American Journal of Applied Sciences 5 (12): 1611-1621, 2008
ISSN 1546-9239

Mining Fuzzy Weighted Browsing Patterns from Time Duration and with Linguistic Thresholds

[1]Tzung-Pei Hong, [2]Ming-Jer Chiang and [3]Shyue-Liang Wang
[1]Department of Computer Science and Information Engineering,
National University of Kaohsiung, Kaohsiung, 811, Taiwan, Republic of China
[2]Institute of Information Engineering,
I-Shou University, Kaohsiung, 840, Taiwan, Republic of China
[3]Department of Computer Science, New York Institute of Technology, New York, USA

Abstract: World-wide-web applications have grown very rapidly and have made a significant impact on computer systems. Among them, web browsing for useful information may be most commonly seen. Due to its tremendous amounts of use, efficient and effective web retrieval has become a very important research topic in this field. Techniques of web mining have thus been requested and developed to achieve this purpose. In this research, a new fuzzy weighted web-mining algorithm is proposed, which can process web-server logs to discover useful users' browsing behaviors from the time durations of the paged browsed. Since the time durations are numeric, fuzzy concepts are used here to process them and to form linguistic terms. Besides, different web pages may have different importance. The importance of web pages are evaluated by managers as linguistic terms, which are then transformed and averaged as fuzzy sets of weights. Each linguistic term is then weighted by the importance for its page. Only the linguistic term with the maximum cardinality for a page is chosen in later mining processes, thus reducing the time complexity. The minimum support is set linguistic, which is more natural and understandable for human beings. An example is given to clearly illustrate the proposed approach.

Key words: Browsing pattern, fuzzy set, linguistic term, time duration, web mining, weight

INTRODUCTION

World-wide-web applications have recently grown very rapidly and have made a significant impact on computer systems. Among them, web browsing for useful information may be most commonly seen. Due to its tremendous amounts of use, efficient and effective web retrieval has thus become a very important research topic in this field. Techniques of web mining have thus been requested and developed to achieve this purpose. Cooley *et al*.[7] divided web mining into two classes: web-content mining and web-usage mining[7]. Web-content mining focuses on information discovery from sources across the world-wide-web. On the other hand, web-usage mining emphasizes on the automatic discovery of user access patterns from web servers[8].

In the past, all the web pages were usually assumed to have the same importance in web mining. Different web pages in a web site may, however, have different importance to users in real applications. For example, a web page with merchandise items on it may be more important than that with general introduction. Also, a web page with expensive merchandise items may be more important than that with cheap ones. Besides, the time durations for the pages browsed are however an important feature in analyzing users' browsing behavior. In this research, we thus attempt to mine fuzzy weighted browsing patterns from the browsing time of customers on each web page. The minimum support is given as a linguistic value, which is more natural and understandable for human beings. Since the time durations are numerical and the page importance and the minimum support are linguistic, fuzzy-set concepts are used to process them.

The fuzzy-set theory has been used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning[20,21]. The theory has been applied in fields such as manufacturing, engineering, diagnosis and economics, among others[11,15,17]. Several fuzzy learning algorithms for inducing rules from given sets of data have been designed and used to good effect with specific

Corresponding Author: Tzung-Pei Hong, Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, 811, Taiwan, Republic of China

domains[2,4,9,10,18]. Some fuzzy mining approaches were proposed in[5,13,16,19].

REVIEW OF RELATED MINING APPROACHES

Agrawal and Srikant proposed a mining algorithm to discover sequential patterns from a set of transactions[1]. Five phases are included in their approach. In the first phase, the transactions are sorted first by customer ID as the major key and then by transaction time as the minor key. This phase thus converts the original transactions into customer sequences. In the second phase, the set of all large itemsets are found from the customer sequences by comparing their counts with a predefined support parameter α. This phase is similar to the process of mining association rules. Note that when an itemset occurs more than one time in a customer sequence, it is counted once for this customer sequence. In the third phase, each large itemset is mapped to a contiguous integer and the original customer sequences are transformed into the mapped integer sequences. In the fourth phase, the set of transformed integer sequences are used to find large sequences among them. In the fifth phase, the maximally large sequences are then derived and output to users.

