Mining Unexpected Web Usage Behaviors

Haoyuan Li, Anne Laurent, Pascal Poncelet

Mining Unexpected Web Usage Behaviors

Dong (Haoyuan) Li[1], Anne Laurent[2], and Pascal Poncelet[1]

[1] LGI2P - École des Mines d'Alès, Parc Scientifique G. Besse, 30035 Nîmes, France
{Haoyuan.Li,Pascal.Poncelet}@ema.fr
[2] LIRMM - Université Montpellier II, 161 rue Ada, 34392 Montpellier, France
laurent@lirmm.fr

Abstract. Recently, the applications of Web usage mining are more and more concentrated on finding valuable user behaviors from Web navigation record data, where the sequential pattern model has been well adapted. However with the growth of the explored user behaviors, the decision makers will be more and more interested in unexpected behaviors, but not only in those already confirmed. In this paper, we present our approach USER, that finds unexpected sequences and implication rules from sequential data with user defined beliefs, for mining unexpected behaviors from Web access logs. Our experiments with the belief bases constructed from explored user behaviors show that our approach is useful to extract unexpected behaviors for improving the Web site structures and user experiences.

1 Introduction

Recently, the applications of Web usage mining are more and more concentrated on finding valuable user behaviors from Web navigation record data (also known as Web access logs). A great deal of research work has been performed on porting data mining technologies to the Web usage analysis, in order to improve the personalization, the recommendation, and even the effectiveness of Web sites [1,2,3,4,5,6,7,8,9,10] by exploring the question: *what resources are frequently visited by whom during which periods?*

Among existing technologies, *sequential pattern* mining [11] has been well adapted to answer the above question [4,6,7,8,9]. All those sequential patterns extracted from Web access logs are typically the relationships like "on the Web site of customer support forum, 40% of users visited the `TopicList` page, then the `Search` page, then the `Login` page, and then the `PostTopic` page", or like "in the online store, 10% of customers visited the `notebook cases` page after having added a `notebook computer` to the `shopping cart`". This kind of relationships reflect the most general and reasonable user behaviors during Web navigations, however it become less important once we interpreted them as domain knowledge. When we regularly perform sequential pattern based Web usage mining on access logs, with the growth of the explored user behaviors, the decision makers will be more and more interested in exploring unexpected user behaviors that contradict existing knowledge, but not only in those the already confirmed.

P. Perner (Ed.): ICDM 2008, LNAI 5077, pp. 283–297, 2008.

In this paper, we focus on finding *unexpected behaviors* (that contradict the explored user behaviors) from Web access logs within the context of domain knowledge (that corresponds to the explored user behaviors). To illustrate our goal, let us consider an online news Web site, where the latest news are listed on the static home page `index.html` by categories. The latest previous news can be visited from static category index pages like `cat1.html`, `cat2.html`, etc., and all news can be visited from server side script page `listnews.php` by specifying the category, like `listnews.php?cat=1&page=3`. The server side script page `readnews.php` provides the detail of a specified news identified by `news`, like `readnews.php?news=20080114-002`. Assume that (1) 60% of users visit `index.html`, then various `readnews.php`, then `cat1.html`, then various `readnews.php`, then `listnews.php`, then various `readnews.php`, and then other categories and various `readnews.php`, etc.; (2) 10% of users visit `index.html`, then `cat5.html`, then various `readnews.php`; (3) 8% of users visit `readnews.php` only once; (4) 0.005% of users visit a large number of `readnews.php` only. From traditional sequential pattern mining approaches, we may find the most general user behaviors described in (1) with a suitable minimum support threshold, but it is quite hard to find the behaviors described in (2), (3) and (4) because:

1. Most existing sequential pattern mining approaches do not consider the missing elements, neither the semantic contradictions between elements (e.g. between `cat1.html` and `cat5.html`) in a sequence. The constraint based approaches like SPIRIT [12] may find the sequences of (2) and (3), but the main drawback is that we cannot find all sequences like the one described in (2) by saying "categories contradicting `cat1.html`", but will have to indicate `cat2.html`, `cat3.html`, etc. exactly, since the constraint **not** `cat1.html` implies all pages different to `cat1.html`.
2. According to the model of sequential patterns, the sequences representing (2), (3) and (4) are *contained* in the sequences representing (1). Existing approaches that distinguish the support value of each frequent sequence (instead of maximal frequent sequence), like the *closed sequential pattern* [13], may find the existence of (2), (3) and (4) by computing and comparing the support values, but it is also difficult to indicate them.

The rest of this paper is organized as follows. Section 2 presents the application of our approach USER (Mining Unexpected SEquential Rules) for finding unexpected behaviors from Web access log files. In Sect. 3 we show our experimental results. We introduce the related work in Sect. 4. The conclusion is listed in Sect. 5.

2 Finding Unexpected Behaviors from Web Access Logs

In this section, we present the application of our approach USER for finding unexpected Web usage behaviors. We first propose a formal definition of the session sequence contained in Web access logs, then we detail our approach USER, that finds unexpected sequences and implication rules with user defined

beliefs within the context of session sequences. We also briefly introduce the main algorithm of the approach USER.

2.1 Session Sequences for Web Access Log Analysis

The process of Web usage mining contains three phases, including the data preparation, pattern discovery and pattern analysis [3]. The first phase is necessary for cleaning the uninterested information contained in access logs, and also for converting the log content. Then at the second phase we can apply data mining algorithms to find interesting (sequential) patterns. Finally the last phase helps the user to further analyze the results with visualization and report tools. The principle of these three phases is not different from a general data mining process.

