

Link Based Session Reconstruction: Finding All Maximal Paths

Murat Ali Bayir^{*}
University at Buffalo, SUNY

Ismail Hakki Toroslu
Middle East Technical University

ABSTRACT

This paper introduces a new method for the session construction problem, which is the first main step of the web usage mining process. Through experiments, it is shown that when our new technique is used, it outperforms previous approaches in web usage mining applications such as next-page prediction.

Keywords

Web Usage Mining, Graph Theory

1. INTRODUCTION

The purpose of Web Usage Mining (WUM) [3] is to find interesting knowledge about navigation behaviors of web users. The first step of WUM includes the session construction from user logs which directly affects the quality of patterns discovered in WUM process. Previous approaches [4] for session reconstruction has two problems. They are either using time information without link data or add artificial backward movements to complete paths in web topology which generates a noise in page view sequences. These problems can be handled by using cookies and adding client specific information in server requests. However, for various reasons, such as security and changes in the internal structure of web site, some site owners may not want to use proactive approaches at all. Instead of that, these site owners prefer to process only their raw server logs.

Our Previous method Smart-SRA [1, 2] solved most of the problems mentioned above. However, it still can not capture particular user behaviors due to its greedy nature when user navigation is more complex. To overcome the problems of Smart-SRA and previous approaches, we propose a new link based technique, called as Complete Session Reconstruction Algorithm (C-SRA). C-SRA is very powerful algorithm which produces complete set of maximal paths

^{*}Murat Ali Bayir is currently with Google, NY, USA, his contact email is bayir@google.com

that can be obtained from given page request sequence and web topology.

2. COMPLETE-SRA

Complete Session Reconstruction Algorithm (CSRA) is a two phased session reconstruction algorithm which produces link based sessions with all possible maximal sequences. In the first phase of C-SRA, user log sequences from server logs including $\langle \text{IP}, \text{URL}, \text{Time} \rangle$ tuples are partitioned into smaller candidate sessions such that each one of these candidate sessions satisfy both page stay and session duration time constraints mentioned in [1]. The second phase of C-SRA constructs all maximal navigation sequences from the candidate sessions generated at the first phase of the algorithm. We define session reconstruction problem as a graph problem which is called Maximal Paths in a Vertex Sequence (MPVS) as follows:

Problem [Maximal Paths in a Vertex Sequence]: Given vertex sequence and directed graph, determine all maximal paths in the given ordered vertex sequence.

Input: A possibly cyclic directed graph $G = (V, E)$ such that $V = \{v_1, v_2, \dots, v_n\}$ is vertex set and $E \subset V \times V$ is a set of edges, and a sequence of vertices $S = [vs_1, vs_2, \dots, vs_k]$ where each $vs_i \in V$ (without any repetition for our problem, since the second request of the same page is always provided by the browser cache for limited time interval).

Output: Set $\sum_j s_j$, where, each $s_j = \langle vs_{j1}, vs_{j2}, \dots, vs_{jm} \rangle$ is a maximal navigation sequences of S corresponding to a paths in G . That is, for every pair of consecutive vertices in a sequence $\sum_j s_j$, such as vs_{jp} and $vs_{j(p+1)}$, there exists an edge $(vs_{jp}, vs_{j(p+1)}) \in E$. In addition, in order to satisfy the maximality property, there is no other sequence $\sum_q s_q$ of S in \sum such that $\sum_j s_j$ is a sub-string of $\sum_q s_q$.

Below we describe the details of C-SRA. The main part of the second phase of C-SRA corresponds to the maximal paths in a vertex sequence problem. As an input to our algorithm, we were given user web page request sequence as vertex sequences of the web site graph, and the web site topology where vertices represent web pages and edges represent links among web pages.

Phases of CSRA

Input: Page request sequence of a user, given in timestamp order (UserRequestSequence) and topology of web site in adjacency matrix form (Link).

Output: The set of all maximal navigation sequences (MSeqSet).

Phase 1: Construct the candidate sessions set (CandidateSessionSet) from user page request sequence (UserRequestSequence), by using both of the time thresholds. That means for each candidate session constructed, both the total duration time of session and the time spent on a page in a session will be limited.

Phase 2: This phase corresponds to MPVS problem and constructs all maximal navigation sequences from the candidate sessions generated at the first phase. The following features related to the navigation sequences are used in this phase:

- **Maximality:** During the execution of phase two, each new sequence which is either constructed by adding a page to an existing sequence or constructed from a single page, is maximal at the beginning. A sequence becomes non-maximal if a new navigation sequence is constructed from it by adding a page to its tail.
- **Degree:** The degree of a sequence indicates how many new sequences can be constructed from it by adding new pages to its tail. Thus, the degree of a sequence is equal to the out-degree of its last page when it is constructed. With the extension by appending a new page, the degree of the current navigation sequence is decreased by one which also makes the extended sequence non-maximal. Moreover, non-maximal sequences must be kept as long as they are extendable, i.e., their degrees are still larger than zero.

