ADAPTABLE ALGORITHM FOR DESIGNED WEB PROCESS SEQUENCE DATA ANALYSIS

Hai Wang
Sobey School of Business
Saint Mary's University
Halifax, Nova Scotia B3H 3C3, Canada
hwang@smu.ca

Shouhong Wang
Charlton College of Business
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
swang@umassd.edu

ABSTRACT

A significant interest for Web process design is to discover the discrepancies between the users' transaction process sequences and the desired process sequence for the Web transaction process. Sequence data analysis has been an important approach to analyzing Web log data in the e-commerce field. There have been many methods for sequence data analysis; however, few existing methods can be applied to analyzing designed Web transaction process sequence data for improving Web process design. This paper proposes an adaptable sequence matching algorithm for analyzing designed Web process sequence data for discovering knowledge about the Web process design. An application of this algorithm to a case of online shopping cart abandonment analysis is presented.

Keywords: Web design, Web process sequence data analysis, designed Web process sequences, shopping cart abandonment.

1. Introduction

A Web process is a business process carried out on the World Wide Web. Traditionally, Web process design has been studied in the field of workflow analysis [Cardoso & Sheth 2003; Cardoso 2006]. Recent research has suggested that Web page design has significant impact on online consumers' attitude and behavior towards Web processes [Chatterjee 2008]. A right Web page design must meet specific needs of right consumers for right Web processes [Zhou et al. 2004; Shergill & Chen 2005; Singh *et al.* 2008].

Web log sequences (i.e., user click-stream sequences) represent users' Web access behaviors in carrying out Web processes. Massive Web log sequences can be used to discover the general patterns of these process sequences [Wen et al. 2007; Greco & Guzzo 2007]. These patterns are useful for us to understand the online consumers' behaviors as well as problems in Web process design. For instance, by analyzing online shopping Web log sequences of an online store, one might be able to understand more about why online shoppers abandon shopping carts so often, and how the design of the Web site and transaction processes can be improved to reduce shopping cart abandonment. This paper presents a Web log sequence analysis method for improving Web process design.

In the literature, various data analysis methods have been proposed to analyze Web log data [Cadez et al. 2000; Ester et al. 2002; Jiang & Tuzhilin 2006; Mobasher et al.2002; Manavoglu et al. 2003; Yang & Padmanabhan 2005]. In general, these methods aim at discovering interesting patterns of sets of sequences [Fayyad et al. 1996; Pei et al. 2004]. Common approaches to analyzing sequence data include time series analysis (e.g., [LeBaron & Weigend 1998]), association rules induction (e.g., [Lee et al. 2003]), and sequences pattern discovery (e.g., [Dutta et al. 2007]). While many research reports have been emphasizing on the performance of algorithms, exploring applications of process sequence data analysis results directly to e-commerce is imperative [Wu et al. 2000]. In this paper, we present a new method of analyzing Web log sequence data for the diagnosis of Web process design. This algorithm can be used to reveal useful information and develop knowledge for improving Web process design.

The remainder of this paper is organized as follows. First, we provide a brief overview and discussion of major methods used for analyzing sequence data. Next, we develop an adaptable process sequence matching algorithm of

analysis of Web process sequence data. Then, we present an application of the proposed algorithm to a case of shopping cart abandonment. Finally, we conclude with a summary of the study.

2. Related Work: Methods of Process Sequence Data Analysis

Web process design is a topic in the field of business process design, and has been extensively studied through workflow analysis and management [Ould 1995; Cardoso & Sheth 2003; Cardoso 2006; Rozinat & van der Aalst 2008; van der Aalst et al. 2008]. However, online consumers' attitude and behavior towards Web processes have received little attention. In this paper, we focus on improving Web process design through analyzing online consumers' Web access behavior in Web processes.

A process sequence is defined as a set of ordered elements. An online consumer' Web accesses in a Web process can be treated as a process sequence. Four major categories of data analysis methods for processing sequence data are discussed below.

2.1. Time Series Analysis

Time series are sequence data measured strictly against the time dimension. Because of this property of time series, approaches to time series analysis are structured, in contrast to data analysis methods for general cases of sequences where time is not a key factor (e.g., DNA sequences). Although there have been a variety of methods for time series analysis, the time window method is typically used to describe the pattern of the time-dependent variable (e.g., stock price). A classical time series analysis method is the linear autoregressive moving average (ARMA) model [Box & Jenkin 1970] expressed as follows.

$$x_t = a + \sum_{i=1}^{p} b_i x_{t-i} + \sum_{j=1}^{q} c_j \varepsilon_{t-j} + \varepsilon_t \quad (1)$$

where t is the time; x_t is the value of the time-dependent variable at time t; i and j are window indices; a, b_i, and c_j are regression parameters, and ε_t is the residual term.