We consider the server side access log files in the NCSA Common Logfile Format (CLF) [14], which is supported by most mainstream Web servers including the Apache HTTP Server and the Microsoft Internet Information Services. The Common Logfile Format is as follows:

```
remotehost rfc931 authuser [date] "request" status bytes
```

A log file is a ASCII text-based file, each line contains a CLF log entry that represents a request from a remote client machine to the Web server. Figure 1(a) shows the CLF log entries contained in a log file of an Apache HTTP Server. Additional fields can be combined into the CLF log entries, such as the *referer* field and the *user agent* field, shown in Fig. 1(b).

```
146.19.33.138 - - [11/Jan/2008:17:40:00 +0100] "GET /~li/ HTTP/1.1" 200 5480
146.19.33.138 - - [11/Jan/2008:17:40:00 +0100] "GET /~li/deepred.css HTTP/1.1" 304 -
146.19.33.138 - - [11/Jan/2008:17:40:27 +0100] "GET /~li/TPBD/TP07.html HTTP/1.1" 200 2599
146.19.33.138 - - [11/Jan/2008:17:40:32 +0100] "GET /~li/TPBD/create.sql HTTP/1.1" 200 1376
146.19.33.138 - - [11/Jan/2008:17:49:21 +0100] "GET /~li/TPBD/TP07.pdf HTTP/1.1" 200 111134
```

(a)

```
146.19.33.138 - - [11/Jan/2008:18:27:35 +0100] "GET /~li/ HTTP/1.1" 200 1436 "-" "Mozilla/5
.0 (Macintosh; U; Intel Mac OS X; fr-fr) AppleWebKit/523.10.6 (KHTML, like Gecko) Version/3
.0.4 Safari/523.10.6"
146.19.33.138 - - [11/Jan/2008:18:29:38 +0100] "GET /~li/doc/ HTTP/1.1" 200 854 "http://www
.lgi2p.ema.fr/~li/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X; fr-fr) AppleWebKit/523.10.6
 (KHTML, like Gecko) Version/3.0.4 Safari/523.10.6"
```

(b)

Fig. 1. (a) Common CLF log file entries. (b) Combined CLF log file entries.

According to the definitions of *item*, *itemset* and *sequence* introduced in [11], an attribute is an item; an itemset $\mathcal{I} = (i_1, i_2, \ldots, i_m)$ is an unordered collection of items; a sequence is an ordered list $s = \langle \mathcal{I}_1 \mathcal{I}_2 \ldots \mathcal{I}_k \rangle$ of itemsets. This model is generally represented as a "Customer-Transaction-Items" relation where each sequence stands for all transactions of a customer identified by "CID" and each

itemset stands a transaction identified by "TID". We propose the *session sequences* representation of Web access log entries, shown in Definition 1.

Definition 1 (Session Sequence). *Let $\mathcal{L}$ be a set of Web server access log entries and $l \in \mathcal{L}$ be a log entry. A session sequence $s \sqsubseteq \mathcal{L}$ is a sequence*

$$s = \langle (ip_s, S_0^s)(l_1^s.url, S_1^s) \dots (l_n^s.url, S_n^s) \rangle,$$

such that for $1 \leq i \leq n$, $l_i^s.url$ is the URL requested from IP address ip_s, and for all $1 \leq i < j \leq n$, $l_i^s.time < l_j^s.time$, where $l_i^s.time$ and $l_j^s.time$ denote the request time for log entries l_i^s and l_j^s. S_0^s is a set of items that contains all optional information of the session s. $S_1^s \dots S_n^s$ are sets of items that contain optional information for each log entry $l_1^s \dots l_n^s$.

The set S_0^s can be empty or contain IP, date, time, user agent, and etc., for reducing the repetition of items. The sets $S_1^s \dots S_n^s$ can be empty or contain HTTP query parameters of each access log entry. So that with session sequences, the log entries can be represented as shown in Fig. 2.

Session No.		IP/URL	Optional Information
1	0	`146.19.33.*`	`17h`
1	1	`/~li/`	
1	2	`/~li/deepred.css`	
1	3	`/~li/TPBD/TP07.html`	
1	4	`/~li/TPBD/create.sql`	
1	5	`/~li/TPBD/TP07.pdf`	
2	0	`146.19.33.*`	`17h`
2	1	`/~li/TPBD/TP07.html`	
2	2	`/~li/TPBD/TP07.pdf`	
2	3	`/index.php`	page=2

⇒

CID	TID	Items
1	1	11, 15
1	2	21
1	3	22
1	4	35
1	5	51
1	6	52
2	1	11, 15
2	2	35
2	3	52
2	4	25, 59

Legend

11: `146.19.3.*`	15: `17h`	21: `/~li/`
22: `/~li/deepred.css`	25: `/index.php`	35: `/~li/TPBD/TP07.html`
51: `/~li/TPBD/create.sql`	52: `/~li/TPBD/TP07.pdf`	59: `page=2`

Fig. 2. Session sequence mapped CLF log entries

The sequential pattern mining finds all maximal frequent sequences with a user defined minimum support value, where the *support* of a sequence is defined as the fraction of the total number of sequences in the database that contain the sequence. So with the minimum support value 0.5, the session sequences shown in Fig. 2 contain a sequential pattern $\langle (11, 15)(35)(52) \rangle$, that is, $\langle (\texttt{146.19.33.*}, \texttt{17h})(\texttt{TP07.html})(\texttt{TP07.pdf}) \rangle$.