Besides, Hong *et al.*[14] proposed a fuzzy mining algorithm to mine fuzzy rules from quantitative data[14]. They transformed each quantitative item into a fuzzy set and used fuzzy operations to find fuzzy rules. Cai *et al.*[3] proposed weighted mining to reflect different importance to different items. Each item was attached a numerical weight given by users. Weighted supports and weighted confidences were then defined to determine interesting association rules. Yue *et al.*[19] then extended their concepts to fuzzy item vectors.

NOTATION

The notation used in this research is defined as follows.

n:	The total number of log records
c:	The total number of clients
m:	The total number of web pages
d:	The total number of managers
l:	The total number of fuzzy regions
D_i:	The browsing sequence of the i-th client, $1 \le i \le c$
n_i:	The number of log data in D_i, $1 \le i \le c$
D_{id}:	The d-th log transaction in D_i, $1 \le d \le n_i$
I^g:	The g-th web page, $1 \le g \le m$
R^{gk}:	The k-th fuzzy region of I^g, $1 \le k \le l$
v_{id}^g:	The browsing duration of page I^g in D_{id}
f_{id}^g:	The fuzzy set converted from v_{id}^g
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 count of region R^{gk}
max-$count^g$:	The maximum count value among all $count^{gk}$ values for page I^g
max-R^g:	The fuzzy region of page I^g with max-$count^g$
W_{gh}:	The transformed fuzzy weight for the importance of page I^g, evaluated by the h-th manager, $1 \le h \le d$
W_g^{ave}:	The fuzzy average weight for the importance of page I^g
u:	The total number of membership functions for item importance
I_t:	The t-th membership function of item importance, $1 \le t \le u$
I^{ave}:	The fuzzy average weight of all possible linguistic terms of item importance
$wsup_g$:	The fuzzy weighted support of page I^g
α:	The predefined linguistic minimum support value
minsup:	The transformed fuzzy set from the linguistic minimum support value α
wminsup:	The fuzzy weighted set of minimum supports
C_r:	The set of candidate weighted sequences with r linguistic terms
L_r:	The set of large weighted sequences with r linguistic terms.

THE PROPOSED ALGORITHM

Log data in a web site are used to analyze the browsing patterns on that site. Many fields exist in a log schema. Among them, the fields date, time, client-ip and file name are used in the mining process. Only the log data with .asp, .htm, .html, .jva and .cgi are considered web pages and used to analyze the mining behavior. The other files such as .jpg and .gif are thought of as inclusion in the pages and are omitted. The number of files to be analyzed is thus reduced. The log data to be analyzed are sorted first in the order of client-ip and then in the order of date and time. The duration of each web page browsed by a client can then be calculated from the time interval between the page and its next page. Since the time durations are numeric, fuzzy concepts are used here to process them and to

form linguistic terms. Each web page uses only the linguistic term with the maximum cardinality in later mining processes, thus making the number of fuzzy regions to be processed the same as the number of original web pages. The algorithm thus focuses on the most important linguistic terms, which reduce its time complexity.

The importance of web pages is considered and represented as linguistic terms. The proposed fuzzy weighted web-mining algorithm then uses the set of membership functions for importance to transform managers' linguistic evaluations of the importance of web pages into fuzzy weights. The fuzzy weights of web pages from different mangers are then averaged. The algorithm then calculates the weighted supports of the linguistic terms of web pages from browsing sequences. Next, the given linguistic minimum support value is transformed into a fuzzy set of numerical minimum support values. All fuzzy weighted large 1-sequences can thus be found by comparing the fuzzy weighted support of the representative linguistic term of each web page with the fuzzy minimum support. Fuzzy ranking techniques can be used to achieve this purpose. After that, candidate 2-sequences are formed from fuzzy weighted large 1-sequences and the same procedure is used to find all fuzzy weighted large 2-sequences. This procedure is repeated until all fuzzy weighted large sequences have been found. Details of the proposed mining algorithm are described below.

