Discovering Fuzzy Unexpected Sequences with Beliefs

Dong (Haoyuan) Li
LGI2P - EMA
Parc scientifique G. Besse
30035 Nîmes Cedex 1, FR
Haoyuan.Li@ema.fr

Anne Laurent
LIRMM - CNRS - UM2
161 rue Ada
34392 Montpellier Cedex 5, FR
laurent@lirmm.fr

Pascal Poncelet
LGI2P - EMA
Parc scientifique G. Besse
30035 Nîmes Cedex 1, FR
Pascal.Poncelet@ema.fr

Abstract

In this paper we present a novel approach for discovering fuzzy unexpected sequences, such as *certainly* unexpected, *almost* unexpected and *a little* unexpected, from databases with respect to user defined beliefs. We first formalize the belief on sequences and the different types of unexpectedness, then we detail the algorithm Taufu that finds fuzzy unexpected sequences with beliefs. Our approach has been verified with various experiments.

Keywords: Data Mining, Belief, Fuzzy Unexpected Sequence.

1 Introduction

As the one most concentrated in KDD and data mining research, the sequential pattern mining [1] gives a frequency based view of the correlations between elements contained in sequences. However, when we consider domain knowledge (in this paper we interpret knowledge as *beliefs*) within the discovery, most of the frequent sequences might have already been confirmed, and in many cases the most interested are the sequences that contradict existing knowledge.

For instance, in Web site log analysis, a belief may require that the access of `home.php` should be followed, but not directly (considering online statistic and advertisement systems involved in the same session), by the access of `login.php`, and the access of `login.php` should not be replaced by the access of `logout.php`. So that an expected sequence like "access of `home.php` is followed by `stats.cgi` then followed by `ad.cgi` and then followed by `login.php`" may have strong frequency support, and an unexpected sequence like "access of `home.php` is directly followed by `login.php`" will be hidden by the sequential pattern model since it is included in expected ones. Furthermore, another unexpected sequence like "access of `home.php` is followed by `stats.cgi` then followed by `logout.php`" may have weak support and be difficult to be discovered by frequency based criteria.

On the other hand, even though we know that the access of `home.php` could not be directly followed by the access of `login.php`, it is difficult to point out how many elements should *exactly* occur between them, since the number of involved online statistic and advertisement systems may be uncertain. It is therefore necessary to consider fuzzy unexpectedness with beliefs to respect such uncertain occurrences, such as "access of `home.php` is directly followed by `login.php`" is *certainly unexpected*, "access of `home.php` is followed by `login.php` after 1 elements" is *almost unexpected*, and "access of `home.php` is followed by `login.php` after 3 elements" is *a little unexpected*.

In this paper, we propose a novel approach, Taufu (τ **fu**zzy), for discovering fuzzy unexpected sequences from databases with respect to user defined beliefs. The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 presents our approach Taufu. Section 4 shows our experimental results. The conclusion and our future research directions are listed in Section 5.

2 Related Work

The interestingness measures for data mining can be classified as objective measures and subjective measures [7]. Objective measures typically depend on the structure of extracted patterns, and the criteria based on probability and statistics approaches like support and confidence; subjective measures are generally user and knowledge oriented, such criteria can be actionability, unexpectedness etc.. The belief driven unexpectedness is first introduced by [9] as a subjective measure where beliefs are categorized to hard beliefs and soft beliefs.

In the most recent approach to semantics based unexpected association rule discovery presented by [8], a belief is represented as a rule. For example, the belief $professional \to weekend$ shows that professionals do shopping at weekend, and a rule $Dec. \to weekday$, shows that in December people do shopping at weekday, is unexpected to the belief $professional \to weekend$ (since $weekend$ semantically contradicts $weekday$) if: (a) the rule $Dec. \cup professional \to weekday$ satisfies given support/confidence threshold values; (b) the rule $Dec. \cup professional \to weekend$ does not satisfy given minimum support/confidence.

On unexpected sequence discovery, [10] proposed an approach based on beliefs constrained by frequency. Given a belief, if the support/confidence values of specified subsequences within a frequent sequence do not satisfy frequency constraints introduced by the belief, then such a frequent sequence is unexpected. On the other hand, various fuzzy approaches have been proposed on discoveries of sequential patterns [3, 2, 5, 11, 4], most of them focused on finding frequent sequences with fuzzy quantity on each items, like "60% of people who eat *a lot of* candies purchase *few* potato chips"

We are concentrating on finding unexpected sequences with semantics and occurrence based fuzzy beliefs.

