Recognizing Unexpected Recurrence Behaviors with Fuzzy Measures in Sequence Databases

Dong (Haoyuan) Li
LGI2P, École des Mines d'Alès
Parc scientifique G. Besse
30035 Nîmes, France
Haoyuan.Li@ema.fr

Anne Laurent
LIRMM, Univ. Montpellier 2
161 rue Ada
34392 Montpellier, France
laurent@lirmm.fr

Pascal Poncelet
LGI2P, École des Mines d'Alès
Parc scientifique G. Besse
30035 Nîmes, France
Pascal.Poncelet@ema.fr

ABSTRACT

The recognition of unexpected behaviors in databases is an important problem in many real-world applications. In the previous studies, the unexpectedness is mainly stated within the context of the most-studied patterns, association rules, or sequential patterns. In this paper, we first propose the notion of fuzzy recurrence rule, a new kind of rule-based behavior in sequence databases, and then we introduce the problem of recognizing unexpected sequences contradicting the beliefs on fuzzy recurrence rules, with fuzzy measures. We also develop, UFR, an algorithm for discovering unexpected recurrence behaviors in a sequence database. Our approach is evaluated with Web access log data.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*; I.5.1 [**Pattern Recognition**]: Models—*Fuzzy set*

General Terms

Algorithms, Management, Theory

Keywords

Unexpected recurrence behavior, fuzzy recurrence rule, fuzzy measure, sequence data mining

1. INTRODUCTION

During the past years, as the most important topics of data mining, association rules [1] (corresponding to frequent patterns) and sequential patterns [2] have received much attention, such as the work addressed in [11, 5, 12] and [26, 18, 28, 3, 27, 22].

The association rule mining finds the frequent behaviors of the correlation between sets of attributes (also called *patterns*), as the rules in form "if X then Y" where X and Y are two patterns. An association rule can reflect the information typically like "60% of customers who purchase Coca Cola also purchase potato chips (if *Coca Cola* then *potato chips*)". Different from mining association rules, the purpose of the sequential pattern mining is to find frequent behaviors of sequential data, thus a sequential pattern is a sequence such that "A then B then C then ...", where A, B, C, ... are patterns. A sequential pattern can help interpreting the information typically like "60% of people purchase beers, then purchase Sci-Fi movies, and then purchase rock music".

On the other hand, the discoveries (normally belief-driven [23]) of unexpected behaviors contradicting the knowledge on known (normally frequent or predefined) behaviors in databases, becomes more and more interesting for many real-word applications. Since frequent patterns and sequential patterns are the most-studied behaviors in databases, in the existing studies of discovering unexpected behaviors, the unexpectedness is mainly stated within the context of patterns [4, 19], association rules [21], or sequential patterns [24, 16].

We proposed a semantics based framework of unexpected sequence mining in our previous work [16]. For instance, according to the behavior "people purchase Sci-Fi movies, and then purchase rock music", the behavior "people purchase Sci-Fi movies, and then purchase classical music" can be considered as unexpected since the classical music can be considered as semantically opposite to the rock music. That work has been extended with fuzzy methods in [17].

In this paper, we are interested in the unexpectedness stated by *fuzzy recurrence rule*, a new kind of rule-based behavior in sequence databases. The fuzzy recurrence rules are in the form "if the sequence s_α repeatedly occurs, then the sequence s_β repeatedly occurs". For instance, a fuzzy recurrence rule can be "60% of customers who *often* purchase Sci-Fi books then Sci-Fi movies later, also purchase PC games *often*". This type of rules reflect the associated correlations between repeatedly occurred elements in sequential data. The unexpectedness on recurrence behaviors is determined by the domain-expert-defined semantic oppositions. For instance, if we consider that the classical music is semantically opposite to PC games, then the fact "1% customers who *often* purchase Sci-Fi books then Sci-Fi movies later, *often* purchase classical music" stands for an unexpected recurrence behavior in a customer transaction database.

Such unexpected recurrence behaviors can be interesting for many application domains, including marketing analysis, finance fraud detection, DNA segment analysis, Web content personalization, network intrusion detection, weather

CSTST 2008 October 27-31, 2008, Cergy-Pontoise, France

prediction, and so on.

The rest of the paper is organized as follows. In Section 2, we introduce the related work. Section 3 presents our proposals of fuzzy recurrence rules and unexpected recurrence behaviors. In Section 4, we develop an effective algorithm UFR for discovering unexpected recurrence behaviors in a sequence database. Finally, we conclude in Section 5.

2. RELATED WORK

In order to find more relevant behaviors, the fuzzy set theory has been well adopted for treating the quantitative attributes [25] in databases, such as the work concentrated in [15, 7, 14, 13, 6, 8].

For instance, in the discovery of fuzzy association rules defined in [15], with the usage of fuzzy sets, the form of discovered rules becomes "if X is A then Y is B", where A and B are fuzzy sets that describe the properties of patterns X and Y, such as "60% of customers who purchase a lot of Coca Cola also purchase a lot of potato chips (if *Coca Cola* is *lot* then *potato chips* is *lot*)". In the same manner that the fuzzy association rule is defined, the notion of fuzzy sequential patterns proposed in [7] considers the sequential patterns on quantitative attributes like "60% of people purchase a lot of beers, then purchase many Sci-Fi movies later, then purchase few PC games", where the sequence is "*beer* is *lot*, then *Sci-Fi movie* is *many*, and then *PC game* is *few*".