2.2 Belief and Unexpectedness on Session Sequences

Before formalizing the belief and unexpectedness on session sequences, we first introduce several additional notions, then propose the occurrence relation and implication rules between sequences.

The *length* of a sequence s is the number of itemsets contained in the sequence, denoted as $|s|$. The *concatenation* of sequences is denoted as the form $s_1 \cdot s_2$, so that we have $|s_1 \cdot s_2| = |s_1| + |s_2|$. The notation $s \sqsubseteq^c s'$ denotes that the sequence s is a *contiguous subsequence* of the sequence s', for example $\langle (a)(b)(c) \rangle \sqsubseteq^c \langle (b)(\underline{a})(a, \underline{b})(\underline{c})(d) \rangle$. We denote the first itemset in a sequence s as $s^\top$ and the last itemset as $s_\perp$. We therefore note $s \sqsubseteq^\top s'$ if $s^\top \sqsubseteq {s'}^\top$, note $s \sqsubseteq_\perp s'$ if $s_\perp \sqsubseteq {s'}_\perp$, and note $s \sqsubseteq_\perp^\top s'$ if $s^\top \sqsubseteq {s'}^\top$ and $s_\perp \sqsubseteq {s'}_\perp$.

Given a sequence s such that $s_1 \cdot s_2 \sqsubseteq s$, the *occurrence relation* $\mapsto^{\langle \mathbf{op}, n \rangle}$ is a constraint on the occurrences of s_1 and s_2 in s, where $\mathbf{op} \in \{\neq, =, \leq, \geq\}$ and $n \in \mathbb{N}$. Let $|s'| \models \langle \mathbf{op}, n \rangle$ denote that the length of sequence s' satisfies the constraint $\langle \mathbf{op}, n \rangle$, then the relation $s_1 \mapsto^{\langle \mathbf{op}, n \rangle} s_2$ depicts $s_1 \cdot s' \cdot s_2 \sqsubseteq^c s$ where $|s'| \models \langle \mathbf{op}, n \rangle$. In addition, we have $\langle \leq, 0 \rangle$ implies $\langle =, 0 \rangle$. We also note $s_1 \mapsto^{\langle \geq, 0 \rangle} s_2$ as $s_1 \mapsto^* s_2$, and note $s_1 \mapsto^{\langle =, 0 \rangle} s_2$ as $s_1 \mapsto s_2$. For example, we have $\langle (a)(b)(c) \rangle \models \langle \geq, 2 \rangle$, $\langle (a)(b) \rangle \not\models \langle >, 2 \rangle$, $\langle (a)(b)(c)(a)(b)(c) \rangle$ satisfies $\langle (a)(b) \rangle \mapsto \langle (c) \rangle$ and $\langle (a)(b) \rangle \mapsto^{\langle \leq, 3 \rangle} \langle (c) \rangle$.

We also propose the *implication rule* on sequences, of the form $s_\alpha \Rightarrow s_\beta$ where s_α and s_β are two sequences. The rule $s_\alpha \Rightarrow s_\beta$ means that the occurrence of s_α in a sequence s, that is, $s_\alpha \sqsubseteq s$, implies $s_\alpha \cdot s_\beta \sqsubseteq^c s$. We constrain such a rule $s_\alpha \Rightarrow s_\beta$ with the occurrence relation $s_\alpha \mapsto^{\langle \mathbf{op}, n \rangle} s_\beta$ between the sequences s_α and s_β, then the constrained rule, denoted as $s_\alpha \Rightarrow^{\langle \mathbf{op}, n \rangle} s_\beta$, means therefore that $s_\alpha \sqsubseteq s$ implies $s_\alpha \cdot s' \cdot s_\beta \sqsubseteq^c s$ where $|s'| \models \langle \mathbf{op}, n \rangle$. Nevertheless, we further consider a semantics constraint on the rule. If the sequence s_γ semantically contradicts to the sequence s_β, denoted as $s_\beta \not\sim s_\gamma$, then $s_\alpha \sqsubseteq s$ implies $s_\alpha \cdot s_\gamma \not\sqsubseteq^c s$, or implies $s_\alpha \cdot s' \cdot s_\gamma \not\sqsubseteq^c s$ with the occurrence relation constraint, where $|s'| \models \langle \mathbf{op}, n \rangle$. With such occurrence relation and semantic contradiction constrained implication rules, we therefore define the belief on sequences as follows.