In general cases of sequence data where the time is not necessarily a useful static reference dimension, the time series analysis methods, such as the ARMA method, are not particularly useful for sequence data analysis. Also, existing time series analysis methods are powerless in dealing with multiple dependent variables as well as non-numerical sequences (e.g., click-streams).

2.2. Associate Distance Measure Method

According to the associate distance measure (ADM) method, a sequence is a vector [Everitt 1980]. In the ADM method, the difference between two sequences S_1 and S_2 is measured by the Euclidean distance between the two vectors.

$$d = \sum_{i=1}^{n} f_i \quad (2)$$

where i is the position index of the longest sequence, n is the number of the elements of the longest sequence, f_i is the dissimilarity of the elements at position i.

$$\begin{cases} f_i = 1 & \text{if } S_1(i) \neq S_2(i) \\ f_i = 0 & \text{otherwise} \end{cases} \quad (3)$$

Missing values are treated as dissimilarity. In terms of the equivalence between a sequence and a vector, the order of the elements in a sequence is not important. This marks the limitations of the ADM method in cases where the order of the elements is crucial (e.g., click-streams).

2.3. Sequence Alignment Method

According to the sequence alignment method (SAM), a sequence is a set of elements arranged in a certain order [Sankoff & Kruskal 1983]. The SAM method also uses a distance measure to calibrate the similarity between sequences. However, in the SAM method, the similarity between two sequences is measured by the necessary operations to covert one sequence to the other. A general formula to calculate the distance between two sequences is

$$d = w_d D + w_i I + w_r R \quad (4)$$

where D is the number of deletion operation, I is the number of insertion operation, R is the number of reordering operation, and w_d, w_i, and w_r are predetermined weights for deletion, insertion, and reordering operations respectively. For instance, suppose that we have the following two sequences:

S_1: {*a, c, d, m, p, y*}

S_2: {*a, d, c, m, t, y*}

where *a, c, d, m, p, t, y* represent distinct values. We can convert S_1 into S_2 by applying the following operations:

(1) Reorder the second and the third elements.
(2) Delete the fifth element.
(3) Insert *t* to the fifth element.
If we choose $w_d=w_i=2$ and $w_r=1$, the distance between S_1 and S_2 is 5.

In comparison with the ADM method, the SAM method takes the order of the elements into account in measuring the distance between the sequences. Nevertheless, the general formula for distance calculation might be over simplified. Also, the time measure in the sequence is missing in the SAM method.

2.4. High Frequency Patterns Methods

High frequency patterns methods aim to discover the most frequent patterns from the sequence database based on predefined criteria. AprioriAll [Agrawal & Srikant 1995], GSP [Srikant & Agrawal 1996], and PSP [Masseglia et al. 1999] are algorithms in this category of sequence data analysis. Originally, these methods require candidate sequences as seeds to discover high frequency patterns. Later, many algorithms, such as FreeSpan [Han et al. 2000] and PrefixSpan [Pei et al. 2004] use recursive projections to generate candidate sequences and use pattern growth approaches to discover high frequency patterns.

High frequency patterns methods rely heavily on predefined thresholds of frequency. More importantly, frequency is just one of the indicators of sequence data, but may not be important for many real-life sequence data analysis problems [Song et al. 2001].

2.5. Discussion

In summary, there have been several popular methods for processing sequence data analysis. In the business field, time series analysis methods have been used for sequence analysis with a single time sensitive variable during the past decades, but have their limitations in general non-numerical process sequence cases. The ADM and SAM methods are often used for sequence pattern analysis, but do not represent the time measure explicitly. High frequency patterns methods assume the frequency is the criterion for discovering patterns with high frequency. This property makes these methods inadequate in cases where the frequency is not important.

In the e-commerce context, there are two major types of Web log process sequences: (1) browsing-oriented: casual information search and Web navigation; and (2) transaction-oriented: designed Web process sequences (DWPS) such as the online shopping process. A designed Web process sequence (DWPS) is a Web log sequence designed by the designer of a Web process, and it represents the expected behavior of online consumers to carry out the Web process. DWPS are less spontaneous than browsing-oriented sequences because each DWPS has a pre-determined start state and an expected termination state. The start state of a DWPS corresponds to the start point of the Web process, and the termination state corresponds to the exit point of the Web process. Existing research on Web log analysis (e.g., [Hay et al. 2004; Dutta et al. 2007]) has been focusing on browsing-oriented sequences, but has not paid much attention to DWPS.

DWPS have two unique characteristics in comparison with casual information search and Web navigation.

(1) The time aspect is important in DWPS for understanding the online consumers' behaviors, and ought to be taken into account in analysis. The formal definition of DWPS in the next section incorporates the time aspect, and associates each Web access event with a time stamp.