Although most of the existing approaches for mining fuzzy association rules or fuzzy sequential patterns concentrate on developing efficient algorithms, the fuzzy sets are also considered on imprecise data (like [10, 9], but that topic is not covered in this paper). In our approach, we consider the binary-valued attributes in databases as other crisp data mining approaches, however we use fuzzy sets for describing the recurrence behaviors of data, instead of the quantitative attributes.

Unexpected behaviors are generally considered within the framework of subjective interestingness measure. In [23], the notion of unexpectedness is addressed with hard belief and soft belief. A hard belief is a belief that can never be changed by new evidences in data,and any contradiction of such a belief implies data error. A soft belief corresponds to the constraints on data measured by a degree, which can be modified with new evidences contradicting this belief. The interestingness of such new evidences is measured by the change of the degree.

With the unexpectedness measure, a belief-driven approach for finding unexpected patterns and association rules is proposed in [19, 20, 21]. In that approach, a belief is given from association rule, and the unexpectedness is stated by semantic opposition 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 $A \cup X \rightarrow B$ hold in the data; (3) the support and confidence of the rule $A \cup X \rightarrow Y$ do not hold in the data. The discovery process is performed within the framework of the *a priori* algorithm.

[24] 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 from our problem in the measure and the notion of unexpectedness contained in data.

In our recent work [17], we proposed a belief-driven approach for recognizing fuzzy unexpected sequences corresponding to sequential implication rules. A *sequential implication rule* is a rule of the form "if the sequence s_α occurs then the sequence s_β occurs latter" so that the beliefs are created with respect to (1) the distance between s_α and s_β; (2) the semantics of the implication between s_α and s_β, i.e., s_β cannot be replaced by another sequence s_γ. The fuzzy sets are considered on the distance between the two sequences.

3. FUZZY RECURRENCE RULES AND UNEXPECTED BEHAVIORS

In this section, we first formalize the fuzzy recurrence rules within the common framework of sequence mining model, we then present the belief base on such fuzzy recurrence rules, with which the unexpected recurrence behaviors are therefore proposed and discovered.

3.1 Data Model

We consider the sequential data that consist in binary-valued attributes. Given a limited numbered set of distinct binary-valued attributes $R = \{i_1, i_2, \ldots, i_n\}$, each attribute is an *item*. An *itemset* is an unordered collection of items, denoted as $I = \{i_1, i_2, \ldots, i_m\}$, such that $I \subseteq R$. A *sequence* is an ordered list of itemsets, denoted as $s = I_1 I_2 \cdots I_k$. A *sequence database* is usually a large set of sequences, denoted as D.

Given two sequences $s = I_1 I_2 \ldots I_m$ and $s' = I'_1 I'_2 \ldots I'_n$, if there exist integers $1 \leq i_1 < i_2 < \ldots < i_m \leq n$ such that $I_1 \subseteq I'_{i_1}, I_2 \subseteq I'_{i_2}, \ldots, I_m \subseteq I'_{i_m}$, then the sequence s is a *subsequence* of the sequence s', denoted as $s \sqsubseteq s'$. If $s \sqsubseteq s'$, we say that s *is contained in* s', or s' *supports* s. For example, the sequence $s_1 = (a)(b)$ is contained in the sequence $s_2 = (a)(b)(c)$, but not contained in the sequence $s_3 = (ab)(c)$.

Given a sequence database D, the *support* or *frequency* of a sequence s, denoted as $\sigma(s, D)$, is the fraction of the total number of sequences in D that support s. Given a user specified threshold of support called *minimum support*, denoted as *min_supp*, a sequence s is *frequent* if $\sigma(s, D) \geq min\_supp$.

In addition, we denote the *concatenation* of n sequences as $s_1 s_2 \cdots s_n$. For example, let $s_1 = (a)(b)$ and $s_2 = (c)(d)$, then we have $s_1 s_1 = (a)(b)(a)(b)$ or $s_1 s_2 = (a)(b)(c)(d)$.

3.2 Fuzzy Recurrence Rules

A *recurrence rule* is a rule on sequences with form $\langle s_\alpha, \psi \rangle \Rightarrow \langle s_\beta, \theta \rangle$, where s_α and s_β are two sequences, and ψ, θ are two integers for describing the recurrence behaviors.

