A Top-Down Fuzzy Cross-Level Web-Mining Approach*

Wei-Shuo Lo
Institute of Information Engineering
I-Shou University
Kaohsiung, Taiwan, ROC
x3168@mail.meiho.edu.tw

Tzung-Pei Hong
Dept. of Electrical Engineering
National University of Kaohsiung
Kaohsiung,Taiwan, ROC
tphong@nuk.edu.tw

Shyue-Liang Wang
Dept. of Computer Science
New York Institute of Technology
New York, USA
slwang@nyit.edu

Abstract - *Web mining of browsing patterns including simple sequential patterns and sequential patterns with browsing times has been studied recently. However, most of these works focus on mining browsing patterns of web pages directly. In this work, we introduce the problem of mining browsing patterns on cross-levels of a taxonomy comprised of web pages. In addition, browsing time is considered and processed using fuzzy set concepts to form linguistic terms. The proposed algorithm thus discovers cross-level relevant browsing behavior from linguistic data and promotes the discovery of coarsen granularity of web browsing patterns.*

Keywords: Top-down, fuzzy set, cross-levels, Web mining, sequential patterns

1 Introduction

Web mining can be viewed as the use of data mining techniques to automatically retrieve, extract and evaluate information for knowledge discovery from web documents and services [10]. It has been studied extensively in recent years due to practical applications of extracting useful knowledge from inhomogeneous data sources in the World Wide Web. Web mining can be divided into three classes: web content mining, web structure mining and web usage mining [7]. In the past, several web-mining approaches for finding user access patterns and user interesting information from the World Wide Web were proposed [3-6,14]. Chen and Sycara proposed the WebMate system to keep track of user interests from the contents of the web pages browsed. It can thus help users to easily search data from World Wide Web [4]. Chen et. al. mined path-traversal patterns by first finding the maximal forward references from log data and then obtaining the large reference sequences according to the occurring numbers of the maximal forward references [3]. Cohen et. al. sampled only portions of the server logs to extract user access patterns, which were then grouped as volumes [5]. Files in a volume could then be fetched together to increase the efficiency of a web server. Spliliopoulou et. al. [14] proposed the Web Utilization Miner to discover interesting navigation patterns. Many efficient algorithms for discovering maximal sequential patterns have been proposed [1,2,12,13,15]. In application to web browsing patterns, techniques for mining simple sequential browsing patterns and sequential patterns with browsing times have been proposed [4,5,6,7,10,14]. However, most of these works focus on mining browsing patterns of web pages directly. In this work, we introduce the problem of mining browsing patterns on cross levels of a taxonomy comprised of web pages. In addition, browsing time is considered and processed using fuzzy set concepts to form linguistic terms. The proposed algorithm thus discovers cross-level relevant browsing behavior from linguistic data and promotes the discovery of coarsen granularity of web browsing patterns. The rest of our paper is organized as follows. Notation used in this paper is given in section 2. Section 3 presents the mining algorithm of fuzzy cross-level browsing patterns. Section 4 gives an example to illustrate the feasibility of the proposed algorithm. A conclusion is given at the end of the paper.

2 Notation

The following notation is used in our proposed algorithm:

n: the total number of log data;
m: the total number of files in the log data;
c: the total number of clients in the log data;
n_i: the number of log data from the i-th client, $1 \le i \le c$;
D_i: the browsing sequence of the i-th client, $1 \le i \le c$;
D_{id}: the d-th log transaction in D_i, $1 \le d \le n_i$;
F^g: the g-th file, $1 \le g \le m$;
$R^g k$: the k-th fuzzy region of F^g, $1 \le k \le |F^g|$, where $|F^g|$ is the number of fuzzy regions for F^g;
v_{id}^g: the browsing duration of file F^g in D_{id};
f_{id}^g: the fuzzy set converted from v_{id}^g;

* 0-7803-7952-7/03/$17.00 © 2003 IEEE.

f_{id}^{gk}: the membership value of v_{id}^{g} in region R^{gk};

f_i^{gk}: the membership value of region R^{gk} in the i-th client sequence D_i;

$count^{gk}$: the scalar cardinality of region R^{gk};

$max\text{-}count^{g}$: the maximum count value among $count^{gk}$ values;

$max\text{-}R^{g}$: the fuzzy region of file I^g with $max\text{-}count^{g}$;

α: the predefined minimum support value;

λ: the predefined minimum confidence value;

C_r: the set of candidate sequences with r files;

L_r: the set of large sequences with r files.