(2) An actual DWPS is a variance of a desired sequence, called "norm process sequence". A norm process sequence is an ideal sequence expected by the designer for the entire online transaction process. Intuitively, a norm process represents the expected online consumers' behavior by the designer. A norm process sequence is usually short, and can be used as a reference for Web log analyses. A Web process may have more than one norm process sequences, depending upon the Web process design. On the other hand, a good Web process design should not have too many DWPS for a specific process to avoid confusion.

Clearly, casual Web navigation and DWPS often mix together. For instance, a buyer may not be so sure about the specific item that she/he needs, and therefore spends much time browsing, adding and removing items repeatedly from the shopping cart, etc. To improve the Web process design, one must make a distinction between these two types of Web logs. The development of an effective method to separate these two types of Web logs and analyze valid DWPS is imperative. Thus, the motivation of research into DWPS is to support pre-meditated Web process design by averting *ad hoc* Web design styles, monitor the users' online shopping behaviors against the design, and continuously improve the Web design.

This study investigates DWPS data analysis for the diagnosis of Web process design to improve e-commerce practices. DWPS are non-numerical, distances measures for process sequences are not relevant, and the frequencies of similar process sequences are not particularly important. Thus, the existing methods of data analysis on process sequence data are not applicable, and a new algorithm must be developed for this study.

3. Analyzing Designed Web Process Sequence Data

3.1. Designed Web Process Sequences

A DWPS is a set of process event-time pairs <e, t>, where e is an event and t is the time of the event; that is,

$$\Theta_k = \{ <e_1, t_1>, <e_2, t_2>, \ldots <e_i, t_i>, \ldots <e_T, t_T> \} \quad (5)$$

where k is the process actor, $<e_1, t_1>$ is the determined designed initial event-time pair for the Web process, $<e_T, t_T>$ is the terminate event-time pair, and e_T is the designed outcome of the Web process. For example, the following is a process sequence of online registration.

$DWPS_{ID\#1054}$ = { <Enter the Home page, 304>, <Click Registration, 309>, <Type User-ID, 313>, ... <Click Finish, 428> }

If we use symbols to encode the event occurrences, and use numbers to encode the time with respect to the time of the initial event, the above DWPS can be rewritten as

Θ_{1054} = { <a, 0>, <b, 5>, <c, 9>, ... <T, 124> }

where a is the event "Enter the Home page", b is the event "Click Registration", c is the event "Type User-ID" … T is the designed outcome "Click Finish". A real DWPS may not end with $<e_T, t_T>$ if the process is aborted.

Formally, Θ can be defined in the Extended Backus-Naur Form (EBNF) notation as follows [Aho et al. 2006]:

$$\Theta ::= <e_1, t_1> | (\Theta, <e, t>) \quad (6)$$

which means that a DWPS has its determined time-stamped initial event and arbitrary numbers of follow-up time-stamped events.

A DWPS has its corresponding designed Web process event sequence which contains events only, as defined below.

$$\theta_k = \{ e_1, e_2, \ldots e_i, \ldots \} \quad (7)$$

The interest of this study is to analyze DWPS to improve Web process design. For a designed Web process such as an online purchasing process, there exists a norm DWPS that is considered to be the desired DWPS for the user to complete the process. Let Ψ denote a DWPS with norm events and their norm times, and ends with the designed termination event $e_{Norm\text{-}T}$. For example, in the above case, the Web designer might define the following Ψ for the online registration process based on the Web design and experimental data of times.

Ψ = { <a, 0>, <b, 2>, <c, 5>, … <T, 70> }

Formally, Ψ can be defined in the EBNF notation as follows [Aho et al. 2006]:

$$\Psi ::= \{<e_{Norm\text{-}1}, t_{Norm\text{-}1}>, <e_{Norm\text{-}T}, t_{Norm\text{-}T}>\} | \{ <e_{Norm\text{-}1}, t_{Norm\text{-}1}>, \alpha, <e_{Norm\text{-}T}, t_{Norm\text{-}T}> \}$$

$$\alpha ::= <e_{Norm}, t_{Norm}> | (<e_{Norm}, t_{Norm}>, \alpha) \quad (8)$$

where α is the norm designed partial Web process sequence which does not include the designed termination event $e_{Norm\text{-}T}$.

Ψ may include local iteration loop process sequences (e.g., repeating shopping items), but must have one certain termination event (e.g., confirm the payment). For a Web process, Ψ might not be unique, depending upon the Web process design. In this study, we assume that there is only one Ψ, which might include certain local iteration loops, for a Web process in investigation. The approach discussed here can be extended to multiple Ψ cases. A norm designed Web process sequence has its corresponding norm designed Web process event sequence, as defined below.

$$\psi = \{ e_{Norm\text{-}1}, \ldots E_{Norm\text{-}i}, \ldots e_{Norm\text{-}T} \} \quad (9)$$

where $E_{Norm\text{-}i}$ is any local iteration loop of norm designed partial Web process events which does not include the designed termination event $e_{Norm\text{-}T}$. In this study only one layer loop is considered.

