



HAL
open science

Mining Unexpected Sequential Patterns and Rules

Haoyuan Li, Anne Laurent, Pascal Poncelet

► **To cite this version:**

Haoyuan Li, Anne Laurent, Pascal Poncelet. Mining Unexpected Sequential Patterns and Rules. RR-07027, 2007, pp.14. lirmm-00193679

HAL Id: lirmm-00193679

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00193679v1>

Submitted on 4 Dec 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Mining Unexpected Sequential Patterns and Rules

Dong (Haoyuan) Li, Anne Laurent, Pascal Poncelet

November 14, 2007

Abstract

Sequential pattern mining is the one most concentrated and applied in sequence mining research, it gives a frequency based view of the correlations between elements contained in the sequences. However, when we consider domain knowledge within the data mining process, the frequency based criterion becomes less interesting since most of the frequent sequences might have already been confirmed, and the most interesting sequences might not be the sequences corresponding to existing knowledge, but be the sequences contradicting existing knowledge that reflect unexpected behaviors. In this paper we introduce the problem of finding unexpected behaviors within the context of sequence mining. We first give formal descriptions of belief base and unexpected sequences, we then introduce unexpected sequential patterns and unexpectedness rules that depict unexpected behaviors within the sequences. We also propose the USER approach for mining unexpected sequential patterns and rules from a sequence database with respect to a given belief base. Our experimental results show that both of the quantity and the quality of the unexpected sequences extracted by the USER approach are improved in comparison with the frequent sequences extracted by general sequential pattern mining approaches.

1 Introduction

To date, for the requirements of modern applications (marketing development, bioinformatics, Internet navigation analysis, etc.), more and more data have been stored in the form of sequence in databases. In order to find pertinent correlations from those databases of sequences, [1] introduced the problem of mining sequential patterns that finds maximal frequent sequences from a given database of sequences with respect to a user specified minimum support threshold. Though the correlations between sequential data are essential for decision making, the unexpected sequential patterns that contradict the beliefs acquired from domain knowledge are more and more concentrated. With such unexpected sequential patterns, it becomes, for example, possible to respond network emergencies, to determine system crashes, to position new commercial strategies, and even to find the frauds.

For instance, in considering a database of retail transactions, we might find by sequential pattern mining that customers typically purchase an iMac computer, then an iPod player, and then an iPhone cell phone. Such a sequential

pattern is too reasonable to be valuable to push commercial strategies since it corresponds obviously to our existing knowledge of customer behaviors. On the other hand, a transaction sequence such like customers purchase an iMac computer then a Windows Mobile cell phone, can be more important because it contradicts general behaviors known in this domain.

Note that our purpose is not to find rare sequences, but to find the sequences contradicting existing knowledge. Though the extraction of semantical contradiction exists in the association rule mining [4], there does not exist any approach to find unexpected sequences corresponding to domain knowledge based on semantics. In this paper, we propose the notion of the semantics based belief system and its contradictions within the context of sequence mining, with which we therefore define the unexpected sequences. Since the traditional sequential patterns mining approaches do not find the "antecedent-consequent" type rules, we extend the notion of unexpected sequences to unexpectedness rules. For extracting such rules from a sequence database with beliefs, we propose the USER (Unexpected Sequence Extracted Rules) approach.

The rest of this paper is organized as follows. Section 2 introduces the related research work. In Section 3 we present the USER approach for mining unexpected sequential patterns and unexpectedness rules. We give formal descriptions of belief base and unexpectedness of sequences, we then introduce unexpected sequential patterns and unexpectedness rules depicting unexpected behaviors within the sequences. We also show the algorithms USER. Section 4 shows our experimental results with real data and synthetic data. The conclusion and our future research in unexpected sequence mining are listed in Section 5.

2 Related Work

In this section we introduce existing work related to unexpected sequence mining including interestingness measures for data mining, unexpected association rule mining and other approaches to unexpected sequence mining.

[2] summarized the measures of interestingness for data mining. Interestingness measures can be classified as objective measures and subjective measures. Objective measures typically depend on the structure of extracted patterns and the criteria based on the approaches of probability and statistics (e.g. support and confidence); subjective measures are generally user and knowledge oriented, the criteria can be actionability, unexpectedness etc.. The belief driven unexpectedness is first introduced by [3] as a subjective measure where beliefs are categorized to hard beliefs and soft beliefs. A hard belief is a constraint that cannot be changed with new evidences, and any contradiction of a hard belief implies the error in gathering new evidence. A soft belief is a constraint that can be changed with new evidences by updating the degree of belief, and the interestingness of new evidence is measured by the changes of degree of belief.

In our approach two interestingness measures are involved. The first one is a subjective measure that is the belief based criterion *unexpectedness*, that is used for finding unexpected sequences. The second one is an objective measure that is the statistical frequency based criteria *support* and *confidence*, that are used for finding unexpected sequential patterns and unexpectedness rules. Note that in our approach though a belief is rather a "hard belief", we do not consider

the contradiction of a belief as errors in data.

Based on the proposition of [3], in the most recent approach to unexpected association rule mining presented by [4], a belief is represented as a rule with the form $X \rightarrow Y$, and a rule $A \rightarrow B$ is unexpected to the belief $X \rightarrow Y$ if: (a) B and Y logically contradict each other, denoted by $B \text{ AND } Y \models \text{FALSE}$; (b) the rule $A \cup X \rightarrow B$ satisfies given support/confidence threshold values; (c) the rule $A \cup X \rightarrow Y$ does not satisfy given support/confidence threshold values. The mining process is done by the *a priori* based algorithms that find the minimal set of unexpected association rules with respect to a set of user defined beliefs.