3 Taufu: An Approach for Fuzzy Unexpected Sequence Discovery

3.1 Preliminary Concepts

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

Given two sequences $s = \langle \mathcal{I}_1 \mathcal{I}_2 \ldots \mathcal{I}_m \rangle$ and $s' = \langle \mathcal{I}'_1 \mathcal{I}'_2 \ldots \mathcal{I}'_n \rangle$, if there exist integers $1 \leq i_1 < i_2 < \ldots < i_m \leq n$ such that $\mathcal{I}_1 \subseteq \mathcal{I}'_{i_1}, \mathcal{I}_2 \subseteq \mathcal{I}'_{i_2}, \ldots, \mathcal{I}_m \subseteq \mathcal{I}'_{i_m}$, then the sequence s is a *subsequence* of the sequence s', denoted as $s \sqsubseteq s'$. In particular, we denote the first itemset of a sequence s as $s^{\top}$ and the last itemset as $s_{\perp}$. We therefore note $s \sqsubseteq^{\top} s'$ if $s^{\top} \sqsubseteq s'^{\top}$, $s \sqsubseteq_{\perp} s'$ if $s_{\perp} \sqsubseteq s'_{\perp}$, and $s \sqsubseteq^{\top}_{\perp} s'$ if $s^{\top} \sqsubseteq s'^{\top}$ and $s_{\perp} \sqsubseteq s'_{\perp}$. If $s \sqsubseteq s'$, we say that *s is contained in s'*, or *s' supports s*.

The *support* of a sequence is defined as the fraction of total sequences in $\mathcal{D}$ that support this sequence. If a sequence s is not a subsequence of any other sequences, then we say that the sequence s is *maximal*.

The *length* of a sequence is the number of itemsets contained in the sequence, denoted as $|s|$. An *empty sequence* is denoted as $\emptyset$, we have $s = \emptyset \iff |s| = 0$. The *concatenation* of sequences is denoted as the form $s_1 \cdot s_2$, and we have $|s_1 \cdot s_2| = |s_1| + |s_2|$.

3.2 Belief on Sequences

In order to discover fuzzy unexpected sequences from databases, we first propose the semantics and occurrence constrained belief on sequences.

Definition 1 (Belief)**.** *A* belief *on sequences consists of a sequence rule $s_\alpha \Rightarrow s_\beta$ and a pair $\langle \eta, \tau \rangle$ of constraints. The rule $s_\alpha \Rightarrow s_\beta$ introduces that in a sequence s, the occurrence of $s_\alpha \sqsubseteq s$ implies an occurrence of $s_\beta \sqsubseteq s$ later.*

The pair $\langle \eta, \tau \rangle$ consists of a semantical constraint $\eta = s_\beta \not\sim s_\gamma$ and an occurrence constraint $\tau = [n_b..n_e]$ on s_β and s_γ. We denote a belief on sequences as $[s_\alpha; s_\beta; s_\gamma; \tau]$. A sequence s verifies a belief b is denoted as $s \models b$.

The semantical constraint η is a contradiction relation $\not\sim$ between two sequences, so that given two sequences s_1 and s_2, the relation $s_1 \not\sim s_2$ constrains that s_1 cannot be replaced by s_2 in any concentrated sequences. For example, as illustrated in Section 1, the access of `login.php` could not be replaced by the access of `logout.php`, so that we have `login.php` $\not\sim$ `logout.php`.

Given a sequence s, the constraint τ is an interval $[n_b..n_e]$ on two subsequences $s_1, s_2 \sqsubseteq s$ that $s_1 \mapsto^{[n_b..n_e]} s_2$, where n_b and n_e are two integers that $0 \leq n_b \leq n_e \leq *$ where $*$ stands for the end of sequence s. The constraint $s_1 \mapsto^{[n_b..n_e]} s_2$ ensures that if s_1 occurs before the occurrence of s_2 in s, then between s_1 and s_2 there should exist a sequence s' such that $n_b \leq |s'| \leq n_e$, denoted as $|s'| \models [n_b..n_e]$. We denote $s_1 \mapsto^{[0..0]} s_2$ as $s_1 \mapsto s_2$ and $s_1 \mapsto^{[0..*]} s_2$ as $s_1 \mapsto^* s_2$.