3 The mining algorithm of fuzzy cross-level web browsing patterns

This section describes the proposed data-mining algorithm of fuzzy cross-level web browsing patterns. The log data are first extracted, sorted, and reorganized into users' browsing sequences. The browsing time of each web page is then transformed into linguistic terms using fuzzy sets. Browsing patterns with no ancestor rules or patterns with interest support greater than the predefined threshold will be output as the interesting web browsing patterns. The detail of the proposed web-mining algorithm is described as follows.

INPUT: A server log, a predefined taxonomy of web pages, a set of membership functions, a predefined minimum support value α, and a predefined interest support threshold R.

OUTPUT: A set of interesting fuzzy cross-level browsing patterns.

STEP 1: Select the web pages with file names including .asp, .htm, .html, .jva .cgi and closing connection from the log data; keep only the fields *date, time, client-ip* and *file-name*. Denote the resulting log data as *D*.

STEP 2: Encode each web page file name using a sequence of number and the symbol "*", with the *t*-th number representing the branch number of a certain web page on level t.

STEP 3: Form a browsing sequence D_j for each client c_j by sequentially listing his/her n_j tuples (web page, duration), where n_j is the number of web page browsed by client c_j. Denote the *d-th* tuple in D_j as D_{jd}.

STEP 4: Set k=1, where k is used to store the level number being processed.

STEP 5: Re-encode the web page file names by retaining the first k digits and replacing the rest of digits by "*" in each browsing sequence

STEP 6: Transform the time duration v_{id}^{g} of the file name I^g appearing in D_{id} into a fuzzy set f_{id}^{g} represented as

$$\left(\frac{f_{id}^{g_1}}{R^{g_1}} + \frac{f_{id}^{g_2}}{R^{g_2}} + \cdots + \frac{f_{id}^{g_l}}{R^{g_l}} \right)$$

using the given membership functions, where I^g is the g-th file name, R^{gk} is the k-th fuzzy region of item I^g, f_{id}^{gk} is v_{id}^{g}'s fuzzy membership value in region R^{gk}, and l is the number of fuzzy regions for Ig.

STEP 7: Find the membership value f_i^{gk} of each region R^{gk} in each browsing sequence D_i as

$$f_i^{gk} = \underset{d=1}{\overset{|D_i|}{MAX}} f_{id}^{gk},$$

where $|D_i|$ is the number of tuples in D_i.

STEP 8: Calculate the scalar cardinality of each region R^{gk} as: $count^{gk} = \sum_{i=1}^{c} f_i^{gk}$,

where c is the number of browsing sequences.

STEP 9: Find $max\text{-}count^{g} = \underset{k=1}{\overset{l}{MAX}} (count^{gk})$, where $1 \le g \le m$, m is the number of files in the log data, and l is the number of regions for file I^g. Let $max\text{-}R^g$ be the region with $max\text{-}count^g$ for file I^g. $max\text{-}R^g$ will be used to represent the fuzzy characteristic of file I^g in later mining processes.

STEP10: Check whether the value $max\text{-}count^g$ of a region $max\text{-}R^g$, g = 1 to m, is larger than or equal to the predefined minimum support value α. If a region $max\text{-}R^g$ is equal to or greater than α, put it in the set of large l-sequences (L_1). That is,

$$L_1 = \{ max\text{-}R^g \mid max\text{-}count^g \ge \alpha, 1 \le g \le m \}$$

STEP11: If L_1 is null, then exit the algorithm; otherwise, do the next step.

STEP12: Set r=1, where r is used to represent the length of sequential patterns currently kept.

STEP13: Generate the candidate sequence C_{r+1} from L_r in a way similar to that in the aprioriall algorithm [1]. The algorithm first joins L_r and L_r, under the condition that r-1 items in the two itemsets are the same and with the same orders.

STEP14: Do the following substeps for each newly formed $(r+1)$-sequence s with contents $(s_1, s_2, \ldots, s_{r+1})$ in C_{r+1}:

(a) Calculate the fuzzy value f_i^s of s in each browsing sequence D_i as:

$f_i^s = \underset{k=1}{\overset{r+1}{Min}} f_i^{s_k}$, where region s_k must appear after region s_{k-1} in D_i. If two or more same subsequences exist in D_i, then f_i^s is the maximum fuzzy value among those of these subsequences.