Algorithm 1 CSRA

```

1: input: CandidateSessionSet
2: output: MSeqSet
3: global variables: FSeqSet, TSeqSet
4: global variables: MSeqSet, flag
5: procedure MPVS (CandidateSession)
6:   for each WPi in CandidateSession // WPi is i-th web
   page.
7:     flag := FALSE
8:     for each Seqj in TSeqSet
9:       newSeqExtend(Seqj, WPi)
10:    end for each
11:    if flag = FALSE then
12:      newSeqInitialize(WPi)
13:    end if
14:  end for each
15: end procedure
16: procedure CSRA_Phase_2 (CandidateSessionSet)
17:   MSeqSet := {} //Maximal Sequences
18:   for each CandidateSession in CandidateSessionSet
19:     TSeqSet := {} //Temporary Sequences
20:     FSeqSet := {} //Final Sequences
21:     MPVS (CandidateSession)
22:     for each Seqj in TSeqSet
23:       if Seqj.maximal = TRUE then
24:         FSeqSet := FSeqSet ∪ Seqj
25:       end if
26:     end for each
27:     MSeqSet := MSeqSet ∪ FSeqSet
28:   end for each
29: end procedure

```

The following global variables are used in the second phase of C-SRA:

- **FSeqSet:** Maximal navigation sequences with degrees zero generated from current candidate sessions.
- **TSeqSet:** Navigation sequences with degrees greater than zero, it can still contain maximal sessions.
- **MSeqSet:** Collection of all maximal navigation sequences obtained from all candidate sessions.

The details of the second phase of C-SRA are given in Algorithm-1 and Algorithm-2 respectively. In the main procedure, called CSRA_Phase_2, each candidate session of the CandidateSessionSet is processed by calling the procedure MPVS. In this procedure each page in the candidate session is processed from left to right to determine if that page can expand existing navigation sequences or it can initiate a new sequence. At any particular step, if the degree of any maximal sequence decreases to zero, it is automatically added to final sequence set since there is no page remained in the candidate session to expand current sequence. After completing the processing of each page of a candidate session, maximal sequences remaining in the temporary sequences set with out-degrees greater than zero are also added to the final sequence set. Finally, at the end of processing each candidate sessions, the final sequence set obtained from that candidate session is added to the global maximal sequences set.

Table 1 illustrates the execution of the Phase 2 of C-SRA for the candidate session $[P_1, P_{20}, P_{23}, P_{13}, P_{34}]$ corresponding to the site topology in Figure 1. In the table, each column represents the processing single page of the candidate session and navigation sequences are shown together with their degrees and maximality flags (as triples of <sequence: degree: maximality flag>).

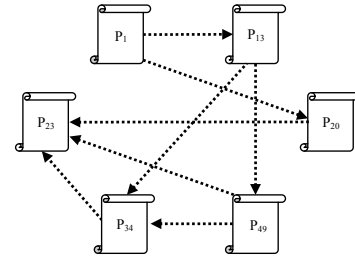


Figure 1: An example web site topology graph

Notice that, in this example we assume that there is no case that violates page stay time constraint. Referring to Table 1, when page P_1 is processed, since it is the first page, a new sequence containing only page P_1 is created. Since it is a new page, it is maximal (T represent the maximality is true), and its out-degree is the out-degree of the page P_1 , which is 2. When the second page, P_{20} , is processed, the sequence $[P_1]$ will be extended and a new sequence $[P_1, P_{20}]$ will be generated. The new Sequence $[P_1, P_{20}]$ will be maximal, but, we will still keep the extended Sequence $[P_1]$ in the temporary sequence set. Although its out-degree is decreased, since it is still greater than zero (which is 1) it can still be extended by using its unused hyperlinks (out-going edges in the graph representation). Moreover, since the sequence $[P_1]$ was extended, its out-degree was decreased to 1