The algorithm

Input: A set of n web log records, a set of m web pages with their importance evaluated by d managers, three sets of membership functions, respectively for browsing duration, web page importance and minimum support and a pre-defined linguistic minimum support value α.

Output: A set of fuzzy weighted browsing patterns.

Step 1: Select the records 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.

Step 2: Transform the client-ips into contiguous integers (called encoded client ID) for convenience, according to their first browsing time. Note that the same client-ips with two closing connections are given two integers.

Step 3: Sort the resulting log data first by encoded client ID and then by date and time.

Step 4: Calculate the time durations of the web pages browsed by each encoded client ID from the time interval between a web page and its next page.

Step 5: Form a browsing sequence D_i for each client c_i by sequentially listing his/her n_i tuples (web page, duration), where n_i is the number of web pages browsed by client c_i. Denote the d-th tuple in D_i as D_{id}.

Step 6: Transform the duration value v_{id}^{g} of the web page I^g in D_{id} into a fuzzy set f_{id}^{g}, represented as $\left(\frac{f_{id}^{g1}}{R^{g1}}+\frac{f_{id}^{g2}}{R^{g2}}+....+\frac{f_{id}^{gl}}{R^{gl}}\right)$, using the given membership functions for the browsing duration of web pages, where I^g is the g-th web page, R^{gk} is the k-th fuzzy region of page 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.

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 count of each fuzzy region R^{gk} in the browsing sequences as:

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

where, c is the number of browsing sequences

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

Step 10: Transform each linguistic term of the importance of the web page I^g, which is evaluated by the h-th manager, into a fuzzy set W_{gh} of weights using the given membership functions of item importance, $1\le g\le m$, $1\le h\le d$.

Step 11: Calculate the fuzzy average weight W_g^{ave} of each web page I^g by fuzzy addition as:

$$W_g^{ave} = \frac{1}{d} * \sum_{h=1}^{d} W_{gh}$$

Step 12: Calculate the fuzzy weighted support $wsup_g$ of the representative region for each web page I^g as:

$$w\sup_g = \frac{\max - R^g \times W_g^{ave}}{c}$$

where c is the number of the clients.

Step 13: Transform the given linguistic minimum support value α into a fuzzy set (denoted minsup) of minimum supports, using the given membership functions for minimum supports.

Step 14: Calculate the fuzzy weighted set (wminsup) of the given minimum support value as:

$$wminsup = minsup \times (\text{the gravity of } I^{ave})$$

Where

$$I^{ave} = \frac{\sum_{t=1}^{u} I_t}{u}$$

with u being the total number of membership functions for item importance and I_t being the t-th membership function. I^{ave} thus represents the fuzzy average weight of all possible linguistic terms of importance.

Step 15: Check whether the weighted support ($wsup_g$) of the representative region for each web page I^g is larger than or equal to the fuzzy weighted minimum support (wminsup) by fuzzy ranking. Any fuzzy ranking approach can be applied here as long as it can generate a crisp rank. If $wsup_g$ is equal to or greater than wminsup, put I^g in the set of large 1-sequences L_1.

Step 16: Set r = 1, where r is used to represent the number of the linguistic items kept in the current large sequences.

Step 17: Generate the candidate set C_{r+1} from L_r in a way similar to that in the aprioriall algorithm[1]. Restated, the algorithm first joins L_r and L_r , under the condition that r-1 linguistic terms in the two sequences are the same and with the same orders. Different permutations represent different candidates. The algorithm then keeps in C_{r+1} the sequences which have all their sub-sequences of length r existing in L_r.