(b) Calculate the scalar cardinality of s as:

$$count^s = \sum_{i=1}^{c} f_i^s ,$$

where c is number of browsing sequences.

(c) If $count^s$ is larger than or equal to the predefined minimum support value α, put s in L_{r+1}

STEP15: IF L_{r+1} is null, then do the next step; otherwise, set r=r+1 and repeat STEP 13 to 15.

STEP16: Output the browsing patterns without ancestor patterns (by replacing the web pages in a pattern with their ancestors in the taxonomy) to users as interesting patterns.

STEP17: For each remaining pattern s (representing $s_1 \rightarrow s_2 \rightarrow \ldots \rightarrow s_r \rightarrow s_{(r+1)}$), find the closest ancestor t (representing $t_1 \rightarrow t_2 \rightarrow \ldots \rightarrow t_r \rightarrow t_{(r+1)}$), and calculate the support interest measure $I_{support}(s)$ of s as:

$$I_{support}(s) = \frac{count_{\ s}}{\frac{\prod_{k=1}^{r+1} count_{\ sk}}{\prod_{k=1}^{r+1} count_{\ lk}} \times count_{\ t}} .$$

Output the patterns with their support interest measure larger than or equal to the predefined interest threshold R to users as interesting patterns.

4 An example

The example shows how the proposed algorithm can be used to discover the fuzzy sequential patterns from the web browsing log data shown in Table 1. The membership functions for the browsing duration on a web page are shown on Figure 1. The browsing duration is divided into three fuzzy regions: Short, Middle, and Long. In addition, the predefined taxonomy for web pages are shown in Figure 2. The predefined minimum support α and interest support threshold R are set at 2 and 1.5 respectively. The proposed data-mining algorithm proceeds as follows.

Table 1: A part of log data used in the example

Time	Client-IP	File-name
05:39:56	140.117.72.1	News.htm
05:40:08	140.117.72.1	Exective.htm
05:40:26	140.117.72.1	University.htm
.....	.....	.....
05:53:33	140.117.72.6	Closing connection

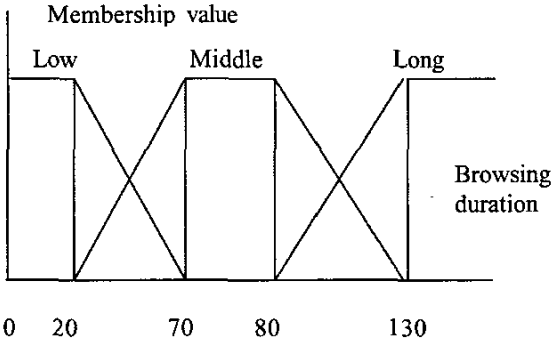

Figure 1: The membership functions used in this example

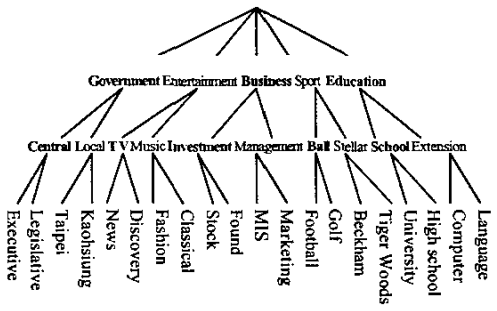

Figure 2: The predefined taxonomy used in this example

STEP 1: Select the web pages with file names including .asp, .htm, .html, .jva .cgi and closing connection from Table 1.

STEP 2: Each file name is encoded using the predefined taxonomy shown in Figure 2. Results are show in Table 2.

Table 2: Codes of file names

Code	File name	Code	File name
111	Executive.htm	1**	Government.htm
211	News.htm	2**	Entertainment.htm
212	Discovery.htm	3**	Business.htm
313	Stock.htm	4**	Sport.htm
321	Found.htm	5**	Education.htm
411	Football.htm	11*	Central.htm
412	Golf.htm	12*	Local.htm
421	Tiger Woods:htm	21*	TV.htm
511	University.htm	22*	Music.htm
512	Highshool.htm	31*	Investment.htm
		32*	Management.htm

		41*	Ball.htm
		42*	Stellar.htm
		51*	School.htm
		52*	Extension.htm

STEP 3: The web pages browsed by each client are listed as a browsing sequence. Each tuple is represented as (web page, duration), as shown in Table 3. .