Example 1. The constraint `home.php` $\mapsto^{[3..5]}$ `login.php` requires that if `home.php` is followed by `login.php`, then between them there should be 3 to 5 occurrences of other elements. Therefore, considering the belief [`home.php`; `login.php`; `logout.php`; [3..5]], the constraint `login.php` $\not\sim$ `logout.php` further requires that if `home.php` is followed by `logout.php`, then between them there should not be 3 to 5 occurrences of other elements. □

3.3 Fuzzy Unexpected Sequences

An *unexpected sequence* is a sequence that violates the constraints introduced by a given belief. Given a belief $b = [s_\alpha; s_\beta; s_\gamma; \tau]$ and an unexpected sequence s, the constraints $\langle \eta, \tau \rangle$ can be represented as a constraint on the length of a subsequence $s' \sqsubseteq s$ such that $|s'| < n_b$ or $|s'| > n_e$ and $s_\alpha \cdot s' \cdot s_\beta \sqsubseteq s$, or such that $n_b \leq |s'| \leq n_e$ and $s_\alpha \cdot s' \cdot s_\gamma \sqsubseteq s$. We denote that s satisfies the constraint $\tau = [n_b..n_e]$, i.e. $n_b \leq |s| \leq n_e$, as $|s| \models \tau$.

We partition the satisfiability of τ into several fuzzy sets by a fuzzy membership function μ, then $s \models (\tau, U)$ denotes that the length of s satisfies the constraint τ, where U is the membership degree. Considering the possible violations of a belief $[s_\alpha; s_\beta; s_\gamma; \tau]$, we propose three types of unexpectedness.

Definition 2 (The α-unexpectedness)**.** *Given a belief $b = [s_\alpha; s_\beta; s_\gamma; *]$ and a sequence s such that $s_\alpha \sqsubseteq s$, if there does not exist s_β, s_γ such that $s_\alpha \mapsto^* s_\beta \sqsubseteq s$ or $s_\alpha \mapsto^* s_\gamma \sqsubseteq s$, then s supports α-unexpectedness, denoted as $s \models (\alpha \vdash b)$, and we say s is α-unexpected.*

The meaning of the α-unexpectedness is given by the primary factor s_α contained in such unexpected sequences. The α-unexpectedness is crisp since the constraint τ is fixed to $*$, which cannot be fuzzy.

Definition 3 (The β-unexpectedness)**.** *Given a belief $b = [s_\alpha; s_\beta; s_\gamma; \tau]$ and a sequence s such that $s_\alpha \sqsubseteq s$, if $\tau \neq *$ and there exists s_β such that $s_\alpha \mapsto^* s_\beta \sqsubseteq s$, and there does not exist s' such that $|s'| \models (\tau, U)$ and $s_\alpha \mapsto s' \mapsto s_\beta \sqsubseteq s$, then s supports β-unexpectedness, denoted as $s \models (\beta \vdash b, U)$, and we say s is β-unexpected.*

Definition 4 (The γ-unexpectedness)**.** *Given a belief $b = [s_\alpha; s_\beta; s_\gamma; \tau]$ and a sequence s such that $s_\alpha \sqsubseteq s$, if there exists s_γ such that $s_\alpha \mapsto^* s_\gamma \sqsubseteq s$ and there exists s' such that $|s'| \models (\tau, U)$ and $s_\alpha \mapsto s' \mapsto s_\gamma \sqsubseteq s$, then s supports γ-unexpectedness, denoted as $s \models (\gamma \vdash b, U)$, and we say s is γ-unexpected.*

The meaning of the β-unexpectedness and γ-unexpectedness is given by the factor s_β and s_γ contained in the unexpected sequences. Such unexpectedness can be fuzzy, where the membership degree U of unexpectedness is measured by the fuzzy membership function μ, and we have $0 < U \leq 1$. Note that we have $U \equiv 1$ for α-unexpected sequences, so that for uniforming the notations, we can also denote an α-unexpected sequence as $s \models (\alpha \vdash b, U)$ where $U = 1$ (in fact we have $\tau = * \Longrightarrow U = 1$). Without loss of generality, a sequence supporting the unexpectedness $u \in \{\alpha, \beta, \gamma\}$ stated by a belief b is denoted as $s \models (u \vdash b, U)$; a fuzzy unexpected sequence s and its membership degree U are denoted as a pair $\langle s, U\rangle$.