Definition 2 (Belief). *A belief on sequences consists of a rule $s_\alpha \Rightarrow s_\beta$, an occurrence relation constraint $\tau = \langle \mathbf{op}, n \rangle$, and a semantic contradiction $s_\beta \not\sim s_\gamma$, denoted as $[s_\alpha; s_\beta; s_\gamma; \tau]$. The rule $s_\alpha \Rightarrow s_\beta$ and the occurrence relation constraint $\tau = \langle \mathbf{op}, n \rangle$ depict that given a sequence s, $s_\alpha \sqsubseteq s$ implies $s_\alpha \cdot s' \cdot s_\beta \sqsubseteq^c s$, where $|s'| \models \tau$. The semantic contradiction $s_\beta \not\sim s_\gamma$ further depicts that $s_\alpha \sqsubseteq s$ implies $s_\alpha \cdot s' \cdot s_\gamma \not\sqsubseteq^c s$, where $|s'| \models \tau$.*

Example 1. Let us consider the online news site illustrated in Sect. 1, assume that most users visit the page `index.html` and then at least 3 news by the page `readnews.php`, and then the page `cat1.html`. This fact can therefore be stated by a belief with the implication rule $\langle (\texttt{index.html}) \rangle \Rightarrow (\texttt{cat1.html})$ and the occurrence relation constraint $\langle \geq, 3 \rangle$. If we know (by the Web site layout strategies or the previous result of sequential pattern mining) that the page `cat5.html` is not considered being visited two early, then we can further add the semantics constraint $\langle (\texttt{cat1.html}) \rangle \not\sim \langle (\texttt{cat5.html}) \rangle$. So that finally we have the

belief $[\langle(\texttt{index.html})\rangle; \langle(\texttt{cat1.html})\rangle; \langle(\texttt{cat5.html})\rangle; \langle \geq, 3\rangle]$ for describing such Web usage behaviors.

According to different beliefs, we therefore propose three forms of unexpectedness on sequences: α-unexpectedness, β-unexpectedness and γ-unexpectedness.

Definition 3 ($\alpha-$unexpectedness). *Given a belief $b = [s_\alpha; s_\beta; s_\gamma; *]$ and a sequence s, if $s_\alpha \sqsubseteq s$ and 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 contains the $\alpha-$unexpectedness with respect to the belief b, and s is so called an $\alpha-$unexpected sequence.*

A belief with the occurrence relation constraint $\tau = *$ states that s_β should occur after the occurrence of s_α, so that a sequence s violates $\tau = *$ if and only if no s_β occurs in s after s_α. We also require that s_γ should not occur after s_α in an α-unexpected sequence, since the occurrence of s_γ with respect to any constraint τ will be categorized to the γ-unexpectedness, see Definition 5. For example, with the belief $[\langle(\texttt{index.html})(\texttt{readnews.php})\rangle; \langle(\texttt{index.html})\rangle; \emptyset; *]$, we can find the users who went never back to the home page `index.html` after reading news.

Definition 4 ($\beta-$unexpectedness). *Given a belief $b = [s_\alpha; s_\beta; s_\gamma; \tau]$ $(\tau \neq *)$ and a sequence s, if $s_\alpha \mapsto^* s_\beta \sqsubseteq s$ and there does not exist s' such that $|s'| \models \tau$ and $s_\alpha \mapsto s' \mapsto s_\beta \sqsubseteq^c s$, then s contains the $\beta-$unexpectedness with respect to belief b, and s is so called a $\beta-$unexpected sequence.*

A β-unexpectedness reflects that the implication rule is broken because the occurrence of s_β violates the constraint τ. For instance, as illustrated in Example 1, even as we expected that most users will visit the category index page `cat1.html` after reading at least 3 news from the home page `index.html`, there exist users who read less than 3 news before leaving the home page. With analyzing the sequences containing such a β-unexpectedness stated by the belief $[\langle(\texttt{index.html})\rangle; \langle(\texttt{cat1.html})\rangle; \langle(\texttt{cat5.html})\rangle; \langle \geq, 3\rangle]$, we might further find that, for example, this unexpected behavior mostly happens at the moments when the site is usually less updated. So that new site promotion strategies can be positioned for these periods.

Definition 5 ($\gamma-$unexpectedness). *Given a belief $b = [s_\alpha; s_\beta; s_\gamma; \tau]$ and a sequence s, if $s_\alpha \mapsto^* s_\gamma \sqsubseteq s$ and there exists s' such that $|s'| \models \tau$ and $s_\alpha \mapsto s' \mapsto s_\gamma \sqsubseteq^c s$, then s contains the $\gamma-$unexpectedness with respect to belief b, and s is so called an $\gamma-$unexpected sequence.*

The γ-unexpectedness is concentrated on semantics: the occurrence of s_β is replaced by its semantic contradiction s_γ with respect to the constraint τ. Considering again the above example, we know that the news listed on the page `cat1.html` are semantically different to those listed on the page `cat5.html` (e.g. "All latest news" vs. "All old news", or "Politics" vs. "Entertainments"). If a lot of users visit the news listed on `index.html` then those listed on `cat1.html`, and only a few users visit `cat5.html` instead of `cat1.html`, then it may valuable to explore such an unexpected behavior. For example, assume that (1) from

08h to 23h, 60% of users confirm the explored behavior $\langle(\texttt{index.html})\rangle \mapsto^{\langle\geq,3\rangle} \langle(\texttt{cat1.html})\rangle$; (2) from 23h to 08h of the second day, 80% of users confirm the unexpected behavior $\langle(\texttt{index.html})\rangle \mapsto^{\langle\geq,3\rangle} \langle(\texttt{cat5.html})\rangle$, then it is not difficult to see that the frequency of the sequence describing the behavior (1) can be much more higher than that of those describing the behavior (2). For this reason, frequency based sequence mining approaches are difficult to extract the behavior (2), but such a behavior is valuable to decision makers.

2.3 Unexpected Sequences and Implication Rules

The fact that a sequence s violates a given belief b is denoted as $s \not\models b$. For better describing the structure of unexpectedness, we propose the notions of the *bordered unexpected sequence*, the *antecedent sequence* and the *consequent sequence* within an unexpected sequence.