ψ of a Web process is determined by the design of the Web process. The norm times are the desired times for the corresponding norm events, and are usually estimated by using typical DWPS data or experimental DWPS data. Updating Ψ for a Web process is important, and can be viewed as one of the outcomes of data analysis on DWPS.

3.2. Designed Web Process Sequences Analysis

3.2.1. Adaptable process sequence matching

Analyzing DWPS is to measure the dissimilarity between all Θ and Ψ. Knowledge and interpretations of those discrepancies can be useful for improving the design of the Web process. This study uses process sequence matching for analyzing DWPS. Process sequence matching is a division of pattern matching in computing (e.g., [Boyer & Moore 1977; Knuth et al. 1977]). There has been a very rich literature on sequence matching covering a wide variety of topics. However, a sequence matching method is always related to a particular problem domain, and none of the existing sequence matching method can be used directly for analyzing DWPS because of the irregularity of DWPS. We develop an adaptable process sequence matching method in this study. Our adaptable process sequence matching method is different from the exact process sequence matching methods in that heuristics relevant to DWPS must be applied. Two characteristics of DWPS call for the adaptable process sequence matching method. First, uncertain local loops in the process sequences are normally involved. For instance, in an online shopping Web

process the customer can repeat purchasing items for arbitrary times. Second, uncertain interruptions of the process sequences might happen. For instance, in an online shopping Web process, the customer might leave the Web site in the middle of shopping, and may or may not return. The interval between leaving and revisiting is uncertain. It is possible to search the entire process sequence database to assemble interrupted process sequences, but the computational cost used for such a search would be prohibitively expensive. The proposed adaptable process sequence matching method meets the needs in these two aspects, as described next.

3.2.2. Assemble DWPS

The first step of the adaptable process sequence matching method is to assemble DWPS based on the Web log database to obtain Ω, the set of Θ, for each of the process actors. To improve the performance of process sequence matching, two practical methods are used in assembling DWPS. First, δ, the threshold for interruption intervals, is applied in searching the Web log database in order not to search the entire database for exceptional cases. For instance, if the online shopping customer left the Web site and did not return to the process within a half hour, it is assumed that the process is inadequately terminated. In such a way, casual Web navigation logs will be excluded so that valid DWPS will be used for the analysis. Second, a process actor (e.g., customer) might re-do a part of the process by clicking a "back" button. To reduce the complexity of the followed matching process, the withdrawn parts are excluded when assembling Θ.

3.2.3. Generate norm DWPS for the Web process

The second step of the adaptable process sequence matching method is to generate Ψ for the Web process. It has two sub-steps. First, ψ (the norm designed Web process event sequence) is generated by the Web designer to normalize the desired event sequence for the Web process. Each norm event in ψ must be reached by the process actor in the correct sequence. ψ might include local loop prototype E, as explained in Equation (9). Second, randomly select complete samples of $\Theta \in \Omega$, use their segments of process pairs <e, t> to find average t for each e and each event in E, and then use ψ and the corresponding t values to generate Ψ.

3.2.4. Adaptable Matching

Once Ψ and Ω are generated based on the Web log database, an adaptable matching process takes place. Each $\Theta \in \Omega$ is the input of the adaptable matching process, and is used to match Ψ. The output of the adaptable matching process is a table. Each row of the table lists the event and local loops specified by Ψ, the number of the processes that has reached the corresponding event (or local loop), and the average interval time between the corresponding event (or local loop) and the next event (or local loop).

The adaptable matching process goes through the following steps. The first event $e_{Norm\text{-}1}$ of Ψ is used to search Θ. If $e_{Norm\text{-}1}$ is found in Θ, then record the time t_1, and the process continues to the second event $e_{Norm\text{-}2}$. This process proceeds until $e_{Norm\text{-}T}$ is reached, or Θ is ill-terminated. If the event in Ψ is a local loop, $E_{Norm\text{-}i}$, which could be a set of event, is used to search Θ repeatedly until the loop is ended in Θ.

3.2.5. The adaptable process sequence matching algorithm

The above DWPS adaptable process sequence matching method was implemented in C++. The algorithm is summarized as follows.

Input: A Web log database Π; a norm designed Web process event sequence ψ; a threshold for interruption intervals of the Web process δ.

Output: A DWPS summary table Γ(*m*,*4*) that includes *m* rows and *4* columns. Each row lists event, indicator of local loop, the number of the processes that has reached the event, and the average interval time between the corresponding event and the next event.

Notations:
w: Web log;
Π: Web log databases;
A: Web process actor;
Ψ: Norm designed Web process sequence;
ψ: Norm designed Web process event sequence;
Ω: Assembled designed Web process sequences database;
Θ: Assembled designed Web process sequence;
q: Number of event-time pairs in Θ;
θ: Assembled designed Web process event sequence;
e: Event of Web process;
E: Local iteration loop of norm designed partial Web process events;
t: Time of the event of Web process;
δ: Threshold for interruption intervals of the Web process;
T: Cumulative interrupted interval of a Web process;
Γ: DWPS summary table.