[5] proposed a framework based on domain knowledge and beliefs for finding unexpected sequence rules from frequent sequences. The author first introduced the generalized sequence $g_1 * g_2 * \dots * g_n$ so called “g-sequence” where g_1, g_2, \dots, g_n are elements of sequence and $*$ is a wildcard. The author then proposed the sequence rule by splitting a g-sequence into two adjacent parts: a premise part LHS and a conclusion part RHS , denoted as $LHS \leftrightarrow RHS$. A belief over g-sequence is a tuple $\langle LHS, RHS, CL, C \rangle$ where CL is a conjunction of constraints on the statistical frequency of LHS and C is a conjunction of constraints involving elements of LHS and RHS . For example, as introduced by [5], let belief $\langle a * b, c, CL, C \rangle$ be a belief with $CL = (\text{support}(a * b) \geq 0.4 \wedge \text{confidence}(a, b) \geq 0.8)$ and $C = (\text{confidence}(a * b, c) \geq 0.9)$. This belief states that the LHS of the sequence rule $a * b \leftrightarrow c$ should appear in at least 40% of sequences, the confidence of the belief given a should be at least 0.8 while the RHS confidence should be at least 0.9. So that a sequence rule is expected if it confirms to a belief in terms of statistics of content. Finally the unexpected rules are grouped by the semantics of there unexpectedness and can be used for creating new rules.

The approach of [5] is to find the sequences that do not satisfy given statistical frequency constraints of each occurrences, it is different to our approach to unexpected sequence mining.

3 The Approach USER

In this section we present our approach USER. We first introduce the preliminary concepts, including the widely considered formal model of sequence and the occurrence relation that we propose for defining constraints within a sequence. Based on this relation we give formal descriptions of the belief base considered in our approach, and with which we then formalize the unexpectedness of sequences that states unexpected sequences. We further propose unexpected sequential patterns within the framework of mining sequential patterns and propose unexpectedness rules including antecedent rules and consequent rules for depicting the causality of unexpectedness. Finally we present the algorithm USER for finding unexpected sequential patterns and unexpectedness rules.

3.1 Preliminary Concepts

Given a set of distinct attributes, an *item* i is an attribute. An *itemset* \mathcal{I} is an unordered collection of items, denoted as $(i_1 i_2 \dots i_m)$. A *sequence* s is an ordered list of itemsets, denoted as $\langle \mathcal{I}_1 \mathcal{I}_2 \dots \mathcal{I}_k \rangle$. A *sequence database* \mathcal{D} is a large set of sequences.

Given two sequences $s = \langle \mathcal{I}_1 \mathcal{I}_2 \dots \mathcal{I}_m \rangle$ and $s' = \langle \mathcal{I}'_1 \mathcal{I}'_2 \dots \mathcal{I}'_n \rangle$, if there exist integers $1 \leq i_1 < i_2 < \dots < i_m \leq n$ such that $\mathcal{I}_1 \subseteq \mathcal{I}'_{i_1}, \mathcal{I}_2 \subseteq \mathcal{I}'_{i_2}, \dots, \mathcal{I}_m \subseteq \mathcal{I}'_{i_m}$, then the sequence s is a *subsequence* of the sequence s' , denoted as $s \sqsubseteq s'$, and we say that s is *contained in* s' . In a set of sequences, if sequence s is not a subsequence of any other sequences, then we say that sequence s is *maximal*; otherwise, if sequence s is contained in sequence s' , we say that sequence s' *supports* sequence s . Given a sequence $s = \langle \mathcal{I}_1 \mathcal{I}_2 \dots \mathcal{I}_k \rangle$, a *segment* $g \sqsubseteq s$ is a sequence that contains a number of contiguous itemsets $\langle \mathcal{I}_i \mathcal{I}_{i+1} \dots \mathcal{I}_{i+n} \rangle$ where $i \leq 1$ and $i+n \leq k$. The *support* of a sequence is defined as the fraction of total sequences in \mathcal{D} that support this sequence.

In this paper we consider the *length* of sequence as the number of itemsets contained in the sequence, denoted as $|s|$. We also consider the empty sequence and the concatenation of sequences. An *empty sequence* is denoted as \emptyset , we have $s = \emptyset \iff |s| = 0$. The *concatenation* of sequences is denoted as the form $s_1 \cdot s_2$ that means a sequence with s_1 appended by s_2 , and we have $|s_1 \cdot s_2| = |s_1| + |s_2|$.

Without ambiguity, in the following, we use uppercased letters $A, B, C \dots$ for depicting individual items, the form like (ABC) for depicting individual itemsets, and the form like $\langle (A)(AC)(BC) \rangle$ for depicting individual sequences.

Now we introduce the *occurrence relation* between subsequences which are contained in a sequence. Let us consider a sequence s , where $s_1, s_2 \sqsubseteq s$ are two subsequences of s and assume that s_1 occurs before the occurrence of s_2 in s . The expression $\langle \mathbf{op}, n \rangle$ represents the constraint on the length of sequences, where $\mathbf{op} \in \{\neq, =, <, \leq, >, \geq\}$ is an operator and $n \in \mathbb{N}$ is an integer. Let $|s'| \models \langle \mathbf{op}, n \rangle$ denotes that the length of sequence s satisfies $\langle \mathbf{op}, n \rangle$, then, for example, we have $|\langle (A)(B)(C) \rangle| \models \langle >, 2 \rangle$ and $|\langle (A)(B) \rangle| \not\models \langle >, 2 \rangle$. The form $s_1 \mapsto^{\langle \mathbf{op}, n \rangle} s_2$ denotes when s_1 and s_2 occur in a sequence s , there must exist a segment g between the occurrences of s_1 and s_2 whose length satisfies the expression $\langle \mathbf{op}, n \rangle$. For simplicity, the form $s_1 \mapsto s_2$ denotes that the occurrence of s_1 is directly followed by the occurrence of s_2 in the sequence s . We thus have $\langle s_1 \mapsto^{\langle =, 0 \rangle} s_2 \rangle \equiv \langle s_1 \mapsto s_2 \rangle$.

As the generalized case, the form $s_1 \mapsto^* s_2$ denotes that s_2 occurs after the occurrence of s_1 in s , that is, $\exists s'$ such that $|s'| \geq 0$ and $\langle s_1 \mapsto s' \mapsto s_2 \rangle \sqsubseteq s$.

