PNP: Mining of Profile Navigational Patterns

Hua-Fu Li[*a] and Man-Kwan Shan[**b]
[a]Department of Computer Science and Information Engineering
National Chiao Tung University, Taiwan, R.O.C.
[b]Department of Computer Science
National Cheng Chi University, Taiwan, R.O.C.

ABSTRACT

Web usage mining is a key knowledge discovery research and as such has been well researched. So far, this research has focused mainly on databases containing access log data only. However, many real-world databases contain users profile data and current solutions for this situation are still insufficient. In this paper we have a large database containing of user profile information together with users web-pages navigational patterns. The user profile data includes quantitative attributes, such as salary or age, and categorical attributes, such as sex or marital status. We introduce the concept of profile navigation patterns, which discusses the problem of relating user profile information to navigation behavior. An example of such profile navigation pattern might be " 20% of married people between age 25 and 30 have the similar navigational behavior $\langle (a,c)(c,b)(b,e)(e,a)(a,d) \rangle$ ", where a, b, c, d, e are web pages in a web site. The navigation sequences may contain the generic traversal behavior, e.g. trend to backward moves, cycles etc. The objective of mining profile navigation patterns is to identify browser profile for web personalization. We present PNP, a new algorithm that discovers these profile navigation patterns. Scale-up experiments show that PNP scales linearly with the number of transactions.

Keywords: Web mining, Web usage mining, categorical, quantitative, navigation sequence, profile navigation patterns, attribute, PNP.

1. INTRODUCTION

Web mining can be defined as the discovery and extraction of implicit, useful, interesting knowledge from the World Wide Web. Research of Web Mining can be classifies into three areas: Web structure mining, Web content mining and Web usage mining [5]. Web structure mining tries to discover the model of Web topology [1]. Web content mining describes the automatic search of useful information from available on-line resources [6]. Web usage mining is the discovery of navigation patterns of Web users [5].

Mining Web navigation pattern, mining frequent Web log pieces, is an important Web mining tasks. There are

[*] hfli@csie.nctu.edu.tw; 1001, Ta Hsueh Rd., Hsinchu, Taiwan, 30050, R.O.C.
[**] mkshan@cs.nccu.edu.tw; phone:886-2-29393091#67622;fax:886-2-22341494;64,Chin-nan Rd., Sec. 2,Wenshan, Taipei,Taiwan, 11623, R.O.C.

Data Mining and Knowledge Discovery: Theory, Tools, and Technology IV, Belur V. Dasarathy, Editor,
Proceedings of SPIE Vol. 4730 (2002) © 2002 SPIE · 0277-786X/02/$15.00

several work toward discover various Web navigation patterns from Web logs [3,7,8,9,10,11]. Mining *path traversal patterns* from Web logs has been studied in [3]. IBM use the method proposed in [3] to construct a mining system, *SpeedTracer* [10]. *WebMiner* [5] applies existing data mining techniques, such as association rule and sequential pattern mining, to discover the interesting navigation patterns. *WebLogMinier* [11] use the techniques of OLAP and data mining to discover interesting patterns. The studies of [8,9] propose the Web Utilization Miner *WUM*, a Web log mining system for the discovering of the user traversal patterns. Finally, [7] proposes a novel tree structure for efficient mining of *Web access patterns* (i.e., intra-transaction sequential patterns) from Web logs.

However, previous work has focus on mining navigation patterns from Web logs. There are applications that need to find associations between navigation patterns and user profile at multiple concept levels. For example, Age[20-25]$\wedge$Salary[50K-75K]: <(YAHOO,MLB) (YAHOO,NBA)>may represent that among the people with age between 20 and 25 and salary between 50K and 75K, those typically accessed YAHOO Web pages, then they are likely to access the web pages of MLB, and then accessed from YAHOO to NBA Web pages. Such patterns can be used to enhance topology design, adaptive Web sites, Web personalization and target marketing in e-business.

In this paper, we defined the problem of mining profile navigation pattern and present a new algorithm PNP to efficiently mining profile navigation patterns from large set of pieces of Web log with user profile data. In this method, we first. The scale-up experiments also confirm that the PNP has good linear scalability with the number of profile navigation sequences.