Step 18: Do the following substeps for each newly formed (r+1)-sequences s with linguistic web browsing pattern $(s_1 \rightarrow s_2 \rightarrow .... \rightarrow s_{r+1})$ in C_{r+1}:

- Find the fuzzy weighted count (wf_{is}) of s in each browsing sequence D_i as:

$$wf_{is} = \min_{j=1}^{r+1} (W_{s_j}^{ave} \times f_{is_j})$$

  where f_{is_j} is the membership value of linguistic term s_j in D_i and $W_{s_j}^{ave}$ (derived in step 11) is the average fuzzy weight for s_j. The region s_j must appear after region s_{j-1} in D_i. If two or more same subsequences exist in D_i, then choose the maximum wf_{is} value among those of these subsequences by fuzzy ranking
- Calculate the fuzzy weighted support ($wsup_s$) of sequences s as:

$$wsup_s = \frac{\sum_{i=1}^{c} wf_{is}}{c}$$

  where c is the number of the clients
- Check whether the weighted support ($wsup_s$) of sequences s is greater than or equal to the fuzzy weighted minimum support (wminsup) by fuzzy ranking. If $wsup_s$ is greater than or equal to wminsup, put s in the set of large (r+1)- sequences L_{r+1}

Step 19: IF L_{r+1} is null, then do the next step; otherwise, set r = r + 1 and repeat Steps 17 to 19.

Step 20: For each large r-sequence s (r>1) with fuzzy weighted support $wsup_s$, find the linguistic minimum support region S_i with $wminsup_i \leq wsup_s < wminsup_{i+1}$ by fuzzy ranking, where:

$$wminsup_i = minsup_i \times (\text{the gravity of } I^{ave})$$

$minsup_i$ is the given membership function for S_i. Output sequence s with linguistic support value S_i.

The linguistic weighted browsing patterns output after step 20 can then serve as meta knowledge concerning the given log data.

AN EXAMPLE

In this section, an example is given to show the proposed fuzzy weighted web-mining algorithm. This is

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

Date	Time	Client-ip	Server-ip	Server-port	File-name
2001-03-01	05:39:56	140.127.194.127	140.127.194.88	21	Inside. htm
2001-03-01	05:40:08	140.127.194.127	140.127.194.88	21	home-bg1.jpg
2001-03-01	05:40:10	140.127.194.127	140.127.194.88	21	line1. gif
:	:	:	:	:	:
2001-03-01	05:40:26	140.127.194.127	140.127.194.88	21	person. asp
:	:	:	:	:	:
2001-03-01	05:40:52	140.127.194.82	140.127.194.88	21	cheap. htm
2001-03-01	05:40:53	140.127.194.82	140.127.194.88	21	line1. gif
:	:	:	:	:	:
:	:	:	:	:	:
2001-03-01	05:41:08	140.127.194.128	140.127.194.88	21	cheap. htm
:	:	:	:	:	:
2001-03-01	05:48:38	140.127.194.44	140.127.194.88	21	closing connection
:	:	:	:	:	:
2001-03-01	05:48:53	140.127.194.22	140.127.194.88	21	cheap. htm
:	:	:	:	:	:
2001-03-01	05:50:13	140.127.194.20	140.127.194.88	21	search. asp
:	:	:	:	:	:
2001-03-01	05:53:33	140.127.194.20	140.127.194.88	21	closing connection

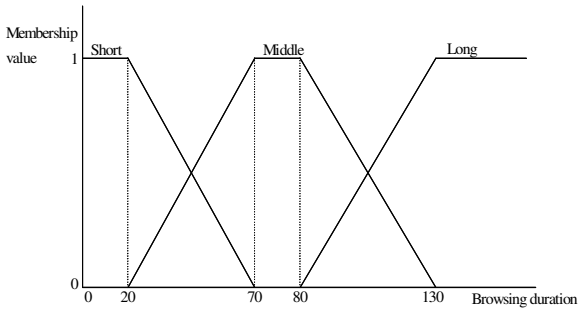

Fig. 1: The membership functions for a browsing duration of a web page

a simple example to show how the proposed algorithm can be used to generate fuzzy weighted browsing patterns for clients' browsing behavior according to the log data in a web server. A part of the log data is shown in Table 1.