Example 2. Given a user defined belief $b = [\texttt{home}; \texttt{login}; \texttt{logout}; [0..5]]$ on Web site log files, we consider three fuzzy sets for the each unexpectedness, they are "little" (μ_L), "almost" (μ_A) and "certainly" (μ_C). To crisp unexpectedness, a sequence $s = \langle(\texttt{home})(\texttt{ad1})(\texttt{ad2})(\texttt{ad3})(\texttt{ad4})(\texttt{login})\rangle$ is expected since $|(\texttt{ad1})(\texttt{ad2})(\texttt{ad3})(\texttt{ad4})| = 4$ and $4 \models [0..5]$. However, let fuzzy membership functions for measuring β-unexpectedness be shown in Figure 1, we have $\mu_L(4) = 0.67$, $\mu_A(4) = 1$ and $\mu_C(4) = 0.5$, so that the best description of the sequence s is "almost" unexpected. In the fuzzy set "certainly" for such β-unexpectedness, we have $s \models (\beta \vdash b, 0.5)$ since the degree $U = \mu_C(4) = 0.5$, and we can so write sequence s as $\langle s, 0.5\rangle$ for β-unexpected of belief b. □

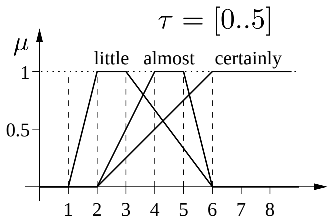

Figure 1: Fuzzy sets for β-unexpectedness with $\tau = [0..5]$.

Figure 2 and Figure 3 represent the fuzzy set "certainly" for β-unexpectedness and γ-unexpectedness with (a) $\tau = [0..3]$, (b) $\tau = [3..3]$, (c) $\tau = [3..5]$ and (d) $\tau = [3..*]$.

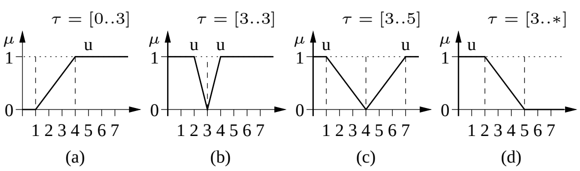

Figure 2: Fuzzy measure of the "certainly" set for β-unexpectedness.

For better describing the behaviors of all those unexpected sequences, we propose the notion of *bordered unexpected sequences.*

Definition 5 (Bordered Unexpected Sequence). *Given a belief $b = [s_\alpha; s_\beta; s_\gamma; \tau]$ and an unexpected sequence $s \models (u \vdash b, U)$, a bordered unexpected sequence s_u is the maximal subsequence of s: (1) if s is α-unexpected, we have $s' \cdot s_u = s$ ($|s'| \geq 0$) such that $s_\alpha \sqsubseteq^\top s_u$; (2) if s is β-unexpected, we have $s_a \cdot s_u \cdot s_c = s$ ($|s_a|, |s_c| \geq 0$) such that $s_\alpha \sqsubseteq^\top s_u$ and $s_\beta \sqsubseteq_\perp s_u$; (3) if s is γ-unexpected, we have $s_a \cdot s_u \cdot s_c = s$ ($|s_a|, |s_c| \geq 0$) such that $s_\alpha \sqsubseteq^\top s_u$ and $s_\gamma \sqsubseteq_\perp s_u$.*

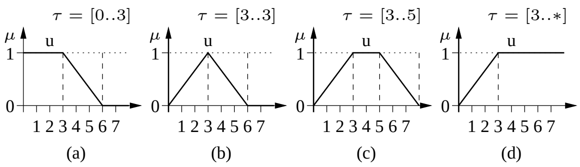

Figure 3: Fuzzy measure of the "certainly" set for γ-unexpectedness.

The composition of an unexpected sequence can therefore be considered as at most three maximal subsequences, called the *antecedent sequence* (denoted as s_a, and $|s_a| \geq 0$), the bordered unexpected sequence (denoted as s_u, and $|s_u| > 0$) and the *consequent sequence* (denoted as s_c, and $|s_c| \geq 0$).

Example 3. Let us consider a belief $b = [\langle 11\rangle; \langle 21\rangle; \langle 31\rangle; [0..2]]$ on sequence of events, where the numbers $11, 21, 31, \ldots$ stand for event IDs. The above belief b requires that the event 11 must be followed by the event 21, but not of the event 31, within no more than two intervals. Thus the event sequence $s = \langle(12)(22)(12)(11)(12)(11)(12)(21)(31)(12)\rangle$ is β-unexpected to the belief b. The antecedent sequence, the bordered unexpected sequence and the consequent sequence of the sequence s are shown in Figure 4. □

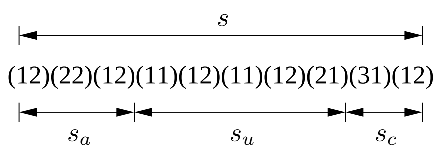

Figure 4: The composition of an unexpected sequence.