The remaining of the paper is organized as follows. The problem of mining profile navigation pattern is defined in Section 2. Section 3 presents the PNP algorithm. The experimental evaluation is described in Section 4. Section 5 concludes our study.

2. ROBLEM DEFINITION

In this paper, we focus on mining *profile navigation patterns*. In general, a traversal log contains a pair of pages (source, destination) for each link traversed. For the beginning of a new traversal, which is not linked to the previous page, the source field is null. Data preprocessing [4] can be applied to the Web logs, so that the sequences of pairs of logs can be obtained.

Let $E = \{e_1, e_2, \ldots, e_m\}$ be a set of nodes (pages) in the site. A *navigation sequence S* is denoted by $\langle S_1 S_2 \ldots S_n \rangle$, where and S_i is called an *event* of the sequence (i.e., $S = \langle (s_1,d_1)(s_2,d_2)\ldots(s_n,d_n) \rangle$), where $\{s_i, d_i\} \subseteq E$ for $1 \le i \le n$. A navigation sequence $\alpha = \langle s_1 s_2 \ldots s_p \rangle$ is said to *sequence-contain* $\beta = \langle r_1\, r_2 \ldots r_q \rangle$ as a consecutive sequence, denoted as $\alpha \subset \beta$, if and only if there exists an i such that $s_{i+j} = r_j$, for $1 \le j \le p$. For example, $\langle (a,b)(b,d)(d,c) \rangle$ sequence-contains $\langle (a,b)(b,d) \rangle$.

A profile attribute can be either categorical or quantitative. Categorical attributes (such as zip code or martial status) have a limited number of values without ordering. Quantitative attributes (such as salary or age) have an implicit ordering and continuous values within a specified range.

Let D be a database of tuples where each tuple is a set of attribute values, called *profile-items* (or *items*), of the from (item$_i$ = value$_i$) with a navigation sequence $\langle(s_1,d_1)(s_2,d_2)\ldots(s_k,d_k)\rangle$. We assume that each item occurs at most once in a tuple. A profile navigation sequence (also called *p-sequence*) is an expression of the form (p_1, p_2,…, p_n, s), where p_1,…, p_m are items and s is a navigation sequence. We call that a tuple (a_1, a_2,…, a_m: s_j) *matches* a profile navigation sequence P = (b_1, b_2,…, b_n, s_k) if and only if there exist $1 \le i_1 \le i_2 \le \ldots \le i_n \le n$ such that $a_{i_1} \subseteq b_1$, $a_{i_2} \subseteq b_2$, …, $a_{i_n} \subseteq b_{\mathrm{n}}$ and s_j sequence-contains s_k. For example, a three-items tuple ((Age:25), (Married:No), (City:Taipei): $\langle(a,b)$ (b,d) (d,e) (e,a) (a,f) $\rangle$) matches a profile navigation sequence ((Age:25), (City:Taipei): $\langle(b,d)$ (d,e) $(e,a)\rangle$).

The number of tuples in the database pattern-matching profile navigation sequence P is called the *support* of P, denoted as *sup*(P). The *minsup* is the user specified minimum support threshold. A profile navigation sequence P is a *frequent* sequence if *sup*(P) ≥ *minsup*. *Generalization*, as defined in this paper, is the combination of adjacent interval of frequent attribute values. For example, generalization (Age:25), (Age:26) and (Age:27) results in (Age:25-27). A generalized profile navigation sequence (also called *g-sequence*) is an expression of the form (g_1, g_2,…, g_m: s_n). g_i, for $1 \le i \le m$, are items of the form (item = value) or (item: intervel$_i$ - interval$_j$), where interval$_i$ denoted the lower bound for values of the *i*-th interval.

A profile navigation sequence (a_1, a_2,…, a_n, s_p) is *pattern-contained* in another sequence (i.e., either p-sequence or g-sequence) (b_1, b_2,…, b_m, s_q) is there exist integer $i_1 < i_2 < \ldots < i_n$ such that $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}$,…, $a_n \subseteq b_{i_n}$ and $s_p \subseteq s_q$. For example, the sequence ((Age:25-26): $\langle(a,b)(b,d)(d,e)\rangle$) is pattern-contained in ((Age:25-29): $\langle(a,b)(b,d)(d,e)(e,f)\rangle$), since (Age:25-26) ⊆ (Age:25-29) and $\langle(a,b)(b,d)(d,e)\rangle \subset \langle(a,b)(b,d)(d,e)(e,f)\rangle$. In a set of profile navigation sequences, a sequence p is *maximal* if s is not pattern-contained in any other sequences. Each such sequence p is also called a *profile navigation pattern*.