Step 1. Assemble designed Web process sequences Θ.

1-1. Select first w $\in \Pi$, identify the process actor A, process event e, and time t.

1-2. If A is not in Ω, search Π for all w with A while A is active in the time window [t, t+δ]. Assemble all w into Θ for actor A, so that the first event-time pair of Θ must be $<e_1, t_1>$, all event-time pairs are ordered in the time sequence, and q is the length of Θ.

1-3. Delete w from Π.

1-4. Repeat Step 1-1 through Step 1-3 until Π=Ø. The set of Θ is Ω.

Step 2. Generate the norm designed Web process sequence Ψ.

2-1. Randomly select *n* complete Θ from Ω;

2-2. For each $e_i \in \psi$, search each of the complete Θ to find $<e_i, t> \subseteq \Theta$, cumulate these t, find average t_i for e_i, and generate $<e_i, t_i>$.

2-3. For each $E_i \in \psi$, find $<e_i, t_i>$ ($e_i \in E_i$), and obtain $<E_i, t_i>$.

2-4. Generate Ψ using all obtained $<e_i, t_i>$ and $<E_i, t_i>$.

Step 3. Match $\Theta \in \Omega$ against Ψ, and produce table Γ.

3-1. Initialize table Γ(*m*,*4*) so that the first two columns list all events in ψ and indicators of local loops, the third and fourth columns are zeros.

3-2. For each $\Theta \in \Omega$, do 3-3 through 3-5 until Ω =Ø .

3-3. For the first event-time pair $<e_1, t_1>$, add 1 to the third column in row e_i of table Γ.

3-4. For each event-time pair $<e_j, t_j> \in \Theta$ (j=2…q), do the following.

Case 1: $e_j \in \theta$ and $e_j \in \psi$ ($e_j \in E_i$ when e_j is a local loop event). Add 1 to the third column in row e_i of table Γ, and add $(t_j - t_{j-1})$ to the fourth column in the row e_{j-1} of table Γ. Set T=0 and exit the case.

Case 2: $e_j \in \theta$ and $e_j \notin \psi$. Add t_j to T. If T< δ, exit the case; otherwise exit this step.

3-5. Delete Θ from Ω.

The pseudo-code of this algorithm is presented in Listing 1. The time complexity of this algorithm is $O(|\Pi|^2)$, where |Π| is the number of DWPS in the Web log database Π. Note that, in principle, the number of DWPS in the Web log database is about the magnitude of the number of online shopping visitors. Theoretically, the size of a Web log database could be accumulated to several gigabytes. However, practically, one might choose a limited number of DWPS (e.g., estimated number of online shoppers of a few recent purchasing cycles) for analysis. More importantly, because the time spans of usual DWPS are not long, multiple full scans of the entire Web log database seldom occur. Hence, the average time complexity of this algorithm is considered to be scalable with a large data size of Web logs.

Listing 1. Adaptable Algorithm for Designed Web Process Sequence Analysis

```
/* Step 1. Assemble designed Web process sequences Θ. */
While (Π=Ø) {
        Select first w ∈ Π;
        A= the process actor of w
        If (A is not in Ω) {
                For each event e in w {
                        t = the time of e;
                        Search Π for all w with A while A is active in the time window [t, t+δ];
                }
                Assemble w into Θ for actor A, so that the first event-time pair of Θ must be <e1, t1 > and all
                event-time pairs are ordered in the time sequence;
        }
        Delete w from Π;
}
 Ω =the set of Θ;
/* Step 2. Generate the norm designed Web process sequence Ψ. */
Randomly select n complete Θ from Ω.;
For each ei ∈ ψ {
      Search each of the complete Θ to find <ei, t>⊆Θ , cumulate these t, find average ti for ei, and generate <ei, ti >;
}
For each Ei ∈ ψ {
        Compute <ei, ti > (ei ∈ Ei), and <Ei, ti >;
```

```
}
Generate Ψ using all obtained <e_i, t_i > and <E_i, t_i >;

/* Step 3. Match Θ∈Ω against Ψ, and produce table Γ. */
Initialize table Γ(m,4) so that the first two columns list all events in ψ and indicators of local loops, the third and
        fourth columns are zeros;
For each Θ∈Ω {
        For the first event-time pair <e_1, t_1>, add 1 to the third column in row e_i of table Γ;
        For each event-time pair <e_j, t_j>∈Θ (j>1) {
                If  (e_j∈θ and e_j∈ψ for e_j∈E_i when e_j is a local loop event) {
                        Add 1 to the third column in row e_i of table Γ;
                        Add (t_j-t_j-1) to the fourth column in the row e_j-1 of table Γ;
                        T=0;
                }
                If (e_j∈θ and e_j ∉ ψ) {
                        T = T + t_j ;
                        If (T>= δ) {exit;}
                }
        }
        Delete Θ from Ω;
}
```

We have reviewed three well known commercial software products for sequence data analysis: IBM DB2 Intelligent Miner [2009], SAS Enterprise Miner [2009], and SPSS PASW Modeler [2009]. While all of these competitive software products are capable to extract general patterns and trends from the sequence data set, none of them is able to perform adaptable process sequence data matching for the diagnosis of Web process design.