Given a belief b and set S of sequences that support an unexpectedness $u \vdash b$, that is, for each $s \in S$ we have $s \models (u \vdash b, U)$. Let S_a be the set of all antecedent sequences, S_u be the set of all bordered unexpected sequences and S_c be the set of all consequent sequences. By studying S_a, S_u and S_c, for example, by performing the sequential pattern mining to them, we can further discover the implication rules on such unexpected behaviors, such as, the maximal frequent sequences in S_a reflect the implications of the unexpectedness $u \vdash b$, and the unexpectedness $u \vdash b$ implies the consequences depicted by the maximal frequent sequences in S_c. All the same, the maximal frequent sequences in S_u depict the internal structures with the unexpectedness $u \vdash b$. The discovery of such rules and structures is out of the scope of this paper and is detailed in our previous article [6].

3.4 The Algorithm Taufu

Our algorithm Taufu finds all fuzzy unexpected sequences from a sequence database $\mathcal{D}$, with respect to the belief base $\mathcal{B}$, the fuzzy sets $\mathcal{F}$ and the minimum membership degree ω. The output of Taufu includes all fuzzy unexpected sequences $\langle s, U \rangle$ associated with the membership degree U, and the bordered unexpected sequence s_u, the antecedent sequence s_a and the consequent sequence s_c of each pair $\langle s, U \rangle$. Algorithm 1 shows the main routine of the algorithm Taufu.

Algorithm 1: The algorithm Taufu.

Input : $\mathcal{D}$, $\mathcal{B}$, $\mathcal{F}$, ω
Output: All $\langle s, U \rangle$, s_u, s_a and s_c

foreach $s \in \mathcal{D}$ **do**
  foreach $s_\alpha \in \mathcal{B}$ **do**
    if $s_\alpha \sqsubseteq s$ **then**
      foreach b *contains* s_α **do**
        if $\alpha \vdash b$ **then**
          $uxps\_alpha(s, b, \mathcal{B})$;
          $uxps\_crisp(s, b.s_\alpha, b.s_\gamma)$;
          continue;
        end
        $uxps\_fuzzy(s, b, \mathcal{B}, \mathcal{F}, \omega)$;
      end
    end
  end
end

The belief base $\mathcal{B}$ is indexed by the sequence s_α contained in each beliefs, so that for each sequence s contained in the sequence database $\mathcal{D}$, and for each s_α indexed in $\mathcal{B}$, the algorithm first verifies whether $s_\alpha \sqsubseteq s$. If $s_\alpha \sqsubseteq s$, then for each belief $b \in \mathcal{B}$ associated with s_α, the algorithm first finds α-unexpectedness from s by the subroutine *uxps_alpha* and finds γ-unexpectedness from s by the subroutine *uxps_crisp* if b states the α-unexpectedness; then finds fuzzy β- or γ-unexpected from s by the subroutine *uxps_fuzzy* if b does not state α-unexpectedness.

Algorithm 2 shows the procedure *uxps_alpha*. Note that in order to maintain the consistence of the belief base $\mathcal{B}$, only the sequences violating all of the beliefs that state an α-unexpectedness and contain the same s_α are considered as α-unexpected to $\mathcal{B}$, see Example 4. Therefore the procedure *uxps_alpha* outputs $\langle s, 1 \rangle$ if s is α-unexpected to each belief that states the α-unexpectedness with s_α.

Algorithm 2: Subroutine *uxps_alpha*.

Input : s, b, $\mathcal{B}$
Output: $\langle s, 1 \rangle$ if s is α-unexpected, s_u, s_a and s_c

foreach b' *associated with* $b.s_\alpha$ **do**
  if $b'.s_\alpha \cdot b'.s_\beta \sqsubseteq s$ **then**
    return;
  end
  if $b.s_\alpha \cdot b.s_\beta \not\sqsubseteq s$ **then**
    generate s_u and s_a from s;
    $s_c = \emptyset$;
    $output(\langle s, 1 \rangle, s_u, s_a, s_c)$;
  end
end

Example 4. Given a belief base consists in two beliefs $b_1 = [\langle(11)\rangle; \langle(21)\rangle; \langle(31)\rangle; *]$ and $b_2 = [\langle(11)\rangle; \langle(22)\rangle; \emptyset; *]$, the sequence $s_1 = \langle(11)(22)\rangle$ is α-unexpected to b_1 but not to b_2; the sequence $s_2 = \langle(11)(21)\rangle$ is α-unexpected to b_2 but not to b_1; the sequence $s_3 = \langle(11)(12)\rangle$ is α-unexpected to both of b_1 and b_2; the sequence $s_4 = \langle(11)(31)\rangle$ is γ-unexpected to both of b_1 and b_2. Our algorithm outputs s_3 as an α-unexpected sequence for b_1 and b_2; outputs s_4 as a γ-unexpected sequence for b_1 and b_2 with membership degree $U = 1$. □

The procedure *uxps_crisp* simply verifies whether $b.s_\alpha \cdot b.s_\gamma \sqsubseteq s$ and outputs the result sequences.