3.2 Belief Base and Unexpectedness of Sequences

We interpret the domain knowledge as causal relationships between the occurrences of elements in a sequence, that depend on two facts contained in the sequence: the occurrence relation and the semantics.

Before we give the definition of the belief on sequences, we first introduce the *sequence rule* between two sequences, denoted as $LHS \Rightarrow RHS$. We borrow the terms from [5] so that we call the sequence LHS the premise and the sequence RHS the conclusion. The semantics of a sequence rule $LHS \Rightarrow RHS$ is that in a sequence s , the occurrence of subsequence $LHS \sqsubseteq s$ implies the occurrence of subsequence $RHS \sqsubseteq s$ such that $LHS \cdot RHS \sqsubseteq s$. For example, the sequence rule $\langle (iMac) \rangle \Rightarrow \langle (iPhone) \rangle$ can be that the purchase of an iMac computer implies a purchase of iPhone cell phone later.

In our approach the sequence rules are given by domain experts against the occurrence relation and semantics, so that we do not consider frequency factors within such rules. In fact, we consider two constraints on such sequence rule: an occurrence relation constraint on the occurrences of LHS and RHS , and a

semantics constraint on *RHS*. Therefore, the belief on sequences can be defined as follows.

Definition 1 (Belief). A belief b on sequences is a pair $b = (p, \mathcal{C})$ such that $p : s_\alpha \Rightarrow s_\beta$ and $\mathcal{C} : \{\tau, \eta\}$ where $\tau : \langle \mathbf{op} \ n \rangle$, $\mathbf{op} \in \{\neq, =, <, \leq, >, \geq\}$, $n \in \mathbb{N}$ and $\eta : s_\beta \not\sim s_\gamma$. p and τ provide a rule on sequences $s_\alpha \mapsto^{\langle \mathbf{op} \ n \rangle} s_\beta$, and η specifies that the occurrence of s_β cannot be replaced by an occurrence of s_γ . The belief b is denoted as $[s_\alpha; s_\beta; s_\gamma; \tau]$. In the case where $\tau : \langle \geq, 0 \rangle$ we denote $*$.

The rule p defines that, in an *expected* sequence s , the occurrence of subsequence s_α should be followed by an occurrence of subsequence s_β . The occurrence relation constraint τ requires that, for an expected sequence s , there must exist sequence s' such that $|s'| \models \tau$ and $\langle s_\alpha \mapsto s' \mapsto s_\beta \rangle \sqsubseteq s$. Furthermore, the semantics constraint η ensures that, for an expected sequence s , there should not exist a sequence s' such that $|s'| \models \tau$ and $\langle s_\alpha \mapsto s' \mapsto s_\gamma \rangle \sqsubseteq s$.

Given a belief b , a sequence s is *unexpected* if s violates any constraint introduced by b , and such a behavior interpreted by s is an *unexpectedness*.

Example 1. Given a belief $b = [\langle (A)(B) \rangle; \langle (C)(D) \rangle; \langle (E)(F) \rangle; < 2]$, the sequence $s_1 = \langle (A)(AB)(E)(C)(DE) \rangle$ is expected to b since between the occurrence of $\langle (A)(B) \rangle$ and $\langle (C)(D) \rangle$ there exists $\langle (E) \rangle$ and $|\langle (C) \rangle| < 2$; the sequence $s_2 = \langle (A)(B)(E)(D)(C)(D) \rangle$ is unexpected to b because of a contradiction of the occurrence relation constraint; the sequence $s_3 = \langle (A)(B)(C)(CE)(F) \rangle$ is unexpected to b because of a contradiction of the semantics constraint; the sequence $s_4 = \langle (A)(B)(E)(F)(C)(D) \rangle$ is unexpected to b because of contradictions of both of the occurrence relation constraint and the semantics constraint; the sequence $s_5 = \langle (A)(C)(B)(E) \rangle$ is not addressed by belief b . \square

Now let us consider a belief $[s_\alpha; s_\beta; s_\gamma; *]$ where the occurrence relation constraint is $*$. Addressed by such a belief, a sequence s is unexpected if $s_\alpha \sqsubseteq s$, $\langle s_\alpha \mapsto^* s_\beta \rangle \sqsubseteq s$, and in such an unexpected sequence s the occurrence relation constraint between s_α and s_β is broken because there does not exist s_β after the occurrence of s_α in s at all. In this case the occurrence relation is not complete because of the lack of s_β . So that we classify the contradictions of a belief into three groups of unexpectedness in order to avoid such ambiguous statement on formalizations.

Definition 2 (α -unexpectedness of sequence). Given a belief $b = [s_\alpha; s_\beta; s_\gamma; *]$ and a sequence s , if there exists s_α such that $s_\alpha \sqsubseteq s$ and there does not exist s_β, s_γ such that $\langle s_\alpha \mapsto^* s_\beta \rangle \sqsubseteq s$ or $\langle s_\alpha \mapsto^* s_\gamma \rangle \sqsubseteq s$, then s contains the α -unexpectedness with respect to belief b , and s is an α -unexpected sequence.

In an α -unexpectedness of belief $[s_\alpha; s_\beta; s_\gamma; *]$, the sequence s_α is the primary factor, and s_β and s_γ should not occur after the occurrences of s_α . Many real world problems can be handled by the α -unexpectedness. For example, here we use a shorthand notation $\langle X \rangle$ for sequence $\langle (X) \rangle$, the belief $[\langle Login \rangle; \langle Logout \rangle; \emptyset; *]$ states that a valid user session should contains a “Logout” action; the belief $[\langle Mac \rangle; \langle iPhone \rangle; \langle Windows Mobile \rangle; *]$ states that we expect a customer who have purchased an iMac computer to purchase a iPhone cell phone; and the belief $[\langle noun \rangle; \langle verb \rangle; \langle noun \rangle; *]$ states that a verb is expected to appear after a noun in a sentence.