The recurrence rule indicates that given a sequence s, if s_α is orderly occurred no less than ψ times within s, then s_β should occurs in s no less than θ times, that is,

$$(\underbrace{s_\alpha \cdots s_\alpha}_{n} \sqsubseteq s) \wedge (n \geq \psi) \Longrightarrow (\underbrace{s_\beta \cdots s_\beta}_{k} \sqsubseteq s) \wedge (k \geq \theta).$$

We call the form $\langle s, \psi \rangle$ a *recurrent sequence*, and we have

$$\langle s, \psi \rangle \sqsubseteq s' \iff (\underbrace{s \cdots s}_{n} \sqsubseteq s') \wedge (n \geq \psi),$$

for that the sequence s' *supports* the recurrent sequence $\langle s, \psi \rangle$. We call the recurrent sequence $\langle s, \psi \rangle$ a ψ-recurrence sequence. We use the wildcard "$*$" for denoting the general-purposed meaning of the support between sequences, that is,

$$\langle s, * \rangle \sqsubseteq s' \equiv s \sqsubseteq s'.$$

Given a sequence s and a recurrence rule $r = \langle s_\alpha, \psi \rangle \Rightarrow \langle s_\beta, \theta \rangle$, if $\langle s_\alpha, \psi \rangle \sqsubseteq s$ and $\langle s_\beta, \theta \rangle \sqsubseteq s$, then we say that s *supports* r, denoted as $s \models r$. For instance, the recurrence rule $r = \langle (a)(b), 3 \rangle \Rightarrow \langle (c)(d), * \rangle$ depicts that given a sequence s, if $(a)(b)$ is contained repeatedly in s no less 3 times, then $(c)(d)$ should be contained in s; in other words, if $(a)(b)(a)(b)(a)(b) \sqsubseteq s$, then $(c)(d) \sqsubseteq s$. Notice that the occurrences of s_α must be ordered, that is, for example, given a rule $r_1 = \langle (a)(b), 2 \rangle \Rightarrow \langle (c), * \rangle$, the sequence $s_1 = (a)(a)(c)(b)(b)$ does not support r_1, but the sequence $s_2 = (a)(b)(c)(a)(b)$ supports r_1; however, the sequence s_1 supports the rules $r_2 = \langle (a), 2 \rangle \Rightarrow \langle (c), * \rangle$ and $r_3 = \langle (b), 2 \rangle \Rightarrow \langle (c), * \rangle$.

Considering the integer ψ, a human-friendly interpretation is more flexible in most applications. For instance, in market basket analysis, to point out that "the customers who often purchase Sci-Fi books often purchase action movies" is more relevant than the conclusion "the customers who purchase at least 7 times of Sci-Fi books purchase at least 5 times of action movies".

We therefore extend the recurrence rule with fuzzy sets, so called the *fuzzy recurrence rule*, in the form $\langle s_\alpha, \omega_\alpha \rangle \Rightarrow \langle s_\beta, \omega_\beta \rangle$, where ω_α and ω_β are two fuzzy sets for describing s_α and s_β. The sequences $\langle s_\alpha, \omega_\alpha \rangle$ and $\langle s_\beta, \omega_\beta \rangle$ are *fuzzy recurrent sequences*. Given a sequence s' and a fuzzy recurrent rule $\langle s, \omega \rangle$, that s' *supports* $\langle s, \omega \rangle$ is defined as

$$\langle s, \omega \rangle \sqsubseteq s' \iff (\underbrace{s \cdots s}_{n} \sqsubseteq s) \wedge (\mu_\omega(n) \geq \zeta),$$

where the fuzzy degree measured by the membership function $\mu_\omega(n)$ must be superior or equal to a threshold ζ.

Let us consider the following example.

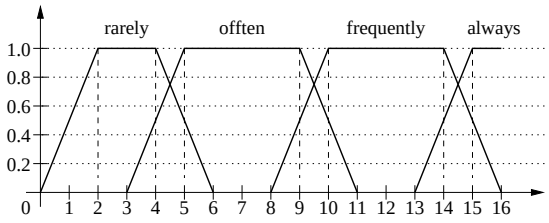

Figure 1: Fuzzy partitions for recurrence rules.

Example 1. Given a set of distinct events $A, B, C, D, \ldots$, an ordered of events can be represented as the data model of sequence. Assuming that given an event sequence s, if s supports the recurrent sequence $\langle (A)(B), 4 \rangle$, then s supports the subsequence $(C)(D)$; if s supports the recurrent sequence $\langle (A)(B), 9 \rangle$, then s supports (C). These behaviors can be described by recurrence rules, such as the rule $r_1 = \langle (A)(B), 4 \rangle \Rightarrow \langle (C)(D), * \rangle$ and the rule $r_2 = \langle (A)(B), 9 \rangle \Rightarrow \langle (C), * \rangle$. Given a sequence s_1 such that $\langle (A)(B), 3 \rangle \sqsubseteq s_1$ and $(C)(D) \sqsubseteq s_1$, a sequence s_2 such that $\langle (A)(B), 8 \rangle \sqsubseteq s_2$ and $(C) \sqsubseteq s_2$, we have $s_1 \not\models r_1$ and $s_2 \not\models r_2$. However, since the recurrent sequences contained in these sequences and rules are close, the sequences s_1 and s_2 can be still potentially interesting. On the other hand, considering the fuzzy recurrence rules $r'_1 = \langle (A)(B), rarely \rangle \Rightarrow \langle (C)(D), * \rangle$ and $r'_2 = \langle (A)(B), often \rangle \Rightarrow \langle (C), * \rangle$, corresponding to the rules r_1 and r_2 with respect to the fuzzy partitions shown in Figure 1, let the threshold $\zeta = 0.5$, then we have $s_1 \models r'_1$ and $s_2 \models r'_2$. We can further define more partitions, such as "always" or "rarely". □

In this paper, the fuzzy recurrence rules are considered as having been predefined by domain experts, the discovery of fuzzy recurrence rules will be covered in our future research work.