Table 1: Database D_{org} for profile navigation pattern mining.

Record ID	Age	Married	Salary	Navigation Sequence	Count
1	23	NO	50K	<*(a,b)(b,a)(a,c)*>	10
2	25	YES	60K	<*(a,b)(b,d)(d,a)(a,c)*>	6
3	26	YES	70K	<*(a,b)(b,a)(a,c)(c,f)(f,d)*>	8
4	24	NO	50K	<*(b,a)(a,c)(c,e)*>	7
5	30	YES	90K	<*(b,c)(c,d)(d,e)*>	5
6	29	NO	80K	<*(a,c)(c,f)(f,d)(d,e)*>	8

Profile navigation patterns (*minsup*=10 records)	Support
(Married:NO)(Age:23-24),(Salary:50K):<*(a,b)(a,c)*>	17
((Married:NO),(Age:23),(Salary:50K): <*(a,b)(b,a)(a,c)*>)	10
(Married:YES),(Salary:60-70K),(Age:25-26):<*(a,b)(a,c)*>)	14

Figure 1 Example of profile navigation pattern generation.

Example 1. In Table 1 we show a number of sequence recorded in the Web log with profile attributes. Along with each sequence we show the number (count) of users that have followed it. In this figure, we use *a, b, c, d, e, f* for the Web pages. Let the minimum support be 10 records (i.e., *minsup*=10), we can find the profile navigation patterns from this example as shown in Figure 1.

3. PNP: MINING PROFILE NAVIGATION PATTERNS

A method for mining navigation patterns is introduced in this section. Because quantitative attributes typically have a wide range of values from their respective domains, we should partition the values of quantitative attributes into intervals. In this paper, we partition attributes using *equi-width* intervals, where the size of each interval is the same. To simplify our discussion, a running example, which simulates the Example 1, is analyzed as follows.

Notation. We call the number of items in a profile navigation sequence its *size*, and called a profile navigation sequence with size *k* a *k-items sequence*. We assume that C_k be a set of candidate k-items sequences (potentially frequent sequences), where each member of this set has two fields: profile navigation sequences and support counts. Let Fk be the set of frequent k-items sequences. Each member of F_k also has two fields: profile navigation sequences and support count. Note that we also call Pnp_k be a set of frequent profile navigation patterns.

Example2. The first pass of the algorithm simply counts item and event occurrences to determined the frequent items, events. Only the frequent events will be useful in the construction of frequent profile navigation patterns. Let the minimum support be 10 records (i.e., *minsup*=10). Note that since the total number of records in the database is fixed, the support is expressed in an absolute value for simplicity. The mining of the profile navigation patterns proceeds as follows.

The frequent items FI[1] (see Figure 2) and events FE[1] (see Figure 3) can be derived by scanning the database D_{org} (see Example 1), counting support of each item or event if a record matched such an item or event and filtering out those events whose accumulated support count is lower than the minimum support. Note that FE[1] can be used to filter out any item which is not frequent in a record, and to remove the records in D_{org} which contain only non-frequent events. This results in a filtered database D_{fil} of Figure 4.

Since there are four frequent items (i.e., *sup*(Married:NO)=23, *sup*(Married:YES)=19, *sup*(Age:23)=10 and *sup*(Salary:50K)=17) in FI[1], the candidate 2-items profile navigation sequences will be {< (Married:NO),(Age:23-24) >, < (Married:NO), (Salary:50K) >, < (Age:23), (Salary:50K) >,< (Married:YES), (Age:25-26) >, < (Married:YES), (Salary:60-70K) >} associated with event set {*(a,b),(b,a),(a,c),(c,f),(f,d),(d,e)*}. The frequent 2-items profile navigation sequences {((Married:NO),(Age:23-24):<*(a,b)(a,c)*>), ((Married:NO),(Salary:50K):<*(a,b)(a,c)*>), ((Age:23),(Salary:50K): <*(a,b)(b,a)(a,c)*>), ((Married:YES),(Age:25-26)): <*(a,b)(a,c)*>}), (((Married:YES),(Salary:60-70K):<*(a,b)(a,c)*>)) can be determined by scanning D_{org} once again. The results are shown in Figure 5. In this pass (second database D_{org} scan), the filtered database D_{fil} can also be determined. Next, C3 are computed in a *condition-join* process (in next subsection). Consequently, the frequent 3-items profile navigation patterns, as shown in Figure 6, are ((Married:NO),(Age:23-24),(Salary:50K): <(a,b)(a,c)>), ((Married:NO),(Age:23),(Salary:50K): <(a,b)(b,a)(a,c)>), ((Married:YES),(Salary:60-70K),(Age:25-26): <(a,b)(a,c)>)) by mining D_{fil}. The mining process terminates since there is no future candidate generation. The above discussion leads the algorithm PNP (see Figure 7) for mining useful profile navigation patterns.