Definition 3 (β -unexpectedness of sequence). *Given a belief $b = [s_\alpha; s_\beta; s_\gamma; \tau]$ where the occurrence relation constraint τ is not $*$, and a sequence s , if there exist s_α, s_β such that $\langle s_\alpha \mapsto^* s_\beta \rangle \sqsubseteq s$ and there does not exist s' such that $|s'| \models \tau$ and $\langle s_\alpha \mapsto s' \mapsto s_\beta \rangle \sqsubseteq s$, then s contains the β -unexpectedness with respect to belief b , and s is a β -unexpected sequence.*

In Example 1, the sequences s_2 and s_4 correspond to the β -unexpectedness stated by the belief b .

Definition 4 (γ -unexpectedness of sequence). *Given a belief $b = [s_\alpha; s_\beta; s_\gamma; \tau]$ and a sequence s , if there exist s_α, s_γ such that $\langle s_\alpha \mapsto^* s_\gamma \rangle \sqsubseteq s$ and there exists s' such that $|s'| \models \tau$ and $\langle s_\alpha \mapsto s' \mapsto s_\gamma \rangle \sqsubseteq s$, then s contains the γ -unexpectedness with respect to belief b , and s is an γ -unexpected sequence.*

Without losing generality, we denote an unexpectedness as u so that $u \in \{\alpha, \beta, \gamma\}$. In additional, we denote an unexpectedness u addressed by a belief b as $u \vdash b$. Given a belief b , a u -unexpected sequence s supports unexpectedness u , that is, u is interpreted by s , denoted as $s \models u$. An unexpected sequence can support at most two unexpectedness stated by a specific belief, that is, $s \models u \in \{\alpha, \gamma\}$, or $s \models u \in \{\beta, \gamma\}$. The following examples show the $\{\alpha, \gamma\}$ and $\{\beta, \gamma\}$ unexpectedness pairs.

Example 2. Let us consider again the instance illustrated in Section 1. Let M depict iMac computer, P depict iPhone cell phone, W depict Windows Mobile cell phone, a belief on the customer transaction database can be $b = [\langle M \rangle; \langle P \rangle; \langle W \rangle; *]$. This belief states that after a customer has purchased an iMac computer, she/he is expected to purchase an iPhone cell phone later. A customer transaction sequence s is unexpected if s shows that this customer would purchase neither an iPhone cell phone nor a Windows Mobile cell phone after purchasing an iMac computer ($\alpha \vdash b$), or shows that this customer purchased an iMac computer and then purchased a Windows Mobile cell phone ($\gamma \vdash b$). It is easy to see that both of the α and the γ stated by b are interesting to the store to send correct promotion information to customers. \square

Example 3. Now let us consider a WebMail system, where a valid user login process (depicted as I) should redirect the user session to the mailbox page (depicted as M). A belief can be $[\langle L \rangle; \langle M \rangle; \langle O \rangle; = 0]$ where O depicts the logout page. A β -unexpectedness $\beta \vdash b$ that the login process does not redirect the user session to the mailbox page should not happen if everything goes normally. A γ -unexpectedness $\gamma \vdash b$ that the login process redirect the user session to the logout page may be caused by service failures. \square

Note a sequence is not restricted to support only one unexpectedness, but an α -unexpectedness and a β -unexpectedness addressed by the same belief cannot appear together, that is, $\alpha \vdash b \implies \beta \not\vdash b$ and $\beta \vdash b \implies \alpha \not\vdash b$. Additionally, it is not difficult to see that generally in a belief $[s_\alpha; s_\beta; s_\gamma; \tau]$, s_γ should not be contained in s_β , otherwise all unexpected sequences that support β -unexpectedness are also expected sequences. We say that a belief $b = [s_\alpha; s_\beta; s_\gamma; \tau]$ is *consistent* if $s_\gamma \not\sqsubseteq s_\beta$. Without special descriptions, we only consider consistent beliefs.

Given a set \mathcal{B}_H of consistent beliefs base \mathcal{B} , if for any two beliefs $b_i, b_j \in \mathcal{B}$ wherer $b_i = [s_{\alpha_i}; s_{\beta_i}; s_{\gamma_i}; \tau_i]$ and $b_j = [s_{\alpha_j}; s_{\beta_j}; s_{\gamma_j}; \tau_j]$, we have $s_{\alpha_i} = s_{\alpha_i}$,

$\tau_j = \tau_j$, $s_{\beta_i} \not\sqsubseteq s_{\beta_j}$ and $s_{\gamma_i} \not\sqsubseteq s_{\gamma_j}$, we say that \mathcal{B}_H is *homologous*. Given a set \mathcal{B}_H of homologous beliefs, a sequence s is a *local unexpected sequence* to b if s is unexpected to at least one belief $b \in \mathcal{B}_H$, and s is a *global unexpected sequence* to \mathcal{B}_H if s is unexpected to each belief $b \in \mathcal{B}_H$.

Example 4. Let $\mathcal{B} = \{b_1 = [\langle(A)\rangle; \langle(B)\rangle; \emptyset; *], b_2 = [\langle(A)\rangle; \langle(C)\rangle; \emptyset; *]\}$, then b_1 and b_2 are homologous. Given two sequences $s_1 = \langle(A)(D)(B)\rangle$ and $s_2 = \langle(A)(D)(D)\rangle$, s_1 is a local unexpected sequence to b_2 but expected to b_1 , s_2 is a global unexpected sequence to \mathcal{B} , and local unexpected sequence to both of b_1 and b_2 . \square

3.3 Unexpected Sequential Patterns and Rules

We consider the notion of unexpected sequential patterns within the framework of mining sequential patterns, that is, an unexpected sequential pattern is a maximal frequent sequence contained in a set of unexpected sequences stated by a specific unexpectedness. Given a sequence database \mathcal{D} and a belief base \mathcal{B} , we use the factor support for measuring the interestingness of an unexpected sequence. For each unexpectedness u addressed by each belief $b \in \mathcal{B}$, let \mathcal{D}_u be the set of all unexpected sequences stated by u in \mathcal{D} , the support is the fraction of the total number of unexpected sequences in \mathcal{D}_u that support u , that is,

$$\text{supp}(s_u) = \frac{|\{s \in \mathcal{D}_u \mid s_u \sqsubseteq s\}|}{|\mathcal{D}_u|}.$$