3.3 Unexpected Recurrence Behaviors

We are considering to discover the sequences contained in a database those are semantically opposite to a given set of fuzzy recurrence rules. In order to find such sequences, we propose an approach that constructs a belief base from given fuzzy recurrence rules with semantic constraints, so that each sequence not respecting the belief base is unexpected, from which the unexpected recurrence behaviors can be further discovered.

A *belief on recurrence behaviors* is a set of constraints that consists of a fuzzy recurrence rule $\langle s_\alpha, \omega_\alpha \rangle \Rightarrow \langle s_\beta, \omega_\beta \rangle$ and a semantic constraint $\langle s_\beta, \omega_\beta \rangle \not\approx_{sem} \langle s_\gamma, \omega_\gamma \rangle$, where ω_γ is a fuzzy set for the sequence s_γ. The fuzzy recurrence rule implies an association relation between $\langle s_\alpha, \omega_\alpha \rangle$ and $\langle s_\beta, \omega_\beta \rangle$, i.e., if the recurrence of s_α is ω_α, then the recurrence of s_β is ω_β. The semantic constraint implies that the recurrent sequences $\langle s_\beta, \omega_\beta \rangle$ and $\langle s_\gamma, \omega_\gamma \rangle$ are semantically opposite to each other. Notice that s_β and s_γ are not necessary to be different: $\langle (game), rarely \rangle$ and $\langle (game), always \rangle$ are semantically opposite to each other.

We use the triple $[\langle s_\alpha, \omega_\alpha \rangle ; \langle s_\beta, \omega_\beta \rangle ; \langle s_\gamma, \omega_\gamma \rangle]$ for denoting a belief, that constrains: given a sequence s, if s supports $\langle s_\alpha, \omega_\alpha \rangle$, then s supports $\langle s_\beta, \omega_\beta \rangle$, however s should not support $\langle s_\gamma, \omega_\gamma \rangle$, since $\langle s_\gamma, \omega_\gamma \rangle$ semantically be opposite to $\langle s_\beta, \omega_\beta \rangle$, that is,

$$(\langle s_\alpha, \omega_\alpha \rangle \sqsubseteq s) \wedge (\langle s_\beta, \omega_\beta \rangle \sqsubseteq s) \wedge (\langle s_\gamma, \omega_\gamma \rangle \not\sqsubseteq s).$$

Example 2. Let us consider the instance in Section 1. We know that the customers who purchase Sci-Fi books (noted as *book*) then Sci-Fi movies (noted as *movie*) latter like to play PC games (noted as *game*). Since we consider that PC games and classical music (noted as *music*) are semantically opposite to each other, the semantic constraint can be $\langle (game), often \rangle \not\approx_{sem} \langle (music), often \rangle$. Thus the belief can be written as

$$[\langle (book)(movie), often \rangle ; \langle (game), often \rangle ; \langle (music), often \rangle]$$

if we assume such customers will not often purchase classical music. The fuzzy sets for the purchase of classical music can also be that shown in Figure 1. The above belief describes that the customers who often purchase Sci-Fi books and then Sci-Fi movies, purchase PC games often, however do not purchase classical music often. □

If a sequence s satisfies these constraints, we say that the sequence s *supports* the belief $[\langle s_\alpha, \omega_\alpha\rangle ; \langle s_\beta, \omega_\beta\rangle ; \langle s_\gamma, \omega_\gamma\rangle]$, denoted as $s \models b$. A sequence s is *unexpected* if s violates a belief b, denoted as $s \not\models b$. Two cases of violation are considered in our approach according to two factors of the unexpectedness contained in the sequence s.

Definition 1. (β-unexpected sequence) Given a belief $b = [\langle s_\alpha, \omega_\alpha\rangle ; \langle s_\beta, \omega_\beta\rangle ; \langle s_\gamma, \omega_\gamma\rangle]$, a sequence s is *β-unexpected*, denoted as $s \not\models_\beta b$, if s supports $\langle s_\alpha, \omega_\alpha\rangle$ and s_β but does not support $\langle s_\beta, \omega_\beta\rangle$, that is,

$$s \not\models_\beta b \iff (\langle s_\alpha, \omega_\alpha\rangle \sqsubseteq s) \wedge (s_\beta \sqsubseteq s) \wedge (\langle s_\beta, \omega_\beta\rangle \sqsubseteq s),$$

and such a behavior is called β-unexpectedness. □

The primary factor of the β-unexpectedness in a sequence s is that the recurrent sequence $\langle s_\beta, \omega_\beta\rangle$ does not occur as expected however at least the sequence s_β occurs in s. For instance, considering the belief in Example 2, noted as b, let s be a customer transaction sequence, if we have $\langle (book)(movie), often\rangle \sqsubseteq s$ and $\langle (game), often\rangle \sqsubseteq s$, then the sequence s is expected in the meaning of fuzzy recurrence rule $\langle (book)(movie), often\rangle \Rightarrow \langle (game), often\rangle$ (we discuss the semantic constraint latter); however, if we have $(game) \sqsubseteq s$ but not $\langle (game), often\rangle \sqsubseteq s$, for example, we have $\langle (game), rarely\rangle \sqsubseteq s$ instead of $\langle (game), often\rangle \sqsubseteq s$, since $\langle (game), rarely\rangle \sqsubseteq s$ implies $(game) \sqsubseteq s$, then s is a β-unexpected sequence, i.e., $s \not\models_\beta b$.