Item	Attribute					
Married	**NO**			**YES**		
Age (interval range:1)	**23**	24	29	25	26	30
Salary (interval range:10)	**50**	**50**	80	60	70	90
Count	**10**	7	8	6	8	5

Figure 2. Frequent items FI[1] for Example 1.

Events	Support
(a,b)	24
(b,a)	25
(a,c)	39
(c,f)	16
(f,d)	16
(d,e)	13

Figure 3. Frequent events FE[1] of Example 1.

Record ID	Age	Married	Salary	Navigation Sequence	Count
1	23	NO	50K	*<(a,b)(b,a)(a,c)>*	10
2	25	YES	60K	*<(a,b)(a,c)>*	6
3	26	YES	70K	*<(a,b)(b,a)(a,c)(c,f)(f,d)>*	8
4	24	NO	50K	*<(b,a)(a,c)>*	7
5	30	YES	90K	*<(d,e)>*	5
6	29	NO	80K	*<(a,c)(c,f)(f,d)(d,e)>*	8

Figure 4. Filter database D_{fil}.

Candidate Generation. The gen-candidate function takes as argument F_{k-1}, the set of frequent (k-1)-items sequences. It returns a superset of the set of all frequent k-items sequences. The function work as follows. First, in the *candidate-join* step, we join F_{k-1} with F_{k-1}:

```
Insert into $C_k$
Select $p.\text{item}_1, p.\text{item}_2, \ldots, p.\text{item}_{k-1}, q.\text{item}_{k-1}, p.\text{event}_1, p.\text{event}_2, \ldots, p.\text{event}_j$
From $F_{k-1}\ p,\ F_{k-1}\ q$
Where $p.\text{item}_1 = q.\text{item}_1, \ldots, p.\text{item}_{k-2} = q.\text{item}_{k-2}, p.\text{item}_{k-1} \neq q.\text{item}_{k-1}$,
      $p.\text{event}_1 \subseteq q.\text{event}_{i_1}, p.\text{event}_2 \subseteq q.\text{event}_{i_2}, \ldots, p.\text{event}_j \subseteq q.\text{event}_{i_j}$
```

Next, we use the *apriori* property, i.e., *all sub-patterns of a frequent pattern must be frequent*, to prune the potential non-frequent k-items sequences. For example, let F_2 be {((A)(B):<*(a,b)(a,c)*>), ((A)(C):<*(ac)*>), ((A)(D):<*(b,c)*>), (B)(C):<*(a,c)(c,d)*>}}, where A,B,C and D are items. After the candidate-join step, C_3 will be {((A)(B)(C):<(a,c)>), ((A)(B)(D): <NULL>), ((A)(C)(D): <NULL>)}. We will delete the 3-items sequences {((A)(B)(D): <NULL>), ((A) (C) (D):<NULL>)} because these 2-items sequences are not in F_2. We will then be left with only {((A)(B)(C):<(a,c)>)} in F_3.

C_2	Event set
(Married:NO),(Age:23-24) (Married:NO),(Salary:50K) (Age:23),(Salary:50K) (Married:YES),(Age:25-26) (Married:YES),(Salary:60-70K)	{*(a,b),(b,a),(a,c),(c,f),(f,d),(d,e)*}

F_2	Support
(Married:NO)(Age:23-24):<*(a,b)(a,c)*>	17
((Married:NO),(Salary:50K):<*(a,b)(a,c)*>)	17
((Age:23),(Salary:50K): <*(a,b)(b,a)(a,c)*>)	10
((Married:YES),(Age:25-26)): <*(a,b)(a,c)*>)	14
((Married:YES),(Salary:60-70K):<*(a,b)(a,c)*>)	14

Figure 5:Example (C_2 and F_2 generation).