Table 3: The browsing sequences with their duration

C-ID	Browsing sequences
1	(211, 30), (511, 42), (412, 98), (313, 91)
2	(412, 62), (211, 31), (421, 102)
3	(111, 92), (411, 89)
4	(412, 70), (211, 29), (512, 118), (212, 11), (321, 42)
5	(412, 75), (211, 29), (313, 74)
6	(421, 80), (313, 61), (511, 122), (212, 17)

STEP 4: k is initially set at 1, where k is used to store the level number being processed.

STEP 5: The re-encoded browsing sequences for level k are shown in Table 4.

Table 4: The re-encoded browsing sequences for level 1-3.

C-ID	Browsing sequences (form level 1 to 3)
1	[(211, 30), (21*, 30), (2**, 30)], [(511, 42), (51*, 42), (5**, 42)], [(412, 98), (41*, 98), (4**, 98)], [(313, 91), (31*, 91), (3**, 91)]
2	[(412, 62), (41*, 62), (4**, 62)], [(211, 31), (21*, 31), (2**, 31)], [(421, 102), (42*, 102), (4**, 102)]
3	[(111, 92), (11*, 92), (1**, 92)], [(411, 89), (41*, 89), (4**, 89)]
4	[(412, 70), (41*, 70), (4**, 64)] [(211, 29), (21*, 29), (2**, 29)] [(512, 118), (51*, 118), (5**, 118)] [(212, 11), (21*, 11), (2**, 11)] [(321, 42), (32*, 42), (3**, 42)]
5	[(412, 75), (41*, 75), (4**, 64),], [(211, 29), (21*, 29), (2**, 29)], [(313, 74), (31*, 74), (3**, 74)]
6	[(421, 80), (42*, 80), (4**, 80)], [(313, 61), (31*, 61), (3**, 61)], [(511, 122), (51*, 122), (5**, 122)], [(212, 17), (21*, 17), (2**, 17)]

STEP 6: The time duration of each file in each browsing sequence is represented as fuzzy set. Take the web page 4** in the first browsing sequence as an example. The time duration "30" of file 2** is converted into the fuzzy set (0./Short + 0.6/Middle + 0.4/Long) by the given membership functions (Figure 1). This transformation is repeated for the other files and browsing sequences.

STEP 7: The membership value of each region in each browsing sequence is found. Take the region 4**.Middle for client 2 as an example. Its membership vale is max (0.8, 0.0, 0.6) = 0.8. The membership values of the other regions can be calculated similarly.

STEP 8: The scalar cardinality of each fuzzy region in all the browsing sequences is calculated as the count value. Take the fuzzy region 4**.Middle as an example. Its scalar cardinality = (0.6+0.8+0.8+1+1+1) = 5.2. This step is repeated for the other regions, and the results are shown in Table 5.

Table 5: The counts of fuzzy regions for level k=1

Item	Region	Cid-1	Cid-2	Cid-3	Cid-4	Cid-5	Cid-6	Total count
4**	Short		0.2					0
	Middle	0.6	0.8	0.8	1	1	1	5.2
	Long	0.4	0.4	0.2				1
41*	Short		0.2					0.2
	Middle	0.6	0.8	0.8	1	1		4.2
	Long	0.4		0.2				0.6
42*	Short							0
	Middle		0.6					1.6
	Long		0.4					0.4
411	Short							0
	Middle			0.8				0.8
	Long			0.2				0.2
412	Short		0.2					0.2
	Middle	0.6	0.8		1	1		3.4
	Long	0.4						0.4
421	Short							0
	Middle		0.6				1	1.6
	Long		0.4					0.4

STEP 9: The fuzzy region with the highest count among the three possible regions for each file is selected. Take file 4** as an example. Its count is 0.0 for "Short", 5.2 for "Middle", and 1 for "Long", since the count for Middle is the highest among the three counts, the region Middle is thus used to represent the file 4** in later mining process.

STEP10: The counts of the regions selected in STEP 9 are checked against the predefined minimum support value α. Assuming that α is set at 2

in this example. Since the count value of 2**.Short, 3**.Middle, 4**.Middle, 21*.Short, 31*.Middle, 41*.Middle, 211.Short, 412.Middle are larger than 2, these regions are put in L_1.