Definition 2. (γ-unexpected sequence) Given a belief $b = [\langle s_\alpha, \omega_\alpha\rangle ; \langle s_\beta, \omega_\beta\rangle ; \langle s_\gamma, \omega_\gamma\rangle]$, a sequence s is *γ-unexpected*, denoted as $s \not\models_\gamma b$, if s supports $\langle s_\alpha, \omega_\alpha\rangle$ and $\langle s_\gamma, \omega_\gamma\rangle$, that is,

$$s \not\models_\gamma b \iff (\langle s_\alpha, \omega_\alpha\rangle \sqsubseteq s) \wedge (\langle s_\gamma, \omega_\gamma\rangle \not\sqsubseteq s),$$

and such a behavior is called γ-unexpectedness. □

Respectively, the primary factor of the γ-unexpectedness in a sequence s is that the semantic constraint $\langle s_\beta, \omega_\beta\rangle \not\approx_{sem} \langle s_\gamma, \omega_\gamma\rangle$ is broken, because the recurrent sequence $\langle s_\gamma, \omega_\gamma\rangle$ occurs in s. Considering again the belief b in Example 2 and let s be a customer transaction sequence, if we have $\langle (book)(movie), often\rangle \sqsubseteq s$ and $\langle (music), often\rangle \not\sqsubseteq s$, then the sequence s is not unexpected in the meaning of semantic constraint $\langle (game), often\rangle \not\approx_{sem} \langle (music), often\rangle$; however, if we have $\langle (music), often\rangle \sqsubseteq s$, then s is a γ-unexpected sequence, i.e., $s \not\models_\gamma b$. Of course, it is not necessary to forbid $(music) \sqsubseteq s$, for example, according to this belief, the occurrence of $\langle (music), rarely\rangle$ does not imply the γ-unexpectedness.

In our approach, we consider only the belief bases of coherent beliefs, means that the beliefs contained in the belief base do not contradict each others. Given a belief base B, for any two beliefs $b, b' \in B$, let $b = [\langle s_\alpha, \omega_\alpha\rangle ; \langle s_\beta, \omega_\beta\rangle ; \langle s_\gamma, \omega_\gamma\rangle]$ and $b' = [\langle s'_\alpha, \omega'_\alpha\rangle ; \langle s'_\beta, \omega'_\beta\rangle ; \langle s'_\gamma, \omega'_\gamma\rangle]$, the following constraint must be satisfied:

$$(s_\alpha \sqsubseteq s'_\alpha) \wedge (\omega_\alpha = \omega'_\alpha) \Longrightarrow (s_\beta \not\sqsubseteq s'_\gamma) \vee (\omega_\beta \neq \omega'_\gamma).$$

For example, let us consider two beliefs b_1 and b_2. Let $b_1 = [\langle (a)(b), often\rangle ; \langle (c)(d), often\rangle ; \langle (e)(f), often\rangle]$ and let $b_2 = [\langle (a), often\rangle ; \langle (e), often\rangle ; \langle (c), often\rangle]$, then b_1 and b_2 are in conflict: we have $\langle (a)(b), often\rangle \Rightarrow \langle (c)(d), often\rangle$ and $\langle (a)(b), often\rangle \not\Rightarrow \langle (e)(f), often\rangle$ for b_1, however for b_2 we have $\langle (a), often\rangle \not\Rightarrow \langle (c), often\rangle$. In the rest of this paper, we assume that all beliefs in a belief base are coherent.

Given a sequence database D and a belief base B, the problem of recognizing unexpected recurrence behaviors is to, therefore, find all sequences $s \in D$ that contain β- and/or γ-unexpectedness corresponding to each belief $b \in B$.

4. UFR: MINING UNEXPECTED FUZZY RECURRENCE BEHAVIORS

UFR stands for mining Unexpected Fuzzy Recurrence behaviors. In this section, we detail the algorithm involved in our approach, which is evaluated in performance study.

4.1 Data Structure

The belief base B is constructed as a prefix tree structure with 3 blocks α, β and γ, where in each block two types of edge are used to represent itemsets and sequences, and between blocks a pair of links are used for representing beliefs.