Definition 5 (Unexpected Sequential Pattern). *Given a sequence database \mathcal{D} and an unexpectedness u , let \mathcal{D}_u be a subset of \mathcal{D} such that for each sequence $s \sqsubseteq \mathcal{D}$ we have $s \models u$. An unexpected sequential pattern of unexpectedness u is a maximal frequent sequence contained in \mathcal{D}_u whose support satisfies a user defined minimum support threshold.*

Unexpected sequential patterns reflect the occurrence dependencies between elements of unexpected sequences stated by an unexpectedness u , so that an unexpected sequential pattern is not required to have the same structure as the unexpectedness u . Example 5 illustrates this property.

Example 5. Given a belief $b = [\langle A \rangle; \langle B \rangle; \langle C \rangle; = 1]$, let us consider three sequences unexpected to b that support $\beta \vdash b$:

$$\begin{aligned} s_1 &= \langle (D)(AB)(CD)(D)(BC)(E) \rangle, \\ s_2 &= \langle (D)(AB)(D)(E)(BD)(E) \rangle, \\ s_3 &= \langle (C)(A)(E)(E)(B)(C) \rangle. \end{aligned}$$

With a minimum support value 0.5, the sequence $\langle (D)(AB)(D)(B)(E) \rangle$ is an unexpected sequential pattern corresponding to $\beta \vdash b$. \square

Before we introduce unexpectedness rules, we first introduce the notion of *bordered unexpected sequence*, which is helpful to identify different parts in an unexpected sequence.

We are given a belief $b = [s_\alpha; s_\beta; s_\gamma; \tau]$. If $\alpha \vdash b$, for an unexpected sequence $s \models \alpha$, then there exist two segments $g', g \sqsubseteq s$ that $|g'| \geq 0$ and $|g| \geq 0$ such that $|g' \cdot s_\alpha \cdot g| = |s|$. We define the bordered unexpected sequence of α

unexpectedness as the segment $s_b \sqsubseteq s$ that $s_\alpha \cdot g \sqsubseteq s_b$ and $|s_\alpha \cdot g| = |s_b|$. If $\beta \vdash b$, for an unexpected sequence $s \models \beta$, then there exist segment $g \sqsubseteq s$ where $|g| \geq 0$ such that $s_\alpha \cdot g \cdot s_\beta \sqsubseteq s$ violates the belief b . We define the bordered unexpected sequence of β -unexpectedness as the segment $s_b \sqsubseteq s$ such that $s_\alpha \cdot g \cdot s_\beta \sqsubseteq s_b$ and $|s_\alpha \cdot g \cdot s_\beta| = |s_b|$. If $\gamma \vdash b$, for an unexpected sequence $s \models \gamma$, then there exist segment $g \sqsubseteq s$ where $|g| \geq 0$ such that $s_\alpha \cdot g \cdot s_\gamma \sqsubseteq s$ violates the belief b . We define the bordered unexpected sequence of γ -unexpectedness as the segment $s_b \sqsubseteq s$ that $s_\alpha \cdot g \cdot s_\gamma \sqsubseteq s_b$ and $|s_\alpha \cdot g \cdot s_\gamma| = |s_b|$.

Example 6. As shown in Example 5, the sequences s_1, s_2 and s_3 are unexpected to the belief $b = [\langle A \rangle; \langle B \rangle; \langle C \rangle; = 1]$. We have sequences $\langle (AB)(CD)(D)(BC) \rangle$, $\langle (AB)(D)(E)(BD) \rangle$ and $\langle (A)(E)(E)(B) \rangle$ are bordered unexpected sequences corresponding to s_1, s_2 and s_3 . \square

An unexpected sequence s can therefore be represented as $s = g_a \cdot s_b \cdot g_c$ where s_b is a bordered unexpected sequence corresponding to the specific unexpectedness and g_a, g_c are two segments of s . We have $|s_b| > 0$, $|g_a| \geq 0$ and $|g_c| \geq 0$. The segment $g_a \sqsubseteq s$ is called the *antecedent sequence* and the segment $g_c \sqsubseteq s$ is called the *consequent sequence*. Given a set of unexpected sequences that support unexpectedness $u \in \{\alpha, \beta, \gamma\}$, we denote the set of all antecedent sequences, including empty ones, as \mathcal{D}_u^a , and denote the set of all consequent sequences, including empty ones, as \mathcal{D}_u^c . The support of a sequence s_a contained in \mathcal{D}_u^a and s_c contained \mathcal{D}_u^c is the fraction of total sequences of \mathcal{D}_u^a or of \mathcal{D}_u^c that support s_c or s_a , that is,

$$\text{supp}(s_a) = \frac{|\{s \in \mathcal{D}_u^a | s_a \sqsubseteq s\}|}{|\mathcal{D}_u^a|}$$

and

$$\text{supp}(s_c) = \frac{|\{s \in \mathcal{D}_u^c | s_c \sqsubseteq s\}|}{|\mathcal{D}_u^c|}.$$

A maximal frequent sequence contained in \mathcal{D}_u^a is a *frequent antecedent sequence* and a maximal frequent sequences contained in \mathcal{D}_u^c is a *frequent consequent sequence*.

Definition 6 (Antecedent Rule). *Given a set \mathcal{D}_u of unexpected sequences supporting unexpectedness $u \in \{\alpha, \beta, \gamma\}$, let \mathcal{D}_u^a be the set of all antecedent sequences contained in \mathcal{D}_u and let s_a be a frequent antecedent sequence contained in \mathcal{D}_u^a with respect to a user defined minimum support threshold σ_a , an antecedent rule is the rule $s_a \Rightarrow u$.*

Antecedent rules reflect the elements in a sequence that anticipates an unexpectedness addressed by a given belief. So that, with an antecedent rule, we can state the causality of an unexpected behavior in sequences.