The procedure *uxps_fuzzy* is shown in Algorithm 3, which is detailed in Example 5.

Example 5. As detailed in Figure 5, to illustrate Algorithm 3, let the input sequence s be $\langle(11)(11)(12)(21)(12)(22)(21)(22)(21)(12)\rangle$, and $[\langle(11)(12)\rangle; \langle(21)(22)\rangle; \langle(31)\rangle; [1..3]]$ be the belief b, where the numbers stand for event IDs. We have two fuzzy sets "almost" (labeled as A) and "certainly" (labeled as C) for the partitions of β-unexpectedness of belief b, shown in Figure 5(b). The part

Algorithm 3: Subroutine *uxps_fuzzy*.

Input : s, b, $\mathcal{F}$, ω
Output: $\langle s, 1\rangle$ if s is β-unexpected or γ-unexpected, s_u, s_a and s_c

$L = get\_labels(\mathcal{F}, b)$;
foreach $s_b \in \{b.s_\beta, b.s_\gamma\}$ **do**
  foreach $l \in L$ **do**
    $l : X = find\_fuzzy\_bounds((b.s_\alpha)_\perp, (s_b)^\top, \mathcal{F}, b, \omega)$;
  end
  foreach $l \in L$ **do**
    foreach $x \in l : X$ **do**
      if $occu = backward(s, x)$ **then**
        if $occu' = forward(s, x)$ **then**
          generate s_u, s_a, s_c from s, $occu$ and $occu'$;
          $output(\langle s, x.degree\rangle, s_u, s_a, s_c)$;
          break;
        end
      end
    end
  end
end

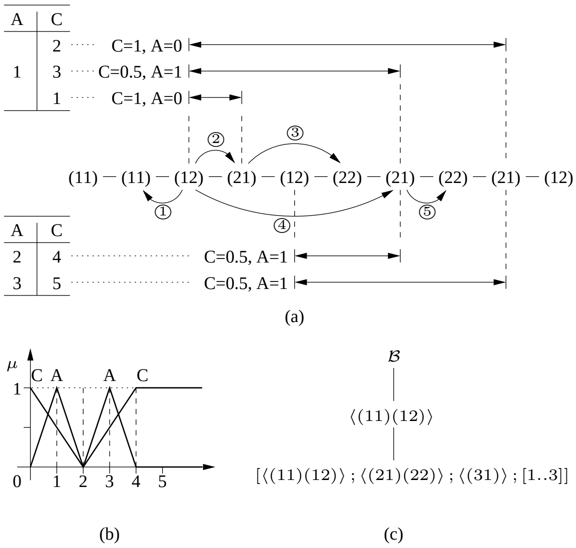

Figure 5: Illustration of a fuzzy β-unexpected sequence extraction.

of the belief base containing the belief b is shown as Figure 5(c).

The procedure *uxps_fuzzy* first finds all the labels of fuzzy partitions corresponding to b in the fuzzy set $\mathcal{F}$, in this example we have $L = \{A, C\}$. The procedure then finds fuzzy unexpectedness with respect to s_β and s_γ of the belief b. In this example, we only illustrate how *uxps_fuzzy* extracts the β-unexpectedness from s. The extraction of the γ-unexpectedness is the same one.

The subroutine $find\_fuzzy\_bounds$ of the procedure *uxps_fuzzy* finds all intervals of itemsets between the last itemset of s_α and the first itemset of s_β (or s_γ) with respect to the fuzzy partitions and the minimum membership degree ω. The result returned by $find\_fuzzy\_bounds$ is a set of intervals and begin-end positions categorized by the label of fuzzy partitions corresponding to current belief b, and the ranges are sorted with the descendant membership degree order. In this example, the algorithm finds all fuzzy bounds between (12) and (21). Within the sequence s, there are totally 3 intervals for the fuzzy partition A and 5 intervals for the fuzzy partition C, the order is shown as the two tables in Figure 5(a).

For each fuzzy partition, the algorithm finds the unexpectedness by the backward matching procedure *backward*, that finds $s_\alpha \sqsubseteq s$, and the forward matching procedure $forward$, that finds $s_\beta \sqsubseteq s$ (or $s_\gamma \sqsubseteq s$).