Each record in the log data includes fields date, time, client-ip, server-ip, server-port and file-name, among others. Only one file name is contained in each record. For example, the user in client-ip 140.127.194.127 browsed the file inside.htm at 05:39:56 on March 1st, 2001.

Assume the membership functions for a browsing duration of a web page are shown in Fig. 1.

In Fig. 1, the browsing duration is divided into three fuzzy regions: Short, Middle and Long. Thus, three fuzzy membership values are produced for each duration according to the predefined membership functions. For the log data shown in Table 1, the proposed fuzzy web-mining algorithm proceeds as follows.

Table 2: The resulting log data for web mining

Date	Time	Client-ip	File-name
2001-03-01	05:39:56	140.127.194.128	inside.htm
2001-03-01	05:40:26	140.127.194.128	person.asp
2001-03-01	05:40:52	140.127.194.82	cheap.htm
2001-03-01	05:41:08	140.127.194.128	cheap.htm
2001-03-01	05:41:30	140.127.194.22	homepage.htm
2001-03-01	05:41:54	140.127.194.82	inside.htm
2001-03-01	05:42:25	140.127.194.82	cheap.htm
2001-03-01	05:42:46	140.127.194.128	search asp
2001-03-01	05:43:02	140.127.194.22	cheap.htm
2001-03-01	05:43:46	140.127.194.44	inside.htm
2001-03-01	05:44:06	140.127.194.44	search asp
2001-03-01	05:44:07	140.127.194.82	closing connection
2001-03-01	05:44:17	140.127.194.128	closing connection
2001-03-01	05:44:31	140.127.194.22	closing connection
2001-03-01	05:45:47	140.127.194.44	person.asp
2001-03-01	05:46:46	140.127.194.38	cheap.htm
2001-03-01	05:47:45	140.127.194.44	inside.htm
2001-03-01	05:47:53	140.127.194.38	inside.htm
2001-03-01	05:47:56	140.127.194.44	search.asp
2001-03-01	05:48:19	140.127.194.38	search.asp
2001-03-01	05:48:38	140.127.194.44	closing connection
2001-03-01	05:48:53	140.127.194.20	cheap.htm
2001-03-01	05:49:33	140.127.194.38	closing connection
2001-03-01	05:50:13	140.127.194.20	search.asp
2001-03-01	05:51:14	140.127.194.20	person.asp
2001-03-01	05:53:16	140.127.194.20	inside.htm
2001-03-01	05:53:33	140.127.194.20	closing connection

Step 1: The records with file names being .asp, .htm, .html, .jva, .cgi and closing connection are selected for mining. Only the four fields date, time, client-ip and file-name are kept. Assume the resulting log data from Table 1 are shown in Table 2.

Step 2: The values of field client-ip are transformed into contiguous integers according to each client's first browsing time. The transformed results for Table 2 are shown in Table 3. Totally six clients logged on the web

Table 3: Transforming the values of field client-ip into contiguous integers

Date	Time	Client ID	File-name
2001-03-01	05:39:56	1	inside.htm
2001-03-01	05:40:26	1	person.asp
2001-03-01	05:40:52	2	cheap.htm
2001-03-01	05:41:08	1	cheap.htm
2001-03-01	05:41:30	3	homepage.htm
2001-03-01	05:41:54	2	inside.htm
2001-03-01	05:42:25	2	cheap.htm
2001-03-01	05:42:44	1	search asp
2001-03-01	05:43:02	3	cheap.htm
2001-03-01	05:43:46	4	inside.htm
2001-03-01	05:44:06	4	search asp
2001-03-01	05:44:07	2	closing connection
2001-03-01	05:44:17	1	closing connection
2001-03-01	05:44:31	3	closing connection
2001-03-01	05:45:47	4	person.asp
2001-03-01	05:46:46	5	cheap.htm
2001-03-01	05:47:45	4	inside.htm
2001-03-01	05:47:50	5	inside.htm
2001-03-01	05:47:56	4	search.asp
2001-03-01	05:48:19	5	search.asp
2001-03-01	05:48:38	4	closing connection
2001-03-01	05:48:53	6	cheap.htm
2001-03-01	05:49:33	5	closing connection
2001-03-01	05:50:13	6	search.asp
2001-03-01	05:51:14	6	person.asp
2001-03-01	05:53:16	6	inside.htm
2001-03-01	05:53:33	6	closing connection

server and five web pages including homepage.htm, longin.htm, search.asp, cheap.htm and person.asp were browsed in this example.