Definition 7 (Consequent Rule). *Given a set \mathcal{D}_u of unexpected sequences supporting unexpectedness $u \in \{\alpha, \beta, \gamma\}$, let \mathcal{D}_u^c be the set of all consequent sequences contained in \mathcal{D}_u and let s_c be a frequent consequent sequence contained in \mathcal{D}_u^c with respect to a user defined minimum support threshold σ_c , an consequent rule is the rule $u \Rightarrow s_c$.*

Consequent rules reflect the elements in a sequence that are resulted by an unexpectedness addressed by a given belief. With a consequent rule we can state the influence of an unexpected behavior in sequences.

We characterize the antecedent rules and consequent rules on terms of *support* and *confidence*. The value of the support of an antecedent rule $s_a \Rightarrow u$ equals the value of the support of s_a and the value of support of a consequent rule $u \Rightarrow s_c$ equals the value of the support of s_c . So that we have $supp(s_a \Rightarrow u) = supp(s_a)$ and $supp(u \Rightarrow s_c) = supp(s_c)$.

Given a sequence database \mathcal{D} and an unexpectedness u that states antecedent sequence set \mathcal{D}_u^a and consequent sequence set \mathcal{D}_u^c , the confidence of an antecedent rule is defined as:

$$conf(s_a \Rightarrow u) = \frac{|\{s \in \mathcal{D}_u^a | s_a \sqsubseteq s\}|}{|\{s \in \mathcal{D} | s_a \sqsubseteq s\}|},$$

and the confidence of a consequent rule is defined as:

$$conf(u \Rightarrow s_c) = \frac{|\{s \in \mathcal{D}_u^c | s_c \sqsubseteq s\}|}{|\{s \in \mathcal{D} | s \models u\}|}.$$

We have the value of the confidence of an consequent rule equals the value of the support of the frequent consequent sequences involved.

Example 7. Let us consider again Example 4. Assume a log file containing 10,000 user sessions of $(Time, IP, Request)$ that *Time* identifies a time range, *IP* identifies an IP range, and *Request* identifies the resources requested that $Request \in \{Begin, End, Help, Login, Logout, Recall, \dots\}$, where *Recall* depicts the password recall page and *Help* depicts the online help page. In such a log file, each user session is a sequence.

With the belief given in Example 4, assume that we found 100 sequences support the β -unexpectedness, then we have $supp(\beta) = 0.01$. Let $\sigma_a = \sigma_c = 0.1$, assume that we found that 80 sequences support the frequent antecedent sequence $\langle(t1, ip1, Begin)\rangle$; 10 sequences support the frequent antecedent sequence $\langle(ip2, Begin)\rangle$; 80 sequences support the frequent consequent sequence $\langle(t1, ip1, End)\rangle$; 15 sequences support the frequent consequent sequence $\langle(ip2, Recall)(ip2, End)\rangle$; 10 sequences support the frequent consequent sequence $\langle(ip2, Help)(ip2, End)\rangle$.

So that we have the antecedent rule $\langle(t1, ip1, Begin)\rangle \models \beta$ has support value 0.8; the antecedent rule $\langle(ip2, Begin)\rangle \models \beta$ has support 0.1; the consequent rule $\beta \models \langle(t1, ip1, End)\rangle$ has support 0.8; the consequent rule $\beta \models \langle(ip2, Recall)(ip2, End)\rangle$ has support 0.15. Furthermore, assume that the total number of sequences supporting $\langle(t1, ip1, Begin)\rangle$ and $\langle(t1, ip1, End)\rangle$ are both 80, then we have that the confidence of rule $\langle(t1, ip1, Begin)\rangle \models \beta$ is 1; the confidence of rule $\beta \models \langle(t1, ip1, End)\rangle$ is 0.8. Assume that the total number of sequences supporting $\langle(ip2, Begin)\rangle$ is 9000, then the confidence of rule $\langle(ip2, Begin)\rangle \models \beta$ is $1/900$ which can be ignored. Obviously, in this example, the connections from IP range 1 at time range 1 can be considered as attacks and we have very strong rules to confirm this behavior. \square

3.4 The Algorithm

We propose the algorithm USER for finding unexpected sequential patterns and unexpectedness rules from a sequence database with respect to a user defined

belief base. The algorithm first extracts all global unexpected sequences from a sequence database for each unexpectedness addressed by the belief base, then finds all unexpected sequential patterns and sequence rules from each set of unexpected sequences with respect to user defined support/confidence threshold values. We present the algorithm USER shown in Algorithm 1.

The algorithm accepts a sequence database \mathcal{D} and a belief base \mathcal{B} , the minimum support values σ_u , σ_a and σ_c , the minimum confidence values δ_a and δ_c as input, and produces the unexpected sequence set \mathcal{D}_u , the antecedent sequence set \mathcal{D}_u^a and the consequent sequence set \mathcal{D}_u^c for each unexpectedness u addressed by the belief base \mathcal{B} .

Algorithm 1 Algorithm USER

Input: A sequence database \mathcal{D} and a belief base \mathcal{B} , minimum support values σ_u , σ_a , σ_c , and minimum confidence values δ_a , δ_c