As discussed in the literature review of Section 2, sequence data analysis has been widely applied to various fields including finance, linguistics, genomics, proteomics, Web log data mining, etc. There have been many sequence data analysis methods, and each of them was designed for a certain type of problem. Our proposed adaptable process sequence matching algorithm is distinct from the existing methods by integrating the following two characteristics. First, in terms of pattern extraction, our algorithm is aimed to discover the patterns of variance of actual Web process sequences against a desired norm process, but not merely the general patterns hidden in the Web log sequence data. Second, in terms of matching, our algorithm is designed to perform adaptable matching, but not exact matching. The context of adaptable matching in this study is analysis of designed Web process sequences. We believe that our algorithm has reached a new development beyond tuning an existing algorithm.

4. A Case of Online Shopping Cart Abandonment

In this section we describe the use of the proposed DWPS data analysis method for an online shopping cart abandonment case. By analyzing the results of the proposed method, the Web designer is able to learn useful information about online shopping cart abandonment to identify the weakest links of the entire Web process sequence. According to eshopability.com [2009], 75% of respondents abandoned their cart without making a purchase. Shopping cart abandonment is costing online merchants billions dollars a year [ME 2009]. There are many reasons for shopping cart abandonment: shopping information is unavailable, checkout process is confusing, a bad Web site usability, etc. [CO 2009; CL 2009; VI 2009]. However, common-sense explanations have not been fully studied in the literature. To discover knowledge about shopping cart abandonment for a particular Web site, data analysis on DWPS is particularly useful.

The Web log database used in this case study was process logs of an online house-hardware catalog sales Web site. These process logs were recorded by the server of the Web site. Each log entry included the visitor's IP address, process point, and process timing.

2856 Θ which reached a shopping cart were assembled for this experiment. Each Θ started from adding an item to the shopping cart, and was supposed to be ended with confirm of purchase. Among these Θ, 821 (28.7%) Θ were processes without shopping cart abandonment, and 2035 (71.3%) Θ abandoned the shopping carts for some reasons. This experiment was to analyze those reasons.

After examining the Web site, ψ (norm designed Web process event sequence) was generated. 821 complete Θ were used to determine the norm time for each norm event. Repetition of choosing merchandise was the only local loop of processes. These data were used to determine Ψ (norm designed Web process sequence).

δ (threshold for interruption intervals of the Web process) was set to 1 hour; that is, if a customer left the Web site for longer than 1 hour without reaching the check out point, the shopping cart was treated as abandoned in this experiment.

The adaptable process sequence matching method was applied, and the output table Γ was obtained (see Table 1 for a segment). The table revealed interesting facts. First, there were significant time deviations between the DWPS with shopping cart abandonment and the designed norm DWPS. Second, the majority of shopping cart abandonment cases (42%) happened right after the check-out point before the payment process.

Table 1. A Segment of Sequence Matching Results

Event (e_i)	Local loop	Number of processes has reached e_i	Average interval time between e_i and e_{i+1} (Seconds)
Adding an item to shopping cart	No	2856	32.75
Adding more than one item to shopping cart	Yes	2573	44.68
Check out	No	1941	10.12
Enter shipping address	No	1086	54.24
Enter credit card number	No	923	41.06
Confirm	No	821	-

Table 2. The Analysis Results of the Web Process Sequences and Interpretations

Abandonment Point	Percentage	Cumulative Time Deviation	Interpretations	Recommendations for Web Designer
Left the Web site after adding an item to a cart	13.9% (283/2035)	N/A	- Prices of products might not be competitive - Process might be confusing	- Make shopping cart visible all the time
In the middle of shopping before check-out	31.1% (632/2035)	+ 356%	- Searching and browsing products are not easy - Process might be confusing and the shopper is lost - Not all merchandise items have competitive prices - Shopping cart might not be functioning	- Improve searching and browsing products - Improve back buttons - Make shopping cart visible all the time
After check-out before entering shipping address	42.0% (855/2035)	+508%	- Shipping cost is higher than expected	- Show shipping and handling costs information earlier
After entering shipping address before entering credit card number	8.0% (163/2035)	+615%	- The shopper does not like to use credit card number	- Provide other payment alternatives (e.g., PayPal)
After entering credit card number before confirm	5.0% (102/2035)	+896%	- Invalid credit card number - Confusion	- Improve back buttons