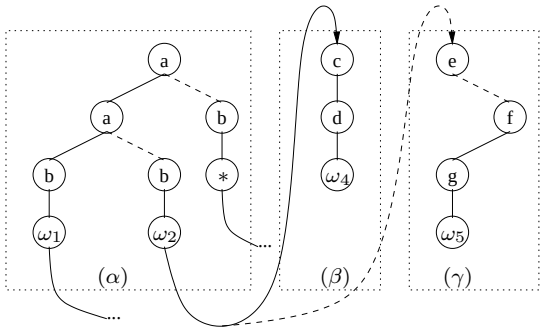

Figure 2: Prefix tree structure of the belief base.

A belief is in the form $[\langle s_\alpha, \omega_\alpha\rangle ; \langle s_\beta, \omega_\beta\rangle ; \langle s_\gamma, \omega_\gamma\rangle]$. In α block, all $\langle s_\alpha, \omega_\alpha\rangle$ parts of each belief are organized as a prefix tree. For example, in Figure 2, block (α) contains 3 recurrent sequences $\langle (a)(a)(b), \omega_1\rangle$, $\langle (a)(ab), \omega_2\rangle$ and $\langle (ab), *\rangle$. All $\langle s_\beta, \omega_\beta\rangle$ parts are contained in β block and all $\langle s_\gamma, \omega_\gamma\rangle$ parts contained in γ block as prefix trees. For example, blocks (β) and (γ) contain recurrent sequences $\langle (c)(d), \omega_4\rangle$ and $\langle (ef)(g), \omega_5\rangle$. The link pair between the 3 blocks shown in Figure 2 represents the belief

$$[\langle (a)(ab), \omega_2\rangle ; \langle (c)(d), \omega_4\rangle ; \langle (ef)(g), \omega_5\rangle] .$$

The verification of each sequence is performed by depth-first traversing the prefix tree in each block.

4.2 Algorithm

The algorithm UFR accepts a belief base B, a sequence database D and a minimum fuzzy degree threshold ζ as input data, and outputs all unexpected sequences in D with respect to B and ζ.

Algorithm 1 shows the verification routine for each sequence $s \in D$. The algorithm first traverses the prefix tree in block α to verify the $\langle s_\alpha, \omega_\alpha\rangle$ part of each belief $b \in B$, with depth-first strategy. If the traverse arrives an ω_α node, the recurrence of sequence s_α is examined within the fuzzy set ω_α and the minimum fuzzy degree ζ. If ζ is satisfied, the link pair is followed to verify the $\langle s_\beta, \omega_\beta\rangle$ part of the same belief b contained in the prefix tree in block β, till to the node ω_β. If the recurrence of s_β does not satisfy ω_β with respect to ζ, then the algorithm outputs the sequence s as

Algorithm 1 Algorithm UFR-seqveri

Input: A sequence s, a belief base B, and a minimum fuzzy degree ζ
Output: The sequence s if unexpected

```
1: for all path p_α ∈ B : α do
2:     ⟨s_α, ω_α⟩ = generate_sequence(p_α);
3:     if seqinc_fuzzy(⟨s_α, ω_α⟩, s, ζ) then
4:         for all path p_β ∈ (p_α → B : β) do
5:             if seqinc(s_β, s) then
6:                 if !seqinc_fuzzy(⟨s_β, ω_β⟩, s, ζ) then
7:                     output s as β-unexpected;
8:                 end if
9:             end if
10:        end for
11:        for all path p_γ ∈ (p_α → B : γ) do
12:            if seqinc(s_γ, s) then
13:                if seqinc_fuzzy(⟨s_γ, ω_γ⟩, s, ζ) then
14:                    output s as γ-unexpected;
15:                end if
16:            end if
17:        end for
18:    end if
19: end for
```

an β-unexpected sequence. The algorithm continue to verify (even if the traverse in block β stops, i.e., s_β not found) the $\langle s_\gamma, \omega_\gamma \rangle$ part of the same belief b with the link, and the prefix tree will be traversed till to the node ω_γ. If the recurrence of s_β satisfies ω_β with respect to ζ, then the algorithm outputs s as an γ-unexpected sequence. The algorithm continues to traverse the prefix tree in block α and repeats the above procedure till to the end of the traverse. All unexpected sequences are therefore discovered.

Corresponding to the fuzzy set in Figure 1 and the belief $[\langle (a)(ab), often \rangle ; \langle (c)(d), rarely \rangle ; \langle (ef)(g), rarely \rangle]$ shown in Figure 2, the UFR-seqveri routine can be illustrated with the sequence s shown in Figure 3, assuming $\zeta = 0.6$.

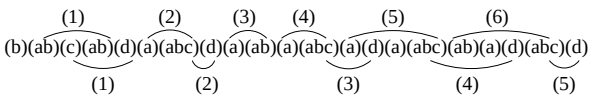

Figure 3: Finding β-unexpected sequence.