STEP11: Since L_1 is not null, the next step is calculated.

STEP12: Set r = 1, where r is used to represent the length of sequential patterns currently kept.

STEP13: The candidate 2-sequence C_2 is generated from L_1 as follows: (2**.Short, 2**.Short), (2**.Short, 3**.Middle), (3**.Middle, 2**.Short),...., (3**.Middle, 4**. Middle), (4**. Middle, 3**.Middle), (4**. Middle, 4**. Middle). The results are shown in Table 6. No 1 is denote 2-sequence (2**.Short, 2**.Short)

Table 6: The candidate 2-sequence C_2 in this example

	2** Short	3** Middle	4** Middle	21* Short	31* Middle	41* Middle	211 Short	412 Middle
2^{**}_{Short}	1	2	3	4	5	6	7	8
3^{**}_{Middle}	9	10	11	12	13	14	15	16
4^{**}_{Middle}	17	18	19	20	21	22	23	24
21^{*}_{Short}	25	26	27	28	29	30	31	32
31^{*}_{Middle}	33	34	35	36	37	38	39	40
41^{*}_{Middle}	41	42	43	44	45	46	47	48
211_{Short}	49	50	51	52	53	54	55	56
412_{Middle}	57	58	59	60	61	62	63	64

STEP14: The following substeps are done for each newly formed candidate 2-sequence in C_2:

(a) The fuzzy membership value of each candidate 2-sequence in each browsing sequence is calculated. Here, the minimum operator is used for the intersection. Take the sequence (2**.Short, 3**.Middle) as an example. Its membership value in the fourth browsing sequence is calculated as: max[min(0.8, 0.4), min(1.0, 0.4)] = 0.4. There are two subsequences of (2**.Short, 3**.Middle) in that browsing sequence. The results for sequence (2**.Short, 3**.Middle) in all the browsing sequences are shown in Table 7.

Table 7: The membership values for sequence (2**.Short, 3**.Middle)

C-ID	Membership value
1	0.8
2	0.0
3	0.0
4	0.4
5	0.8
6	0.0

(b) The scalar cardinality (count) of each candidate 2-sequence in C_2 is calculated. Results for this example are shown in Table 8.

Table 8: The fuzzy counts of the candidate 2-sequence

No	Cid-1	Cid-2	Cid-3	Cid-4	Cid-5	Cid-6	Total count
1				0.8			0.8
2	0.8			0.4	0.8		2
3	0.6	0.6					1.2
4				0.8			0.8
5	0.8				0.8		1.6
6	0.6						0.6
7							0
8	0.6						0.6

(c) Since only the counts of 2-sequences (2**.Short, 3**.Middle), (4**.Middle, 2**.Short), (4**.Middle, 3**.Middle), (41*.Middle, 31*.Middle), (4**.Middle, 21*.Short), (4**.Middle, 31*.Middle), (21*.Short, 3**.Middle), (41*.Middle, 3**.Middle), (211.Short, 3**.Middle), (412.Middle, 3**.Middle) and (412.Middle, 31*.Middle) are larger than the predefined minimum support value 2, they are thus kept in L_2.

STEP15: Since L_2 is not null, set r=r+1=2. Steps 13-15 are then repeated to find L_3. C_3 is first generated from L_2, and the sequence (4**.Middle, 2**.Short, 3**.Middle) and (4**.Middle, 21*.Short, 3**.Middle)is generated. Since its count is 0.4, smaller than 2.0, it is thus not put in L_3. L_3 is an empty set.

STEP16: The browsing patterns discovered as follows: However, only two browsing patterns (41*.Middle → 3**.Middle and 412.Middle → 31*.Middle), they do have ancestor patterns. 2**.Short → 3**.Middle, 4**.Middle → 2**.Short, 4**.Middle → 3**.Middle, 41*.Middle → 31*.Middle, 4**.Middle → 21*.Short, 4**.Middle → 31*.Middle, 21*.Short → 3**.Middle, 41*.Middle → 3**.Middle, 211.Short → 3**.Middle, 412.Middle → 3**.Middle, 412.Middle → 31*.Middle

STEP17: For the eleventh browsing patterns in STEP 16, their support interest measures are:

$$I_{support}(412.\ Middle \rightarrow 31^{*}.Middle) = \frac{412^{*}.Middle \cap 31^{*}.Middle}{\frac{412Middle}{41^{*}Middle} \times \frac{31^{*}Middle}{3^{**}Middle} \times 412Middle \cap 31^{*}Middle}$$

=1.25

These values are smaller than the predefined interest support threshold 1.5.