Definition 6 (Unexpected Sequence). *A sequence s that violates a belief $b = [s_\alpha; s_\beta; s_\gamma; \tau]$ is an unexpected sequence. The bordered unexpected sequence s_u is the maximum contiguous subsequence of s, (1) if s is α-unexpected, we have $s_a \cdot s_u = s$ ($|s_a| \geq 0$) 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 $s_\gamma \sqsubseteq_\perp s_u$. The subsequence s_a is the antecedent sequence. The subsequence s_c is the consequent sequence.*

Given a belief b and a sequence database $\mathcal{D}$, let $\mathcal{D}_U$ be the set of bordered unexpected sequences of each $s \in \mathcal{D}$ that $s \not\models b$, $\mathcal{D}_A$ be the set of antecedent sequences of each $s \in \mathcal{D}$ that $s \not\models b$, and $\mathcal{D}_C$ be the set of consequent sequences of each $s \in \mathcal{D}$ that $s \not\models b$. We therefore define the *unexpected sequential patterns* and the *unexpected implication rules* (including the *antecedent rules* and the *consequent rules*) as follows.

Definition 7 (Unexpected Sequential Pattern). *An unexpected sequential pattern s_U is a maximal frequent sequence in $\mathcal{D}_U$.*

The support for an unexpected sequential pattern is defined as the fraction of total number of sequences in $\mathcal{D}_U$ that support this unexpected sequential pattern:

$$supp(s_U) = \frac{|\{s \mid s_U \sqsubseteq s, s \in \mathcal{D}_U\}|}{|\mathcal{D}_U|}. \tag{1}$$

Definition 8 (Unexpected Implication Rules). *Let u denote the unexpectedness stated by b, an antecedent rule is a rule $s_A \Rightarrow u$ where s_A is a maximal frequent sequence in $\mathcal{D}_A$, and a consequent rule is a rule $u \Rightarrow s_C$ where s_C is a maximal frequent sequence in $\mathcal{D}_C$.*

The support for an antecedent rule $s_A \Rightarrow u$ (or for a consequent rule $u \Rightarrow s_C$) is defined as the fraction of total sequences in $\mathcal{D}_A$ (or in $\mathcal{D}_C$) that support the sequence s_A (or the sequence s_C):

$$supp(s_A \Rightarrow u) = \frac{|\{s \mid s_A \sqsubseteq s, s \in \mathcal{D}_A\}|}{|\mathcal{D}_A|} \tag{2}$$

or

$$supp(u \Rightarrow s_C) = \frac{|\{s \mid s_C \sqsubseteq s, s \in \mathcal{D}_C\}|}{|\mathcal{D}_C|}. \tag{3}$$

In order to describe the implication relations between the antecedent/consequent sequences and the unexpectedness, we measure the confidence of the antecedent rule within the sequence database $\mathcal{D}$, and that of the consequent rule within the consequent sequence set $\mathcal{D}_C$:

$$conf(s_A \Rightarrow u) = \frac{|\{s \mid s_A \sqsubseteq s, s \in \mathcal{D}_A\}|}{|\{s \mid s_A \sqsubseteq s, s \in \mathcal{D}\}|} \tag{4}$$

and

$$conf(u \Rightarrow s_C) = \frac{|\{s \mid s_C \sqsubseteq s, s \in \mathcal{D}_C\}|}{|\{\mathcal{D}_C\}|}. \tag{5}$$

The unexpected sequential patterns reflect the internal structure of the unexpectedness, and the unexpected implication rules reflect the implications and influences of the unexpectedness. For instance, as illustrated in the precedent example for describing Definition 5, assume that from 23h to 08h of the second day, 80% of users confirm the unexpected behavior $\langle(\texttt{index.html})\rangle \mapsto^{\langle\geq,3\rangle} \langle(\texttt{cat5.html})\rangle$, then an antecedent rule can be:

$$\langle(\texttt{23h-08h})\rangle \Rightarrow \langle(\texttt{index.html})\rangle \not\mapsto^{\langle\geq,3\rangle} \langle(\texttt{cat5.html})\rangle,$$

and the confidence of such a rule is 0.8. Assume that 60% of users who violate the behavior $\langle(\texttt{index.html})\rangle \mapsto^{\langle\geq,3\rangle} \langle(\texttt{cat5.html})\rangle$ like to click visit the advertisement displayed by `ads3.php`, then a consequent rule can be:

$$\langle(\texttt{index.html})\rangle \not\mapsto^{\langle\geq,3\rangle} \langle(\texttt{cat5.html})\rangle \Rightarrow \langle(\texttt{ads3.php})\rangle.$$

If we further assume that a unexpected sequential pattern within the unexpectedness $\langle(\texttt{index.html})\rangle \not\mapsto^{\langle\geq,3\rangle} \langle(\texttt{cat5.html})\rangle$ is explored with support value 0.9, for example, the sequential pattern $\langle(\texttt{index.html})(\texttt{readnews.php})(\texttt{cat3.html})\rangle$, then all those facts can interpreted as following:

"Between 23h and 8h, 80% of users do not follow the explored behavior, 90% of those users read news listed on the home page, and then visit the category 3 instead of visiting the category 5, and then 60% of them follow the same kind of online advertisements."

