, C. (2007). Efficient mining of iterative patterns for software specification discovery. In *KDD* (pp. 460-469).

Mannila, H., Toivonen, H., & Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. *Data Min. Knowl. Discov.*, 1(3), 259-289.

Masseglia, F., Cathala, F., & Poncelet, P. (1998). The PSP approach for mining sequential patterns. In *PKDD* (pp. 176-184).

McGarry, K. (2005). A survey of interestingness measures for knowledge discovery. *Knowl. Eng. Rev.*, 20(1), 39-61.

Padmanabhan, B., & Tuzhilin, A. (1998). A belief-driven method for discovering unexpected patterns. In *KDD* (pp. 94-100).

Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient Algorithms for Mining Outliers from Large Data Sets. In *SIGMOD* (pp. 427-438).

Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., et al. (2004). Mining sequential patterns by pattern-growth: the PrefixSpan approach. *IEEE Trans. Knowl. Data Eng.*, 16(11), 1424-1440.

Spiliopoulou, M. (1999). Managing interesting rules in sequence mining. In *PKDD* (pp. 554-560).

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: generalizations and performance improvements. In *EDBT* (pp. 3-17).

Sun, P., Chawla, S., & Arunasalam, B. (2006). Mining for Outliers in Sequential Databases. In *SDM* (pp. 94-105).

Wang, J., & Han, J. (2004). BIDE: Efficient mining of frequent closed sequences. In *ICDE* (pp. 79-90).

Yan, X., Han, J., & Afshar, R. (2003). CloSpan: Mining closed sequential patterns in large databases. In *SDM* (pp. 166-177).

Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1-2).