In this example, for both of the fuzzy partitions A and C, the $forward$ procedure finds the first sequence that contains s_α, that is $\sqsubseteq$ (11)(12) shown as ① in Figure 5(a). For the fuzzy partition A, the first (the best) interval between (12) and (21) is shown as ④, with which the algorithm finds an occurrence of s_β, as shown as ⑤. For the fuzzy partition C, the first (the best) interval between (12) and (21) is shown as ②, with which the algorithm finds an occurrence of s_β, as shown as ③. Imagine that if the first interval between (12) and (21) does not drive the s_β, then the second one will be verified, and till to the last one; if no s_β is found, the algorithm returns without output.

Therefore, finally the algorithm outputs the bordered unexpected sequence $\langle(11)(12)(21)(12)(22)(21)(22)\rangle$ (① ④ ⑤) for the fuzzy partition A; outputs the bordered unexpected sequence $\langle(11)(12)(21)(12)(22)\rangle$ (① ② ③) for the fuzzy partition C. □

As depicted in the above instance, our algorithm will minimize the length of the bordered unexpected sequences.

4 Experiments

To evaluate our approach Taufu, we perform a group of experiments to extract unexpected sequences from the access log of a security testing Web server, where a large number of attacks are logged. The sequence database converted from the access log contains 67,228 session sequences corresponding to 27,552 distinct items.

Totally 4 groups of 20 beliefs corresponding to 4 categories of occurrence constraints are considered in our experiments: `CAT1` stands for 5 beliefs with $\tau = [0..*]$; `CAT2` stands for 5 beliefs with $\tau = [0..X]$ where $X \geq 0$ is an integer; `CAT3` stands for 5 beliefs with $\tau = [Y..*]$ where $Y > 0$ is an integer; and `CAT4` stands for 5 beliefs with $\tau = [X..Y]$ where $Y \geq X > 0$ are two integers.

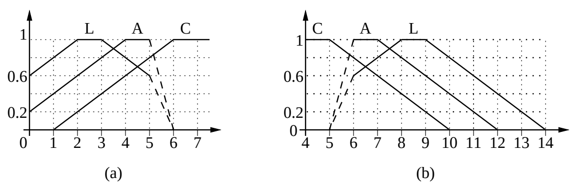

Figure 6: (a) β-unexpected fuzzy partitions. (b) γ-unexpected fuzzy partitions.

To simplify the procedure of our experiments, the ratio of membership function μ is fixed to ± 0.2 for all fuzzy partitions, further more, the partitions "almost" and "a little" do not cover the interval ranges within which the membership degree of the partition "certainly" is 1. The interval value of the partitions "almost" and "a little" where there membership degree equals 1 is fixed to 2.

For instance, for the following belief of `CAT2` $[\langle(login)\rangle\,;\langle(list)(view)\rangle\,;\langle(logout)\rangle\,;[0..5]]$, the fuzzy partitions are shown in Figure 6. Note that the partitions "almost" and "a little" are partial. The numbers of unexpected sequences that we find with respect to $\omega = 1$, $\omega = 0.7$ and $\omega = 0.2$ are listed in Table 1 (β-unexpected/γ-unexpected).

	$\omega = 1$	$\omega = 0.7$	$\omega = 0.2$
Certainly	47/22	49/23	55/25
Almost	4/2	7/2	10/6
A **L**ittle	4/1	5/5	6/12

Table 1: β-unexpected/γ-unexpected sequences extracted from a belief in `CAT2`.

Figure 7 shows the number of unexpected sequences in the fuzzy sets "certainly unexpected", "almost unexpected" and "a little unexpected" when the minimum fuzzy degree $\omega = 1.0$. Figure 8 and Figure 9 show the number of unexpected sequences in the same fuzzy sets when $\omega = 0.7$ and $\omega = 0.2$.

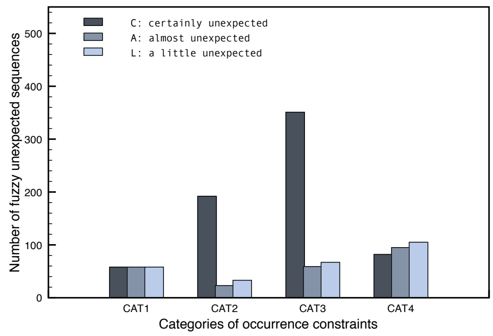

Figure 7: Minimum fuzzy degree $\omega = 1.0$.