Step 3: The resulting log data in Table 3 are then sorted first by encoded client ID and then by date and time. Results are shown in Table 4.

Step 4: The time durations of the web pages browsed by each encoded client ID are calculated. Take the first web page browsed by client 1 as an example. Client 1 retrieves the file inside.htm at 05:39:56 on March 1st, 2001 and the next file person.asp at 05:40:26 on March 1st, 2001. The duration of inside.htm for client 1 is then 30 seconds (2001/03/01, 05:39:56-2001/03/01, 05:40:26).

Simple symbols are used here to represent web pages for convenience. Let A, B, C, D and E respectively represent homepage.htm, inside.htm, search.asp, cheap.htm and person.asp. The durations of all pages browsed by each client ID are shown in Table 5.

Step 5: The web pages browsed by each client are listed as a browsing sequence. Each tuple is represented as (web page, duration). The resulting browsing sequences from Table 5 are shown in Table 6.

Table 4: The resulting log data sorted first by client ID and then by data and time

Date	Time	Client ID	File-name
2001-03-01	05:39:56	1	inside.htm
2001-03-01	05:40:26	1	person.asp
2001-03-01	05:41:08	1	cheap.htm
2001-03-01	05:42:46	1	search asp
2001-03-01	05:44:17	1	closing connection
2001-03-01	05:40:52	2	cheap.htm
2001-03-01	05:41:54	2	inside.htm
2001-03-01	05:42:25	2	cheap.htm
2001-03-01	05:44:07	2	closing connection
2001-03-01	05:41:30	3	homepage.htm
2001-03-01	05:43:02	3	cheap.htm
2001-03-01	05:44:31	3	closing connection
2001-03-01	05:43:46	4	inside.htm
2001-03-01	05:44:06	4	search asp
2001-03-01	05:45:47	4	person.asp
2001-03-01	05:47:45	4	inside.htm
2001-03-01	05:47:56	4	search.asp
2001-03-01	05:48:38	4	closing connection
2001-03-01	05:46:46	5	cheap.htm
2001-03-01	05:47:53	5	inside.htm
2001-03-01	05:48:19	5	search.asp
2001-03-01	05:49:33	5	closing connection
2001-03-01	05:48:50	6	cheap.htm
2001-03-01	05:50:13	6	search.asp
2001-03-01	05:51:14	6	person.asp
2001-03-01	05:53:16	6	inside.htm
2001-03-01	05:53:33	6	closing connection

Table 5: The web pages browsed with their durations

Client ID	(Web page, duration)
1	(B, 30)
1	(E, 42)
1	(D, 98)
1	(C, 91)
2	(D, 62)
2	(B, 31)
2	(D, 102)
3	(A, 92)
3	(D, 89)
4	(B, 20)
4	(C, 101)
4	(E, 118)
4	(B, 11)
4	(C, 42)
5	(D, 64)
5	(B, 29)
5	(C, 74)
6	(D, 80)
6	(C, 61)
6	(E, 122)
6	(B, 17)

Table 6: The browsing sequences formed from Table 5

Client ID	Browsing sequence
1	(B, 30) (E, 42) (D, 98) (C, 91)
2	(D, 62) (B, 31) (D, 102)
3	(A, 92) (D, 89)
4	(B, 20) (C, 101) (E, 118) (B, 11) (C, 42)
5	(D, 64) (B, 29) (C, 74)
6	(D, 80) (C, 61) (E, 122) (B, 17)