Meaningful knowledge development through data analysis always depends upon the interactions among business insiders and data analysts. In this case, the data analysis results were reviewed by online sales managers and Web designers and lead to recommendations for improving Web design to reduce shopping cart abandonment. As the processes with shopping cart abandonment wasted much time in the middle shopping process, the design for searching and browsing products was called for further examination. The design of back buttons might also need to be inspected to see whether these buttons were helpful for searching and browsing products. Given the fact that the majority of shopping cart abandonment cases took place after the check-out point, it was naturally considered that

the uncertainty of shipping and handling costs contributed to the problem. Indeed, it was a challenging task for the Web designer to design the way of presenting shipping and handling cost information, as shipping and handling costs highly depended on the sizes and weights of hardware items. The data analysis results and their interpretations by the online sales team are summarized in Table 2. Clearly, an effective use of the proposed adaptable algorithm is always supported by a full understanding of the specific Web design and a comprehensive content analysis of the particular Web process, as illustrated in this case.

5. Conclusions

Our discussion on existing methods for analyzing process sequence data as well as the characteristics of DWPS suggests that the current existing sequence data analysis methods are inadequate for analyzing DWPS for Web process design. This paper proposes the adaptable process sequence matching method for analyzing DWPS. This method is to assemble DWPS in the online process database, and match these irregular DWPS against the norm DWPS. The analysis of the output of this method by business insiders and Web process designers can reveal useful information and develop knowledge for improving Web process design.

As an application example, we have applied this method to a shopping cart abandonment case. Based on the case study, we are convinced that this method can be useful for improving Web process design.

As a powerful technique, sequence data analysis has been widely applied in various fields. A good sequence data analysis method shall address specific types of problems, as this study suggests. We conclude that designed Web process sequence data analysis is useful for improving e-commerce practices.

Acknowledgment

The comments of three anonymous reviewers have contributed significantly to the revision of the article. The first author is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC Grant 312423).

REFERENCES

Agrawal, R. and R. Srikant, R. "Mining sequential patterns," *Proceedings of the 11th International Conference on Data Engineering*, Taiwan, 3-14, 1995.

Aho, A. V., M. S. Lam, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques and Tools*, 2nd edition. New York, NY: Addison-Wesley, 2006.

Box, G. E. P. and G. M. Jenkin, *Time Series Analysis: Forecasting and Control*, San Francisco, CA: Holden-Day, 1970.

Boyer, R. and J. Moore, " A fast string search algorithm," *Communication of the ACM*, Vol. 20: 762-772, 1977.

Cadez, I. V., D. Heckerman, C. Meek, P. Smyth and S. White, "Visualization of navigation patterns on a web site using model-based clustering," *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 280-284, 2000.

Cardoso, J. "Poseidon: A framework to assist Web process design based on business cases," *International Journal of Cooperative Information Systems*, Vol. 15, No. 1: 23-55, 2006.

Cardoso, J. and A. Sheth, "Semantic e-workflow composition," *Journal of Intelligent Information Systems*, Vol. 21, No. 3:191-225, 2003.

Chatterjee, P. "Are unclicked ads wasted? Enduring effects of banner and pop-up ad exposure on brand memory and attitudes," *Journal of Electronic Commerce Research*, Vol. 9, No. 1:51-61, 2008.

CO, content.websitegear.com, <http://content.websitegear.com/article/shopping_cart_abandonment.htm> [accessed on January 30, 2009].

CL , clickz.com, <http://clickz.com/showPage.html?page=2245891> [accessed on January 30, 2009].

eshopability.com, <http://www.eshopability.com> [accessed on January 30, 2009].

Dutta, K., D. VanderMeer, A. Datta, P. Keskinocak, and K. Ramamritham, "A fast method for discovering critical edge sequences in e-commerce catalogs," *European Journal of Operational Research*, Vol. 181: 855-871, 2007.

Ester, M., H. P. Kriegel, and M. Schubert, "Web site mining: a new way to spot competitors, customers and suppliers in the world wide web," *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 249-258, 2002.

Everitt, B. *Cluster Analysis*, NY: Halsted Press, 1980.

Fayyad, U., D. Haussler and P. Stolorz, "Mining scientific data," *Communications of the ACM*, Vol. 39, No. 11:51-57, 1996.

Greco, G. and A. Guzzo, "An information-theoretic framework for process structure and data mining," *International Journal of Data Warehousing and Mining*, Vol. 3, No. 4: 99-118, 2007.

Han, J., J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal and M. C. Hsu, "FreeSpan: Frequent pattern-projected sequential pattern mining," *Proceedings of 2000 International Conference on Knowledge Discovery and Data Mining*, Boston, MA, 355-359, 2000.