Output: The set \mathcal{P}_u of unexpected sequential patterns, the set \mathcal{R}_u^a of antecedent rules and \mathcal{R}_u^c of consequent rules for each unexpectedness u addressed by belief base \mathcal{B}

```

1: while  $s = \text{getseq}(\mathcal{S})$ 
2:   while  $s_\alpha = \text{getnode}(\mathcal{B}, \emptyset)$ 
3:     if  $o_\alpha = \text{find\_alpha}(s, s_\alpha)$ 
4:       while  $\tau = \text{getnode}(\mathcal{B}, s_\alpha)$  begin
5:         while  $s_\beta = \text{getnode}(\mathcal{B}, \tau)$ 
6:           if  $\tau == *$  begin
7:             if  $o_\beta = \text{matchf}(s, s_\beta, \tau, o_\alpha)$ 
8:               failed and continue
9:             else begin
10:              if  $o_\beta = \text{matchf}(s, s_\beta, \tau, o_\alpha)$  begin
11:                 $\mathcal{D}_\beta \leftarrow \mathcal{D}_\beta \cup s$ 
12:                 $\mathcal{D}_\beta^a \leftarrow \mathcal{D}_\beta^a \cup \text{subseq}(s, s.\text{begin}, o_\beta.\text{begin})$ 
13:                 $\mathcal{D}_\beta^c \leftarrow \mathcal{D}_\beta^c \cup \text{subseq}(s, o_\beta.\text{end}, s.\text{end})$ 
14:              end
15:              while  $s_\gamma = \text{getnode}(\mathcal{B}, s_\beta)$ 
16:                if  $o_\gamma = \text{matchf}(s, s_\gamma, \tau, o_\alpha)$  begin
17:                   $\mathcal{D}_\gamma \leftarrow \mathcal{D}_\gamma \cup s$ 
18:                   $\mathcal{D}_\gamma^a \leftarrow \mathcal{D}_\gamma^a \cup \text{subseq}(s, s.\text{begin}, o_\gamma.\text{begin})$ 
19:                   $\mathcal{D}_\gamma^c \leftarrow \mathcal{D}_\gamma^c \cup \text{subseq}(s, o_\gamma.\text{end}, s.\text{end})$ 
20:                end
21:              end
22:            if  $\tau == *$  begin
23:               $\mathcal{D}_\alpha \leftarrow \mathcal{D}_\alpha \cup s$ 
24:               $\mathcal{D}_\alpha^a \leftarrow \mathcal{D}_\alpha^a \cup \text{subseq}(s, s.\text{begin}, o.\text{begin})$ 
25:               $\mathcal{D}_\alpha^c \leftarrow \mathcal{D}_\alpha^c \cup \text{subseq}(s, o.\text{end}, s.\text{end})$ 
26:            end
27:          end
28: for each set of  $\mathcal{D}$ , find and output unexpected sequential patterns with  $\sigma_u$ 
29: for each group of  $\mathcal{D}^a$  and  $\mathcal{D}^c$ , find and output rules with  $\sigma_a$ ,  $\sigma_c$  and  $\delta_a$ ,  $\delta_c$ 

```

For each sequence $s \in \mathcal{D}$, a belief $b = [s_\alpha; s_\beta; s_\gamma; \tau]$ and an unexpectedness $u \vdash b$, the fact $s \models u$ can be determined by different cases. To generalize the

problem, let us first consider the occurrence of sequence s' in sequence s where $s' \sqsubseteq s$. According to our formal model of sequence, if $s' \sqsubseteq s$, then there exist at least one occurrence of s' in s . However, an itemset $\mathcal{I}_i \in s'$ might be *redundant* in such an occurrence. Given a sequence s and two subsequences $s', s'' \sqsubseteq s$ where $s' = \langle \mathcal{I}'_1 \dots \mathcal{I}'_{n'} \rangle$ and $s'' = \langle \mathcal{I}''_1 \dots \mathcal{I}''_{n''} \rangle$, if we have $s' \sqsubseteq s''$ and $|s'| < |s''|$, then s'' is a *redundant occurrence* of s' in s ; otherwise if we have $|s'| = |s''|$, then s'' is an *irredundant occurrence* of s' in s .

In our current approach we find the first irredundant occurrence of s_α and find any occurrence of s_β or of s_γ with respect to the occurrence relation after the occurrence of s_α .

For α -unexpectedness, the function *matchi* finds the first irredundant occurrence of $s_\alpha \sqsubseteq s$ and then ensures that $\langle s_\alpha \mapsto^* s_\beta \rangle \not\sqsubseteq s$ and $\langle s_\alpha \mapsto^* s_\gamma \rangle \not\sqsubseteq s$ by the function *match*. For β and γ unexpectedness, we need to find a segment $g_u \sqsubseteq s$ with occurrence relation constraint between the occurrences of s_α and s_β , or of s_α and s_γ , that is, to find a sequence s' such that $|s'| \not\models \tau$ and $\langle s_\alpha \mapsto s' \mapsto s_\beta \rangle \sqsubseteq s$, or such that $|s'| \models \tau$ and $\langle s_\alpha \mapsto s' \mapsto s_\gamma \rangle \sqsubseteq s$. So that after the function *matchi* finding the first irredundant occurrence of $s_\alpha \sqsubseteq s$, the function *matchf* finds the first occurrence of $s_\beta \sqsubseteq s$ or $s_\gamma \sqsubseteq s$ by using the occurrence of s_α and the occurrence relation constraint τ as additional parameters.

For each $u \vdash b$ where $b \in \mathcal{B}$, this algorithm first finds unexpected sequential patterns from the unexpected sequence set \mathcal{D}_u with respect to σ_u , then finds sequential patterns from the sequence sets \mathcal{D}_u^a and \mathcal{D}_u^c with respect to σ_a and σ_c . Finally the algorithm generates antecedent rules and consequent rules for each unexpectedness u with respect to δ_a and δ_c .

We use general purposed sequential pattern mining approach for finding maximal frequent sequences with a minimum support value. Many efficient approaches have been proposed and developed for sequential pattern mining, such as the PSP approach proposed by [6], the SPADE approach proposed by [7] and the PrefixSpan approach proposed by [8].

4 Experiments

To evaluate our approach we have performed two groups of experiments. The first group of experiments are performed to extract unexpected sequential patterns and unexpectedness rules from a large log file of a real Web server, where the belief base is defined by domain experts. The second group of experiments are considered as scalability tests against various dense synthetic data files generated by the IBM Quest Synthetic Data Generator¹, 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.

We analyzed a large log file, that contains 2,271,955 access records from a real Web server, with the USER approach. The log file has been preprocessed to be a sequence database that contains 67,228 sequences corresponding to 27,552 distinct items.