Table 7: The fuzzy sets transformed from the browsing sequences

Client ID	Fuzzy sets
1	$\left(\frac{0.8}{\text{B.Short}}+\frac{0.2}{\text{B.Middle}}\right),\left(\frac{0.6}{\text{E.Short}}+\frac{0.4}{\text{E.Middle}}\right),$ $\left(\frac{0.6}{\text{D.Middle}}+\frac{0.4}{\text{D.Long}}\right),\left(\frac{0.8}{\text{C.Middle}}+\frac{0.2}{\text{C.Long}}\right)$
2	$\left(\frac{0.2}{\text{D.Short}}+\frac{0.8}{\text{D.Middle}}\right),\left(\frac{0.8}{\text{B.Short}}+\frac{0.2}{\text{B.Middle}}\right),$ $\left(\frac{0.6}{\text{D.Middle}}+\frac{0.4}{\text{D.Long}}\right)$
3	$\left(\frac{0.8}{\text{A.Middle}}+\frac{0.2}{\text{A.Long}}\right),\left(\frac{0.6}{\text{D.Middle}}+\frac{0.4}{\text{D.Long}}\right)$
4	$\left(\frac{1.0}{\text{B.Short}}\right),\left(\frac{0.6}{\text{C.Middle}}+\frac{0.4}{\text{C.Long}}\right),\left(\frac{0.2}{\text{E.Middle}}+\frac{0.8}{\text{E.Long}}\right),$ $\left(\frac{1.0}{\text{B.Short}}\right),\left(\frac{0.6}{\text{C.Short}}+\frac{0.4}{\text{C.Middle}}\right)$
5	$\left(\frac{1.0}{\text{D.Middle}}\right),\left(\frac{0.8}{\text{B.Short}}+\frac{0.2}{\text{B.Middle}}\right),\left(\frac{1.0}{\text{C.Middle}}\right)$
6	$\left(\frac{1.0}{\text{D.Middle}}\right),\left(\frac{0.2}{\text{C.Short}}+\frac{0.8}{\text{C.Middle}}\right),$ $\left(\frac{0.2}{\text{E.Middle}}+\frac{0.8}{\text{E.Long}}\right),\left(\frac{1.0}{\text{B.Short}}\right)$

Step 6: The time durations of the file names in each browsing sequence are represented as fuzzy sets. Take the web page B in the first browsing sequence as an example. The time duration 30 of the web page B is converted into the fuzzy set $(\frac{0.8}{\text{B.Short}}+\frac{0.2}{\text{B.Middle}}+\frac{0.0}{\text{B.Long}})$ by the given membership functions (Fig. 1). This step is repeated for the other web pages and browsing sequences. The results are shown in Table 7.

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

Step 8: The cardinality of each fuzzy region in all the browsing sequences is calculated as the count value. Take the fuzzy region D.Middle as an example. Its cardinality = (0.6+0.8+0.8+0.0+1.0+1.0) = 4.2. This step is repeated for the other regions and the results are shown in Table 8.

Step 9: The fuzzy region with the largest count value among the three possible regions for each file is selected. Take the web page A as an example. Its count is 0.0 for Short, 0.8 for Middle and 0.2 for Long. Since the count for Middle is the largest among the three counts, the region Middle is thus used to represent the web page A in later mining processes. This step is repeated for the other web pages. Thus, Short is chosen for B, Middle is chosen for A, C and D and Long is chosen for E.

Table 8: The counts of the fuzzy regions

Region	Count	Region	Count	Region	Count
A.Short	0.0	C.Short	0.8	E.Short	0.6
A.Midlle	0.8	C.Midlle	3.2	E.Midlle	0.8
A.Long	0.2	C.Long	0.6	E.Long	1.6
B.Short	4.4	D.Short	0.2		
B.Midlle	0.6	D.Midlle	4.2		
B.Long	0.0	D.Long	1.0		