Minsup=10

F_3	Support
(Married:NO)(Age:23-24),(Salary:50K):<*(a,b)(a,c)*>	17
((Married:NO),(Age:23),(Salary:50K): <*(a,b)(b,a)(a,c)*>)	10
(Married:YES),(Salary:60-70K),(Age:25-26):<*(a,b)(a,c)*>)	14

Figure 6.Example (C_3 and F_3 generation)

ALGORITHM PNP: *Find profile navigation patterns from a large database.*

INPUT: (1) A database D_{org}, in the format of (*RecordID*, *pitems*, *S*), and (2) the minimum support threshold (*minsup*).

OUTPUT: Profile navigation patterns *FPnp*.

METHOD:

```
For (k=1; FI[k]$\neq\emptyset$; K++) do {
    If (k=1) then {
        FI[1]=frequent-ones($D_{org}$, minsup);
        FE[1]=frequent-ones($D_{org}$, minsup);
        Dfil=gen-filtered-database($D_{org}$ ,F E[1]);}
    for(k=1;FI[k-1]$\neq\emptyset$; K++) do {
        $C_k$=gen-gandidate($F_{k-1}$);        // $F_0$=FI[1] candidate-join FE[1]
        For each record r$\in D_{fil}$ do{
            $C_t$=get-subset($C_k$,t);    //using hash-tree structure
            For each candidate c$\in D_{fil}$ do
                c.support++;}
        $F_k$={c$\in$Ck|c.support $\geq$ minsup}}
    FPnp=$U_k$ $F_k$;
}
```

Figure 7: Algorithm PNP

Notice that in the mining process of PNP, D_{org} need to be scanned twice (one pass to determined the FI[1] and FE[1], and another ones to generated F_1 and the filtered database D_{fil}), whereas D_{fil} needs to be scanned t times where $t = k$-2, and k is the size of the maximum k-items sequences.

4. EXPERIMENTAL RESULTS

The performance of the proposed algorithm PNP was measured using synthetic data. The method used by this paper for generating synthetic web log data is based on the principle introduced in [3]. Profile information is generated randomly so that values are distributed evenly in every attribute interval. All experimental results were conducted on a 350-MHz Pentium II PC machine with 128 megabytes main memory, running Microsoft Windows 2000.

First, a Web *tree* is generated to simulate the topology of a Web site. This Web tree consists of two kinds of nodes, namely internal nodes and leaf nodes. The *fan-out* of each internal node is determined by a uniform distribution between 4 and 7. The height of a sub-tree whose sub-root is a child node of the root is determined by a Poisson distribution with mean μ_h. The height of a sub-tree whose root is the child of an internal node N_i is determined by the Poisson distribution with mean equal to the maximal height of N_i.

The length of each web traversal walk is determined by the Poisson distribution with mean $|P|$. The next move of each internal node is assigned with a weight. An internal node with n child nodes is assigned with a weight p_0, which is $1/(n+1)$. The probability of moving to each child node, p_i, is determined by an exponential distribution with unit mean. The sum of the weights for all child nods is normalized to $1-p_0$. If an internal node has a internal jump with a weight p_j, then p_0 is changed to $p_0(1-p_j)$ and all the corresponding probability for each child node is changed to $p_i(1-p_j)$. When a user moves to a leaf node, the next node would be either its parent node with a weight 0.25 or any internal node with probability 0.75.

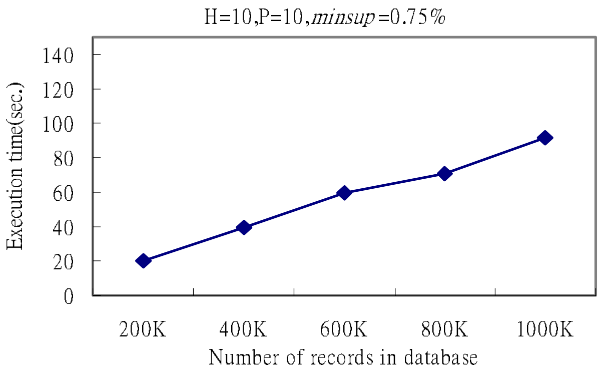

Figure 8: Number of records scale-up.