Such unexpected user behaviors can be enough important to decision makers for pushing new Web site design or collaboration strategies.

2.4 The Algorithm USER

Figure 3 briefly shows the algorithm USER. The algorithm accepts a sequence database $\mathcal{D}$, a user defined belief base $\mathcal{B}$, a minimum support threshold min_supp and a minimum confidence threshold min_conf as inputs. It finds all unexpected sequences contained in the sequence database $\mathcal{D}$ with respect to each unexpectedness u stated by each belief $b \in \mathcal{B}$. If a sequence s is unexpected to b, then

s will be partitioned to the antecedent sequence s_a ($|s_a| \geq 0$), the unexpected bordered sequence s_u and the consequent sequence s_c ($|s_c| \geq 0$). When all unexpected sequences have been extracted, the algorithm starts finding unexpected sequential patterns from each group of unexpected bordered sequences with the minimum support threshold min_supp. Finally the algorithm generates the antecedent/consequent rules from each group of antecedent/consequent sequences with the minimum confidence threshold min_conf.

A detailed description of the algorithm USER is listed in [15].

Input : a sequence database $\mathcal{D}$, a user defined belief base $\mathcal{B}$, a minimum support threshold min_supp and a minimum confidence threshold min_conf
Output: all unexpected sequential patterns and implication rules for each unexpectedness

```
1 for each sequence s ∈ D do
2     for each belief b ∈ B do
3         if s is α/β/γ-unexpected to b
4             partition s to s_a, s_u and s_c
5             save s_a, s_u and s_c
6 for each unexpectedness u stated by each b ∈ B do
7     find sequential patterns from each D_U with min_supp
8     generate rules s_A ⇒ u from each D_A with min_conf
9     generate rules u ⇒ s_C from each D_C with min_conf
```

Fig. 3. Sketch of the algorithm USER

3 Experiments

To evaluate of our approach, we performed a number of experiments on two large access log files containing the access records of two Web servers during a period of 3 months. The first log file, labeled as `LOGBBS`, corresponds to a PHP based discussion forum Web site of an online game provider; the second log file, labeled as `LOGWWW`, corresponds to a laboratory Web site that also hosts the personal home pages of researchers and teaching staffs.

In our experiments, we split each log file into three 1-month period files, i.e., `LOGBBS-{1,2,3}` and `LOGWWW-{1,2,3}`. We generate the session sequences with the information of day (Monday to Sunday) and hour (0h to 23h) of the first log entry of a session. If the interval time of two log entries with the same remote client IP address is greater than 30 minutes, then the last log entry starts a new session sequence. Because the CLF log entry does not contain the "`remoteport`" information, at this moment we do not identify the accesses from remote clients hidden behind the proxy servers and NAT gateways, so that long session sequences with the same remote address will be cut into multiple sequences with a length no more longer than 50.

For each session sequence, the remote client IP address is considered as a block, such that the IP `146.19.33.138` will be converted to `146.19.33.*`. We map only significant HTTP query parameters to items. For `LOGBBS`, the number of PHP page are very limited and the parameter can stand for an access request. For example, the request

`/forumdisplay.php?f=2&sid=f2efeb85fcfd94ecbc2dba0f97b678a1`

can be considered as an itemset (`f=2`), and the request

`/viewtopic.php?t=57&sid=f2efeb85fcfd94ecbc2dba0f97b678a1`

can be replaced by the itemset (`t=57`). We focus on the accesses of static HTML pages, server side script pages and JavaScript scripts, hence all other unconcerned files like cascading style sheets, images and data files are ignored (PDF and PS files are kept for `LOGWWW`). Table 1 details the number of session sequences and distinct items, and the average length of the session sequences contained in the Web access logs.

Table 1. Web access logs in our experiments

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

In order to compare our approach with the sequential pattern mining, we first apply the sequential pattern mining algorithm to find the frequent behaviors from `LOGBBS-{1,2,3}` and `LOGWWW-{1,2,3}` with different minimum support thresholds, shown in Fig. 4 and Fig. 5. In an acceptable range, the number of sequential patterns discovered are similar in all of the 3 periods, that increases the difficulty in analyzing new user behaviors. The post analysis shows that, for instance, within the frequent behaviors discovered with the minimum support 0.04 from `LOGBBS-{1,2,3}`, 149 sequential patterns discovered from all of `LOGBBS-{1,2,3}` are similar (contained in each other), i.e. $> 40\%$ of `LOGBBS-1`; 197 sequential patterns discovered from `LOGBBS-{2,3}` are similar, i.e., the similarity of accesses is $> 70\%$ in these two periods.