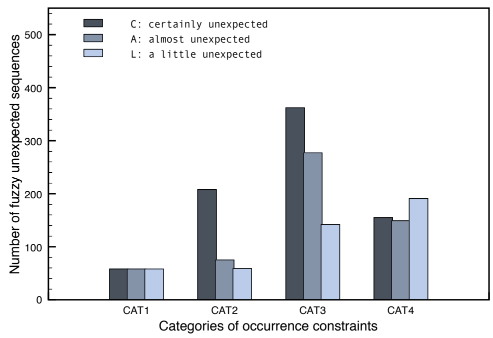

Figure 8: Minimum fuzzy degree $\omega = 0.7$.

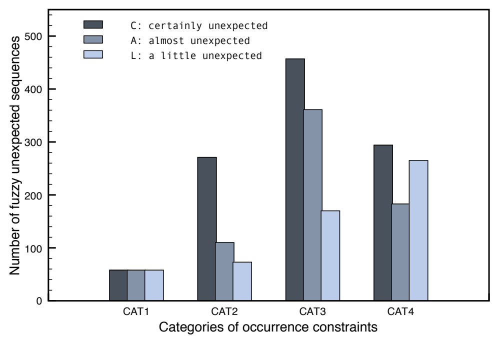

Figure 9: Minimum fuzzy degree $\omega = 0.2$.

Such unexpected sequences are difficult to discovered by classical sequential pattern algorithms because of the low support and the inclusion of sequences, and even because of the classification of those fuzzy partitions. As shown in this section, in our testing sequence database of Web attacks, some kind of beliefs drive a clear view of the unexpetedness, for example `CAT2` and `CAT3`, but the unexpectedness stated by the beliefs of `CAT4` are quite "fuzzy". Hence in the case of `CAT4`, the unexpected sequences extracted by a fuzzy method is more important for post analysis, and even for improving the belief base. On the other hand, even in such a database, the sequential pattern $\langle(login)(logout)\rangle$ can be discovered with a minimum support less than 0.1, but such a sequential pattern cannot state any unexpectedness contained in the database.

5 Conclusion

In this paper we introduce a novel approach for the discovery of fuzzy unexpected sequences from databases, with respect to user defined beliefs. We also present the algorithm Taufu, which has been verified with real Web server log file analyzing. The experimental results show that our approach Taufu extracts the unexpected sequences corresponding to all predefined fuzzy partitions.

We are interested in discovering belief driven *fuzzy unexpected sequential patterns* and *fuzzy unexpected sequential rules* from database, that are helpful to extract the internal relations within the unexpectedness and to find the implications before/after the occurrences of unexpectedness, where a fuzzy method can be creditable.

References

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE*, pages 3–14, 1995.

[2] R.-S. Chen and Y.-C. Hu. A novel method for discovering fuzzy sequential patterns using the simple fuzzy partition method. *JASIST*, 54(7):660–670, 2003.

[3] R.-S. Chen, G.-H. Tzeng, C. C. Chen, and Y.-C. Hu. Discovery of fuzzy sequential patterns for fuzzy partitions in quantitative attributes. In *AICCSA*, pages 144–150, 2001.

[4] Y.-L. Chen and T. C. K. Huang. A new approach for discovering fuzzy quantitative sequential patterns in sequence databases. *Fuzzy Sets and Systems*, 157(12):1641–1661, 2006.

[5] Y.-C. Hu, R.-S. Chen, G.-H. Tzeng, and J.-H. Shieh. A fuzzy data mining algorithm for finding sequential patterns. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 11(2):173–194, 2003.

[6] D. H. Li, A. Laurent, and P. Poncelet. Mining unexpected sequential patterns and rules. Technical Report RR-07027 (2007), Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier, 2007.

[7] K. McGarry. A survey of interestingness measures for knowledge discovery. *Knowl. Eng. Rev.*, 20(1):39–61, 2005.

[8] B. Padmanabhan and A. Tuzhilin. On characterization and discovery of minimal unexpected patterns in rule discovery. *IEEE Trans. Knowl. Data Eng.*, 18(2):202–216, 2006.

[9] A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *KDD*, pages 275–281, 1995.

[10] M. Spiliopoulou. Managing interesting rules in sequence mining. In *PKDD*, pages 554–560, 1999.

[11] R. B. V. Subramanyam and A. Goswami. A fuzzy data mining algorithm for incremental mining of quantitative sequential patterns. *International Journal of Uncertainty Fuzziness Knowledge-Based Systems*, 13(6):633–652, 2005.