¹<http://www.almaden.ibm.com/cs/quest/>

N.B.	U.S.	U.P.	U.R.
5	48	4	3
10	240	11	19
15	591	15	23
20	673	22	30

N.B. : Number of Beliefs
U.S. : Number of Unexpected Sessions
U.P. : Number of Unexpected Sequential Patterns
U.R. : Number of Unexpectedness Rules

Table 1: Experimental results to real data from a Web server.

Table 1 shows our three classes of experiments on real data with different beliefs.

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.

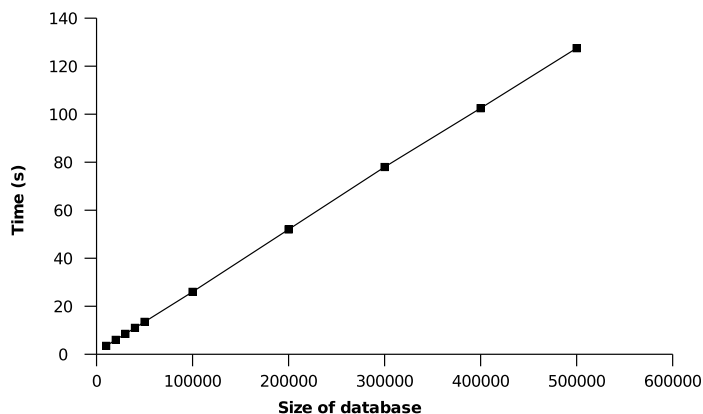


Figure 1: Time for extracting all unexpected sequences stated by 20 beliefs.

Figure 1 shows that, when the belief number is fixed, the extracting time of all unexpected sequences increases linearly with the increasing of the size of sequence database.

Figure 2 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 would 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 3 shows the increment of extracting time of all unexpected sequences illustrated in Figure 2, and from which we can find that the increasing rate of extracting time depends on the number of unexpected sequences. In fact, 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.

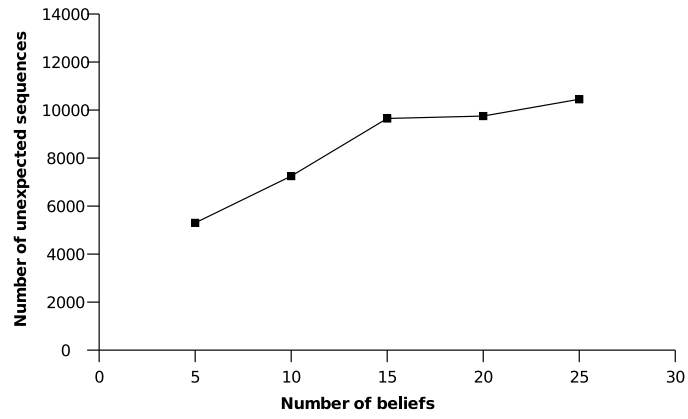


Figure 2: Number of all unexpected sequences in 100,000 sequences.

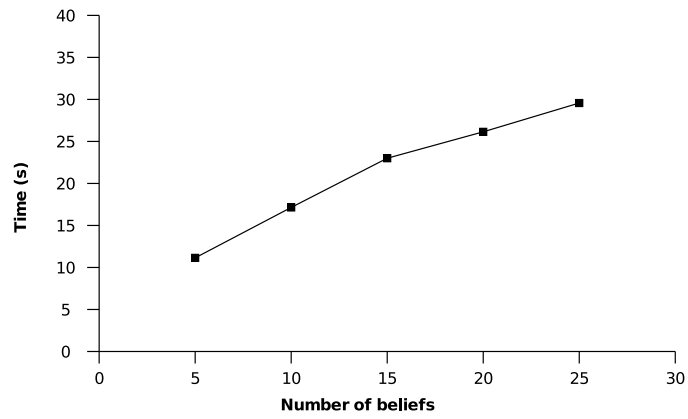


Figure 3: Time for extracting all unexpected sequences from 100,000 sequences.

5 Conclusions

We introduced the problem of mining unexpected sequential patterns and unexpectedness rules in a sequence database with respect to a user defined belief base, that states the unexpected behaviors within the context of sequence mining. We proposed the approach USER for resolving this problem. We first introduced the belief base and unexpectedness of sequences within the formal models that we considered with sequence databases, we then proposed the unexpected sequential patterns and unexpectedness rules including antecedent rules and consequent rules for measuring the unexpected behaviors in sequence mining. Our experiments show the USER approach is robust.

Our future research in unexpected sequence mining includes several aspects. We are interested in mining hierarchised unexpectedness rules. Furthermore, we are interested in generating belief bases by using objective measures. We are also interested in extending the notion of “unexpectedness” to general sequence mining process, that is, dynamically increase the belief base with new-found unexpectedness rules.

References

- [1] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *ICDE*, pages 3–14, 1995.
- [2] Ken McGarry. A survey of interestingness measures for knowledge discovery. *Knowl. Eng. Rev.*, 20(1):39–61, 2005.
- [3] Abraham Silberschatz and Alexander Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *KDD*, pages 275–281, 1995.
- [4] Balaji Padmanabhan and Alexander Tuzhilin. On characterization and discovery of minimal unexpected patterns in rule discovery. *IEEE Trans. Knowl. Data Eng.*, 18(2):202–216, 2006.
- [5] Myra Spiliopoulou. Managing interesting rules in sequence mining. In *PKDD*, pages 554–560, 1999.
- [6] Florent Masseglia, Fabienne Cathala, and Pascal Poncelet. The psp approach for mining sequential patterns. In *PKDD*, pages 176–184, 1998.
- [7] Mohammed Javeed Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.
- [8] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Meichun Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Trans. Knowl. Data Eng.*, 16(11):1424–1440, 2004.
- [9] Minos N. Garofalakis, Rajeev Rastogi, and Kyuseok Shim. Spirit: Sequential pattern mining with regular expression constraints. In *VLDB*, pages 223–234, 1999.