They are not considered as interesting browsing patterns. In this example, the ten browsing patterns (2**.Short, 3**.Middle), (4**.Middle, 2**.Short), (4**.Middle, 3**.Middle), (41*.Middle, 31*.Middle), (4**.Middle, 21*.Short), (4**.Middle, 31*.Middle), (21*.Short, 3**.Middle), (41*.Middle, 3**.Middle), (211.Short, 3**.Middle) and (412.Middle, 3**.Middle) are output as meta-knowledge concerning the given log data.

5 Conclusion

In this work, we have proposed a novel web-mining algorithm that can process web server logs to discover fuzzy cross-level web browsing patterns. The duration time of a web page is considered and processed using fuzzy set concepts to form linguistic terms. The adoptions of linguistic terms to express the discovered patterns are more natural and understandable for human begins. In addition, the inclusion of concept hierarchy (taxonomy) of web pages produces browsing patterns of different granularity. This allows the views of users' browsing behavior from various levels of perspectives.

Acknowledgement

This work was partially supported by National Science Council of the Republic of China, under grant number NSC-91-2213-E-214-019.

References

[1] R. Agrawal, and R. Srikant, "Mining Sequential Patterns", Proc. of the 11th International Conference on Data Engineering, pp. 3-14, 1995.

[2] N. Chen, A, Chen, "Discovery of Multiple-Level Sequential Patterns from Large Database", Proc. of the International Symposium on Future Software Technology, Nanjing, China, pp. 169-174, 1999.

[3] M.S. Chen, J.S. Park and P.S. Yu, "Efficient Data Mining for Path Traversal Patterns", IEEE Transactions on Knowledge and Data Engineering, Vol. 10, pp. 209-221, 1998.

[4] L. Chen, K. Sycara, "WebMate: A Personal Agent for Browsing and Searching", The Second International Conference on Autonomous Agents, ACM, 1998.

[5] E. Cohen, B. Krishnamurthy and J. Rexford, "Efficient Algorithms for Predicting Requests to Web Servers", The Eighteenth IEEE Annual Joint Conference on Computer and Communications Societies, Vol. 1, pp. 284-293, 1999.

[6] R. Cooley, B. Mobasher and J. Srivastava, "Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns ", Knowledge and Data Engineering Exchange Workshop, pp. 2-9, 1997.

[7] R. Cosala, H. Blockleel, "Web Mining Research: A Survey", ACM SIGKDD, Vol. 2, Issue 1, pp. 1-15, 2000.

[8] H. Mannila and H. Toivonen, "Discovering Generalized Episodes Using Minimal Occurrences", Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 146-151, 1996.

[9] T. Oates, et al, "A Family of Algorithms for Finding Temporal Structure in Data", Proc. of the 6th International Workshop on AI and Statistics, pp. 371-378, Mar 1997.

[10] S.K. Pal, V. Talwar, and P. Mitra, "Web Mining in Soft Computing Framework:Relevance, State of the Art and Future Directions", to appear in IEEE Transactions Neural Network, 2002.

[11] J. Pei, J.W. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal and M.C. Hsu, "Prefixspan: Mining Sequential Patterns by Prefix-Projected Growth", Proc. of the 17th IEEE International Conference on Data Engineering, Heidelberg, Germany, April 2001.

[12] G. Piategsky-Shapiro, "Discovery, Analysis and Presentation of Strong Rules", Knowledge Discovery in Databases, AAAI/MIT press, pp. 229-248, 1991.

[13] H. Pinto, "Multiple-Dimensional Sequential Patterns Mining", University of Lethebridge, Alberta, Canada, Master Thesis, April, 2001.

[14] M. Spliliopoulou, L.C. Faulstich, "WUM: A Web Utilization miner", Workshop on the Web and Data Base (WEBKDD), pp. 109-115, 1998.

[15] R. Srikant and R. Agrawal,"Mining Sequential Patterns: Generalizations and Performance Improvements", Proc. of the 5th International Conference on Extending Database Technology, pp. 3-17, March 1996.