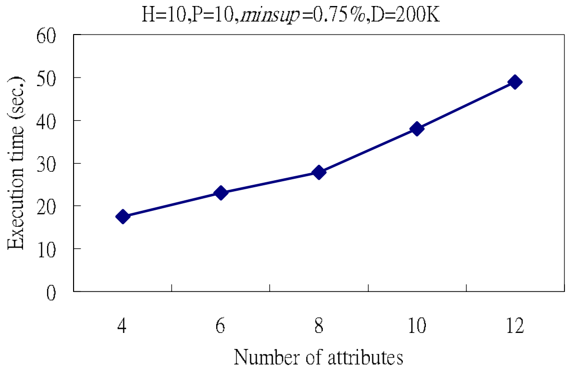

Figure 9: Scalability over attributes.

Figure 8 shows the scalability of the algorithm PNP over the number of records in the database. The database size ranges from 200,000 to 1,000,000, and the support is set to 0.75%. Figure 9 shows the scalabili ty of the algorithm PNP over the number of profile attributes. The minimum support threshold is set to 0.75%. As the attributes increase, the execution time of PNP goes up. However, PNP is more scalable.

5. CONCLUSIONS

We have extend the scope of the study of Web usage mining from the level of discovering Web access log sequences to various level of profile attribute with navigation sequences and present an algorithm, PNP, for discovering all these patterns in a large database. Profile navigational patterns are interesting and useful in practice since people are often interested in Web navigation sequence associated with different profile attributes. The experimental results shows that the execution time of PNP increases linearly as the database size increases.

Extension of the methods for mining Web navigation patterns to profile navigation patterns poses many new issues for further investigation. Mining profile (or multiple-level) sequential patterns and meta-query guided mining of profile navigation patterns are interesting topics for future study.

REFERENCES

1. Chakrabarti, S., Dom, B. E., Gibson, D., Kleingerg, J., Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. S. Mining the Link Structure of the World Wide Web. *IEEE Computer*, 32(8), 60-67, 1999.
2. Chen, M. S., Han, J., and Yu, P. S. Data Mining: An Overview from Database Perspective. *IEEE Transactions on Knowledge and Data Engineering,* 8(6), 866-883, 1996.
3. Chen, M. S., Park, J. S., and Yu, P. S. Efficient Data Mining for Path Traversal Patterns. *IEEE Transactions on Knowledge and Data Engineering*, 10(2), 209-221, 1998.
4. Cooley, R., Mobasher, B., and Srivastava, J. Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns. *Technical Report TR 97-021*, Dept. of Computer Science, Univ. of Minnesota, Minneapolis, 1997.
5. Cooley, R., Mobasher, B., and Srivastava, J. Web Mining：Information and Pattern Discovery on the World Wide Web. *Proceedings of the 1997 IEEE International Conference on Tools with Artificial Intelligence*, 558-567,1997.
6. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S. Learning to Extract Symbolic Knowledge from the World Wide Web. *Proceedings of the 1998 National Conference on Artificial Intelligence*, 509-516, 1998.
7. Pei, J., Han, J., Mortazavi-Asl, B., and Zhu, H. Mining Access Pattern efficiently from Web logs. *Proceedings of the 2000 Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 396-407, 2000.
8. Spiliopoulou, M. and Faulstich, L. C. WUM: A Tool for Web Utilization Analysis. *EDBT Workshop WebDB'98*, Valencia, Spain, 184-203, 1998.
9. Spiliopoulou, M., Faulstich, L. C., and K. Winkler, K. A Data Miner Analyzing the Navigational Behaviour of Web Users. *Proceedings of the 1999 Workshop on Machine Learning in User Modelling of the ACAI'99 International Conference*, 113-126, 1999.
10. Wu, K. L., Yu, P. S., and Ballman, A. SpeedTracer: A Web Usage Mining and Analysis Tool. *IBM System Journal*,

37(1), 89-105, 1998.

11. Zaïane, O. R., Xin, M., and Han, J. Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. *Proceedings of the 1998 Advances in Digital Libraries Conference*, 19-29, 1998.