We then construct the belief bases from the workflow and those frequent behaviors discovered by sequential pattern mining. For `LOGBBS`, we first generate 5 beliefs from the workflow considered on this forum site, and then generate 5 beliefs from a set of selected sequential patterns discovered from `LOGBBS-1`. The following belief corresponds to an "expected" forum browsing order:

$$[\langle(\texttt{/})\rangle; \langle(\texttt{t=2})(\texttt{t=5})\rangle; \langle(\texttt{t=5})(\texttt{t=2})\rangle; \langle=, 0\rangle].$$

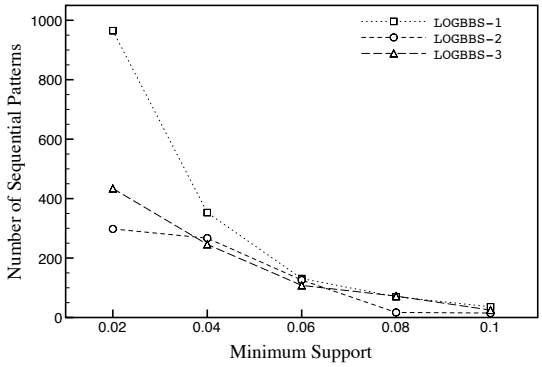

Fig. 4. Frequent behaviors discovered by the sequential pattern mining algorithm from `LOGBBS`

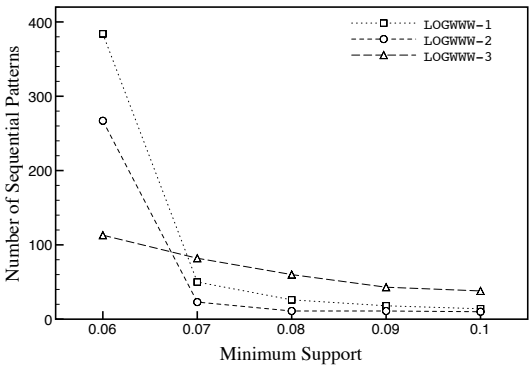

Fig. 5. Frequent behaviors discovered by the sequential pattern mining algorithm from `LOGWWW`

For `LOGWWW` we create 10 beliefs corresponding to the most frequent behaviors discovered from `LOGWWW-1`, for example, according to the navigation menu on the home page of this Web server, we have:

$$[\langle(\texttt{0018-04.html})\rangle; \langle(\texttt{0019-27.html})\rangle; \langle((\texttt{0018-04.html})\rangle; \langle\leq, 5\rangle],$$

where `0018-04.html` corresponds to the index page of the section "Research", `0019-27.html` corresponds to a subsection in "Research" and `0018-04.html` corresponds to a subsection in the section "Publications".

Figure 6 and Fig. 7 show the number of unexpected sequential rules discovered by our approach USER. With comparison between the quantities of frequent user behaviors, our approach generates less than the sequential pattern mining approaches. The analysis of similarity shows that, with the minimum confidence 0.2, only 3 rules are similar from `LOGBBS-{1,2,3}`, 4 similar rules from `LOGBBS-{1,3}`, and from `LOGWWW-{2,3}` we find only 1 similar rule.

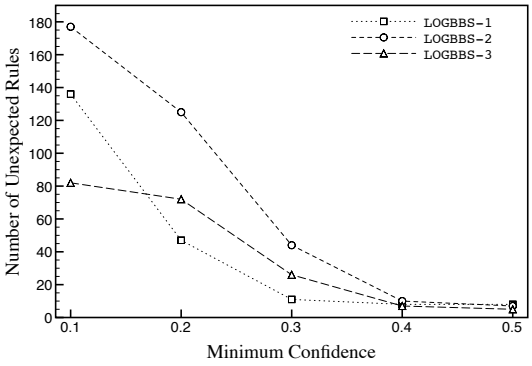

Fig. 6. Unexpected implication rules discovered by the approach USER from `LOGBBS`

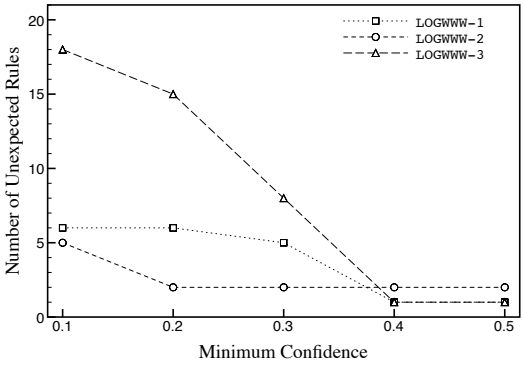

Fig. 7. Unexpected implication rules discovered by the approach USER from `LOGWWW`

We add 10 unexpected rules discovered from `LOGBBS-1`/`LOGWWW-1` into the belief base for `LOGBBS-2`/`LOGWWW-2`, and add 10 (or all of them if the number is less than 10) unexpected rules discovered from `LOGBBS-{1,2}`/`LOGWWW-{1,2}` into the belief base for `LOGBBS-3`/`LOGWWW-3`, where these latest discovered unexpected rules are considered as explored behaviors. Table 2 [1] shows the results.

The principle of our approach does not imply that to add latest-discovered unexpected behaviors into the belief base will regularly affect the number of unexpected rules discovered from the next-period data, however, Table 2 also shows that the number of unexpected implication rules increased between `LOGWWW-1` and `LOGWWW-{2,3}`, that is because the Web server corresponding to `LOGWWW` contains a great deal of non-profilable (e.g. personal home pages) Web content and the accesses are highly depended on the period, so that the beliefs could not be coherent to all data contained in `LOGWWW`.

[1] Number of beliefs: Number of unexpected implication rules.