Hay, B., G. Wets and K. Vanhoof, "Mining navigation patterns using a sequence alignment method," *Knowledge and Information Systems*, Vol. 6:150-163, 2004.

IBM, DB2 Intelligent Miner, <http://www.ibm.com> [accessed on April 22, 2009].

Jiang, T. and A. Tuzhilin, "Improving personalization solutions through optimal segmentation of customer bases," *Proceedings of the 6th IEEE Conference on Data Mining*, 307-318, 2006.

Joh, C. H., H. J. P. Timmermans and P. T. L. Popkowski-Leszczyc, "Identifying purchase-history sensitive shopper segments using scanner panel data and sequence alignment methods," *Journal of Retailing and Consumer Services*, Vol. 10:135-144, 2003.

Knuth, D., J. Morris and V. Pratt, "Fast pattern matching in strings," *SIAM Journal of Computing*, Vol. 6, No. 1:323-350, 1977.

LeBaron, B. and A. S. Weigend, "A bootstrap evaluation of the effect of data splitting on financial time series," *IEEE Transactions on Neural Networks*, Vol. 9, No. 1:213-220, 1998.

Lee, C. H., M. S. Chen and C. R. Lin, "Progressive pattern miner: An efficient algorithm for mining general temporal association rules," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 4:1004-1017, 2003.

Manavoglu, E., D. Pavlov and C. L. Giles, "Probabilistic user behavior models," *Proceedings of the 3rd IEEE Conference on Data Mining*, 203-210, 2003.

Masseglia, F., P. Poncelet and R. Cicchetti, "An efficient algorithm for Web usage mining," *Networking and Information Systems Journal*, Vol. 2, No. 5/6: 571-603, 1999.

ME, marketingexperiments.com, <http://www.marketingexperiments.com/> [accessed on January 30, 2009].

Mobasher, H., H. Dai, T. Luo and M. Nakagawa, M. "Using sequential and non-sequential patterns for predictive web usage mining tasks," *Proceedings of the 2nd IEEE Conference on Data Mining*, 669-672, 2002.

Ould, M. A. *Business Processes: Modelling and Analysis for Re-engineering and Improvement*. Chichester, England: John Wiley & Sons, 1995.

Pei, J., J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal and M. C. Hsu, "Mining sequential patterns by pattern-growth: The PrefixSpan approach," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 10:1424-1440, 2004.

Rozinat, A. and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior." *Information Systems*, Vol. 33, No. 1:64-95, 2008.

SAS, SAS Enterprise Miner, <http://www.sas.com> [accessed on April 25, 2009].

Sankoff, D. and J. B. Kruskal, *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*, Reading, MA: Addison-Wesley, 1983.

Shergill, G. S. and Z. Chen, "Web-based shopping: Consumers' attitudes towards online shopping in New Zealand," *Journal of Electronic Commerce Research*, Vol. 6, No. 2:78-92, 2005.

Singh, N., D. W. Baack, S. K. Kundu and C. Hurtado, "U.S. Hispanic consumer e-commerce preferences: Expectations and attitudes towards Web content," *Journal of Electronic Commerce Research*, Vol. 9, No. 2:162-175, 2008.

Song, H. S., J. K. Kim, and S. H. Kim, "Mining the change of customer behavior in an internet shopping mall," *Expert Systems with Applications*, Vol. 21, No. 3:157-168, 2000.

SPSS, PASW Modeler, <http://www.spss.com> [accessed on April 23, 2009].

Srikant, R. and R. Agrawal, "Mining sequential patterns: generalizations and performance improvements," *Proceedings of the 5th International Conference on Extending Database Technology*, Avignon, France, 1996.

van der Aalst, W. M. P., M. Dumas, C. Ouyang, A. Rozinat and E. Verbeek, "Conformance checking of service behavior," *ACM Transactions on Internet Technology*, Vol. 8, No. 3:13:1-13:30, 2008.

VI, bisibility.tv, <http://visibility.tv/tips/shopping_cart_abandonment.html> [accessed September 14, 2009].

Wen, L., W. M. P. van der Aalst, J. Wang and J. Sun, "Mining process models with non-free-choice constructs," *Data Mining and Knowledge Discovery*, Vol. 15, No. 2:145-180, 2007.

Wu, X., P. Yu, G. Piatesky-Shapiro et al. "Data mining: How research meets practical development?" *Knowledge and Information Systems*, Vol. 5, No. 2: 248-261, 2000.

Yang, Y. and B. Padmanabhan, "GHIC: A hierarchical pattern-based clustering algorithm for grouping web transactions," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 9:1300-1304, 2005.

Zhou, L., W. K. Chiang and D. Zhang "Discovering rules for predicting customers' attitude toward Internet retailers," *Journal of Electronic Commerce Research*, Vol. 5, No. 4:228-239, 2004