We have $\langle (a)(ab), often \rangle \sqsubseteq s$, which is first verified by calling $seqinc\_fuzzy(\langle (a)(ab), often \rangle, s, 0.6)$, and the recurrence of $(a)(ab)$, which is marked as (1) to (6) above the sequence shown in Figure 3, satisfies the minimum fuzzy degree 0.6. Thus $\langle (c)(d), rarely \rangle \sqsubseteq s$ will be verified by calling $seqinc\_fuzzy(\langle (c)(d), rarely \rangle, s, 0.6)$, where the recurrence of $(c)(d)$ is marked as (1) to (5) below the sequence shown in Figure 3. According to Figure 1, we have the fuzzy degree of $\langle (c)(d), 5 \rangle$ is 0.5 in the partition "rarely", so that we have $\langle (c)(d), rarely \rangle \not\sqsubseteq s$, and the sequence s shown in Figure 3 is β-unexpected to the given belief.

The recognition of γ-unexpectedness is similar to the above illustration.

4.3 Performance

Our approach is evaluated with Web access logs. Two types of Web access log are used in our experiments: one is a large access log file of an online forum site (labeled as `BBS`), and another is a large access log file of a mixed homepage hosting server (labeled as `WWW`).

Table 1: Web access logs in the experiments

Data Set	Seq. Num.	Dist. Items	Avg. Len.
`BBS`	135,562	126,383	15.5591
`WWW`	53,325	85,810	8.3507

The composition of the two data sets are listed in Table 1. We first apply a sequential pattern mining algorithm to discover frequent sequences for studying the general behaviors of the data sets. The frequent 4-recurrence sequences and 8-recurrence sequences are shown in Figure 4 and Figure 5.

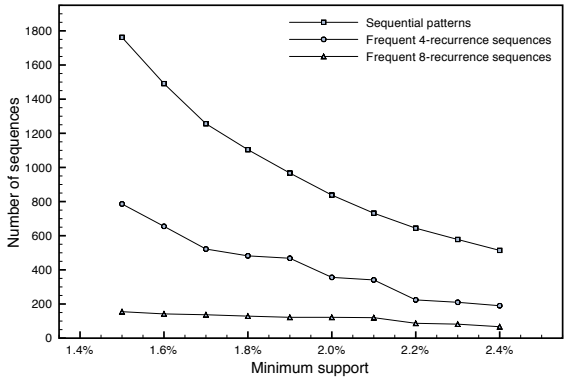

Figure 4: Frequent recurrent sequences in `BBS`.

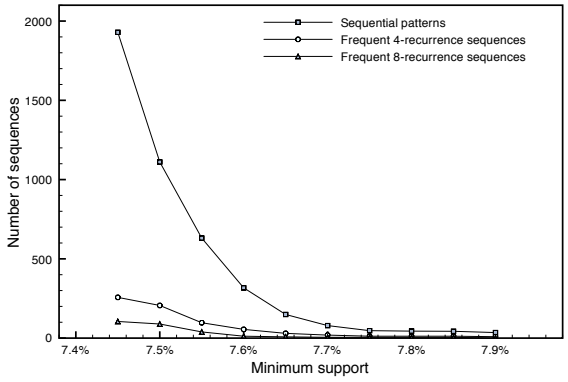

Figure 5: Frequent recurrent sequences in `WWW`.

The recurrent sequences in the data sets show that the recurrence behaviors depend on the semantic characteristics of data, for instance, in our experimental data sets, the recurrence behaviors in online forum site are more stronger than those in mixed content Web site.

We generate 15 beliefs for each data set after examining the sequential patterns and frequent 4-recurrence and 8-recurrence sequences discovered in last step, corresponding to 3 groups of 5 beliefs: with "rarely", "often" and "frequently" (according to Figure 1) appearing in the $\langle s_\alpha, \omega_\alpha \rangle$ part of a belief. Table 2 lists several sample beliefs in our ex-

Table 2: Sample beliefs

$\langle s_\alpha, \omega_\alpha \rangle$	$\langle s_\beta, \omega_\beta \rangle$	$\langle s_\gamma, \omega_\gamma \rangle$
(f=4), rarely	(f=9), rarely	(f=9), often
(f=0)(f=5), often	(f=8), often	(f=4), often
(f=5), frequently	(f=4), rarely	(f=9), often
/~li/, rarely	/~li/pub/, often	/~li/pub/, rarely
/~li/pub/, often	/~li/, rarely	/~li/doc/, often
/~li/, frequently	/~li/doc/, rarely	/~li/doc/, often

periments, where the symbols like "(f=4)" represent a forum ID in URL query string.

Figure 6 and Figure 7 show our experimental results.

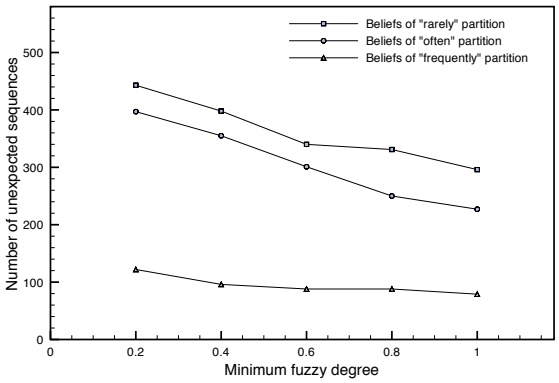

Figure 6: Unexpected sequences in BBS.