Table 2. Unexpected rules with the increment of explored behaviors

min_conf	LOGBBS-1	LOGBBS-2	LOGBBS-3	LOGWWW-1	LOGWWW-2	LOGWWW-3
0.1	10:136	20:24	30:127	10:6	16:42	26:34
0.2	10:47	20:18	30:19	10:6	16:18	26:25
0.3	10:11	20:16	30:12	10:5	15:9	24:12
0.4	10:8	18:9	27:4	10:1	11:9	20:10
0.5	10:8	18:6	24:4	10:1	11:8	19:10

4 Related Work

In application domain, a great deal of Web analyzing tools, like the Webalizer [16], offer statistics based Web access analysis. In research domain, many Web usage mining approaches focus on the personalization and recommendation of Web sites by finding the most frequent user behaviors. To the best of our knowledge, we propose the first approach to unexpected Web usage mining, in considering constraints on both of the occurrence and semantics. Our approach considers a knowledge based *subjective interestingness measure* on sequence mining. As being summarized in [17], the interestingness measures for data mining 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 [18] 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.

Based on the proposition of [18], in the most recent approach to unexpected association rule mining presented by [19]. The mining process is done by the *APriori* based algorithms that find the minimal set of unexpected association rules with respect to a set of user defined beliefs. On sequence mining, [20] proposed a framework for finding unexpected sequential rules based on the frequency. In this approach, the author defines the unexpectedness from the constraints on sequential rules that depict the frequency of the content contained in a discovered sequential pattern, and the goal is to find all the sequences that do not satisfy given statistical frequency constraints.

5 Conclusion

In this paper we present the application of our general purposed approach USER for mining unexpected Web usage behaviors. We propose the formal definition of

session sequence contained in Web access log files, for mining user behaviors. We introduce the notions of belief and unexpectedness within the context of session sequences, and propose the unexpected sequential patterns and implication rules. With finding unexpected implication rules, unexpected user behaviors can be studied in order to improve Web site structures and user experiences.

We also present our experiments on the access log files form different kinds of Web sites, the preprocess of such access logs and the management of beliefs are also introduced. Our experimental results show that the effects of unexpected Web usage mining highly depend on explored user behaviors, purpose, and structure of a Web site.

We are interested in applying similarity and fuzzy related approaches to unexpected Web usage mining, such as "many users like to visit the pages similar to `some.page` at **about** 6h p.m.". We are also interested in mining unexpected user behaviors with hierarchical data, for example, with the categories of Web pages, or with the path information contained in the URLs, in order to find more pertinent rules.

References

1. Büchner, A.G., Mulvenna, M.D.: Discovering internet marketing intelligence through online analytical web usage mining. SIGMOD Record 27(4), 54–61 (1998)
2. Spiliopoulou, M., Pohle, C., Faulstich, L.: Improving the effectiveness of a web site with web usage mining. In: WEBKDD, pp. 142–162 (1999)
3. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.-N.: Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations 1(2), 12–23 (2000)
4. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Using sequential and non-sequential patterns in predictive web usage mining tasks. In: ICDM, pp. 669–672 (2002)
5. Eirinaki, M., Vazirgiannis, M.: Web mining for web personalization. ACM Trans. Internet Techn. 3(1), 1–27 (2003)
6. Masseglia, F., Teisseire, M., Poncelet, P.: HDM: A client/server/engine architecture for real-time web usage mining. Knowl. Inf. Syst. 5(4), 439–465 (2003)
7. Huang, Y.-M., Kuo, Y.-H., Chen, J.-N., Jeng, Y.-L.: NP-miner: A real-time recommendation algorithm by using web usage mining. Knowl.-Based Syst. 19(4), 272–286 (2006)
8. Missaoui, R., Valtchev, P., Djeraba, C., Adda, M.: Toward recommendation based on ontology-powered web-usage mining. IEEE Internet Computing 11(4), 45–52 (2007)
9. Masseglia, F., Poncelet, P., Teisseire, M., Marascu, A.: Web usage mining: Extracting unexpected periods from web logs. In: DMKD (2007)
10. Mobasher, B.: Data mining for web personalization. In: The Adaptive Web, pp. 90–135 (2007)
11. Agrawal, R., Srikant, R.: Mining sequential patterns. In: ICDE, pp. 3–14 (1995)
12. Garofalakis, M.N., Rastogi, R., Shim, K.: SPIRIT: Sequential pattern mining with regular expression constraints. In: VLDB, pp. 223–234 (1999)
13. Yan, X., Han, J., Afshar, R.: CloSpan: Mining closed sequential patterns in large databases. In: SDM (2003)

14. NCSA HTTPd Development Team: NCSA HTTPd Online Document: TransferLog Directive (1995), http://hoohoo.ncsa.uiuc.edu/docs/setup/httpd/TransferLog.html
15. Li, D.H., Laurent, A., Poncelet, P.: Mining unexpected sequential patterns and rules. Technical Report RR-07027 (2007), Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (2007)
16. Barrett, B.L.: Webalizer (1997-2006), http://www.mrunix.net/webalizer/
17. McGarry, K.: A survey of interestingness measures for knowledge discovery. Knowl. Eng. Rev. 20(1), 39–61 (2005)
18. Silberschatz, A., Tuzhilin, A.: On subjective measures of interestingness in knowledge discovery. In: KDD, pp. 275–281 (1995)
19. Padmanabhan, B., Tuzhilin, A.: On characterization and discovery of minimal unexpected patterns in rule discovery. IEEE Trans. Knowl. Data Eng. 18(2), 202–216 (2006)
20. Spiliopoulou, M.: Managing interesting rules in sequence mining. In: Żytkow, J.M., Rauch, J. (eds.) PKDD 1999. LNCS (LNAI), vol. 1704, pp. 554–560. Springer, Heidelberg (1999)