Algorithm 2 Session Functions

```
1: input: CandidateSessionSet
2: output: MSeqSet
3: global variables: FSeqSet, TSeqSet
4: global variables: MSeqSet, flag
5: procedure NewSeqExtend (Seqj, WPi)
6:   linkStatus := Link[LastElement(Seqj), WPi]
7:   timeDiff := TimeDiff(LastElement(Seqj), WPi)
8:   if linkStatus=true and timeDiff < δ then
9:     flag := TRUE
10:    Seqj.degree := Seqj.degree -1
11:    Seqj.maximal := FALSE
12:    NewSeq.degree := WPi.outdegree
13:    NewSeq.maximal := TRUE
14:    NewSeq := Seqj • WPi //Append
15:    if NewSeq.degree = 0 then
16:      FSeqSet := FSeqSet U {NewSeq}
17:    else
18:      TSeqSet := TSeqSet U {NewSeq}
19:    end if
20:    if Seqj.degree = 0 then
21:      TSeqSet := TSeqSet - {Seqj}
22:    end if
23:  end if
24: end procedure
25: procedure NewSeqInitialize (WPi)
26:   NewSeq.degree := WPi.outdegree
27:   NewSeq.maximal := TRUE
28:   NewSeq := [WPi]
29:   if NewSeq.degree = 0 then
30:     FSeqSet := FSeqSet U {NewSeq}
31:   else
32:     TSeqSet := TSeqSet U {NewSeq}
33: end procedure
```

and it becomes non-maximal. The new sequence, $[P_1, P_{20}]$, on the other hand is marked as maximal with the out-degree 1, which is the out-degree of its last page P_{20} . After that, the new sequence, $[P_1, P_{20}, P_{23}]$ is obtained while processing the third page, P_{23} . This sequence has an out-degree 0, thus rather than keeping it in the temporary Sequence set, it is directly moved into the final Sequence set, since it can not be expanded any further. After processing the last two pages of the candidate session (P_{13} and P_{34}), the sequence $[P_1, P_{13}, P_{34}]$ was also generated. Since this sequence has a degree 1, it is placed into the temporary Sequence set. After all pages in the candidate session have been processed, this sequence is also moved into the Final Sequence set due to maximality. As a result, after completing the processing of the candidate session $[P_1, P_{20}, P_{23}, P_{13}, P_{34}]$, the second phase of C-SRA discovers two maximal sequences: $[P_1, P_{20}, P_{23}]$ and $[P_1, P_{13}, P_{34}]$.

3. EXPERIMENTS AND DISCUSSIONS

Sequential pattern discovery is the next phase of the Web Usage Mining after session construction. In this phase, frequent user access patterns are discovered from session sequences. In our experiments, we applied our web usage mining component (C-SRA + Pattern Discovery) to the server logs generated by simulator described in [1]. We have compared our algorithm by replacing C-SRA in the WUM tool

Table 1: Execution of Complete-SRA

Page	P_1	P_{20}
Temp Sequences		$\langle [P_1] : 2 : T \rangle$
Extended Set		$\langle [P_1] : 1 : F \rangle$
New Sequence	$\langle [P_1] : 2 : T \rangle$	$\langle [P_1, P_{20}] : 1 : T \rangle$
Final Set		
Page	P_{23}	P_{13}
Temp Sequences	$\langle [P_1] : 1 : F \rangle$ $\langle [P_1, P_{20}] : 1 : T \rangle$	$\langle [P_1] : 1 : F \rangle$
Extended Set	$\langle [P_1, P_{20}] : 0 : F \rangle$	$\langle [P_1] : 0 : F \rangle$
New Sequence	$\langle [P_1, P_{20}, P_{23}] : 0 : T \rangle$	$\langle [P_1, P_{13}] : 2 : T \rangle$
Final Set	$\langle [P_1, P_{20}, P_{23}] : 0 : T \rangle$	$\langle [P_1, P_{20}, P_{23}] : 0 : T \rangle$
Page	P_{34}	
Temp Sequences	$\langle [P_1, P_{13}] : 2 : T \rangle$	
Extended Set	$\langle [P_1, P_{13}] : 1 : F \rangle$	
New Sequence	$\langle [P_1, P_{13}, P_{34}] : 1 : T \rangle$	
Final Set	$\langle [P_1, P_{20}, P_{23}] : 0 : T \rangle$ $\langle [P_1, P_{13}, P_{34}] : 1 : T \rangle$	

with previous session construction techniques [3](time and navigation oriented techniques) in terms of accuracy metric introduced in [1]. Our experiments show that C-SRA performs at least 20% - 25% better than previous approaches.

4. REFERENCES

- [1] M. A. Bayir, I. H. Toroslu, A. Cosar, and G. Fidan. Smart miner: a new framework for mining large scale web usage data. In *WWW*, pages 161–170, 2009.
- [2] M. A. Bayir, I. H. Toroslu, M. Demirbas, and A. Cosar. Discovering better navigation sequences for the session construction problem. *Data Knowl. Eng.*, 73:58–72, 2012.
- [3] B. Liu, B. Mobasher, and O. Nasraoui. Web usage mining. In *Web Data Mining, Data-Centric Systems and Applications*, pages 527–603. Springer Berlin Heidelberg, 2011.
- [4] B. Mobasher. Data mining for web personalization. In *The Adaptive Web*, pages 90–135, 2007.