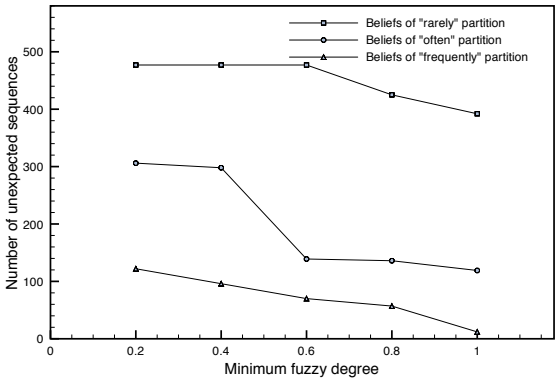

Figure 7: Unexpected sequences in WWW.

With the decrease of the minimum fuzzy degree threshold, the number of unexpected sequences increases. In Figure 6, we find in the "frequently" partition, the number of unexpected sequences is much less than those in the other two partitions, because in the data set the number of long recurrent sequences, such as 8-recurrence sequences, is less. We can also find that the unexpected behaviors focus on the recurrences between "rarely" and "often". In Figure 7, there is a sharp increase of the number of unexpected sequences in the "often" partition when the minimum fuzzy degree decreases from 0.6 to 0.4, because in the "often" partition, the fuzzy degree 0.5 corresponds to 4-recurrence sequences, so that a lot of unexpected sequences in the "rarely" partition are counted as "often".

5. CONCLUSION

In this paper, we introduce the problem of discovering unexpected recurrence behaviors in sequence databases. We propose a novel notion, the fuzzy recurrence rules, for depicting the recurrence behaviors of the data, where fuzzy set theory is applied to describe the recurrence of sequences. We present a belief-driven approach for modeling two types of unexpectedness in recurrence behaviors, where the belief consists in a fuzzy recurrence rule and a semantic constraint on the rule. We also develop an effective algorithm UFR, which discovers all unexpected sequences in a sequence database with respect to domain expert specified belief base and minimum fuzzy degree threshold. The experimental results on Web access logs show the usefulness of our propositions.

Our future research includes the discovery of fuzzy recurrence rules in sequential data, we believe that our proposal of this novel rule model on sequences can be interesting for many real-word application domains.

6. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, pages 207–216, 1993.

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

[3] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential pattern mining using a bitmap representation. In *KDD*, pages 429–435, 2002.

[4] G. Berger and A. Tuzhilin. Discovering unexpected patterns in temporal data using temporal logic. In *Temporal Databases, Dagstuhl*, pages 281–309, 1997.

[5] T. Calders. Computational complexity of itemset frequency satisfiability. In *PODS*, pages 143–154, 2004.

[6] 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.

[7] 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.

[8] 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.

[9] Y.-L. Chen and C.-H. Weng. Mining association rules from imprecise ordinal data. *Fuzzy Sets and Systems*, 159(4):460–474, 2008.

[10] C. Fiot, A. Laurent, and M. Teisseire. Approximate sequential patterns for incomplete sequence database mining. In *FUZZ-IEEE*, pages 1–6, 2007.

[11] D. Gunopulos, R. Khardon, H. Mannila, S. Saluja, H. Toivonen, and R. S. Sharm. Discovering all most specific sentences. *ACM Trans. Database Syst.*, 28(2):140–174, 2003.

[12] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: a frequent-pattern tree approach. *Data Min. Knowl. Discov.*, 8(1):53–87, 2004.

[13] T.-P. Hong, K.-Y. Lin, and S.-L. Wang. Fuzzy data mining for interesting generalized association rules. *Fuzzy Sets and Systems*, 138(2):255–269, 2003.

[14] 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.

[15] C. M. Kuok, A. W.-C. Fu, and M. H. Wong. Mining fuzzy association rules in databases. *SIGMOD Record*, 27(1):41–46, 1998.

[16] D. H. Li, A. Laurent, and P. Poncelet. Mining unexpected sequential patterns and rules. Technical Report RR-07027 (2007), LIRMM, 2007.

[17] D. H. Li, A. Laurent, and P. Poncelet. Discovering fuzzy unexpected sequences with beliefs. In *IPMU*, pages 1709–1716, 2008.

[18] F. Masseglia, F. Cathala, and P. Poncelet. The PSP approach for mining sequential patterns. In *PKDD*, pages 176–184, 1998.

[19] B. Padmanabhan and A. Tuzhilin. A belief-driven method for discovering unexpected patterns. In *KDD*, pages 94–100, 1998.

[20] B. Padmanabhan and A. Tuzhilin. Small is beautiful: discovering the minimal set of unexpected patterns. In *KDD*, pages 54–63, 2000.

[21] 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.

[22] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Mining sequential patterns by pattern-growth: the prefixspan approach. *IEEE Trans. Knowl. Data Eng.*, 16(11):1424–1440, 2004.

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

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

[25] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *SIGMOD*, pages 1–12, 1996.

[26] R. Srikant and R. Agrawal. Mining sequential patterns: generalizations and performance improvements. In *EDBT*, pages 3–17, 1996.

[27] X. Yan, J. Han, and R. Afshar. CloSpan: Mining closed sequential patterns in large databases. In *SDM*, 2003.

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