Table 9: The importance of the web pages evaluated by three managers

	Manager		
Web page	Manager 1	Manager 2	Manager 3
A	Important	Ordinary	Ordinary
B	Very important	Important	Important
C	Ordinary	Important	Important
D	Unimportant	Unimportant	Very important
E	Important	Important	Important

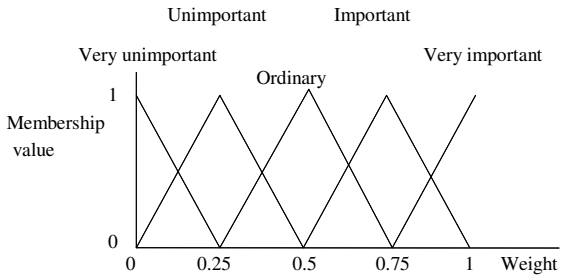

Fig. 2: The membership functions of importance of the web page used in this example

Step 10: Assume the importance of the five web pages (A, B, C, D and E) is evaluated by three managers as shown in Table 9.

Assume the membership functions for importance of the web page are given in Fig. 2.

In Fig. 2, the importance of the web page is divided into five fuzzy regions: Very Unimportant, Unimportant, Ordinary, Important and Very Important. Each fuzzy region is represented by a membership function. The membership functions in Fig. 2 can be represented as follows:
Very Unimportant (VU): (0, 0, 0.25),
Unimportant (U): (0, 0.25, 0.5),
Ordinary (O): (0.25, 0.5, 0.75),
Important (I): (05, 075, 1) and
Very Important (VI): (0.75, 1, 1).

Table 10: The fuzzy weights transformed from the importance of the web pages in Table 9

	Manager		
Web page	Manager 1	Manager 2	Manager 3
A	(0.5, 0.75, 1)	(0.25, 0.5, 0.75)	(0.25, 0.5, 0.75)
B	(0.75, 1, 1)	(0.5, 0.75, 1)	(0.5, 0.75, 1)
C	(0.25, 0.5, 0.75)	(0.5, 0.75, 1)	(0.5, 0.75, 1)
D	(0, 0.25, 0.5)	(0, 0.25, 0.5)	(0, 0, 0.25)
E	(0.5, 0.75, 1)	(0.5, 0.75, 1)	(0.5, 0.75, 1)

Table 11: The average fuzzy weights of all the web pages

Web page	Average fuzzy weight
A	(0.333, 0.583, 0.833)
B	(0.583, 0.833, 1)
C	(0.417, 0.667,0.917)
D	(0, 0.167, 0.417)
E	(0.5, 0.75, 1)

Table 12: The fuzzy weighted supports of the representative regions for the web pages

Item	Fuzzy weighted support
A.Middle	(0.044, 0.078, 0.111)
B.Short	(0.428, 0.611, 0.733)
C.Middle	(0.222, 0.356, 0.489)
D.Middle	(0, 0.117, 0.292)
E.Long	(0.133, 0.2, 0.267)

The linguistic terms for the importance of the web pages given in Table 9 are transformed into fuzzy sets by the membership functions given in Fig. 2. For example, Page A is evaluated to be important by Manager 1. It can then be transformed as a triangular fuzzy set (0.5, 0.75, 1) of weights. The transformed results for Table 9 are shown in Table 10.

Step 11: The average weight of each web page is calculated by fuzzy addition. Take web page A as an example. The three fuzzy weights for web page A are respectively (0.5, 0.75, 1), (0.25, 0.5, 0.75) and (0.25, 0.5, 0.75). The average weight is then ((0.5+0.25+0.25)/3, (0.75+0.5+0.5)/3, (1+0.75+0.75)/3), which is derived as (0.33, 0.58, 0.83). The average fuzzy weights of all the web pages are calculated, with results shown in Table 11.

Step 12: The fuzzy weighted support of each web page is calculated. Take the web page A as an example. The average fuzzy weight of A is (0.333, 0.583, 0.833) from Step 11. Since the region Middle is used to represent the web page A and its count is 2.0, its weighted support is then (0.333, 0.583, 0.833) *0.8/6, which is (0.044, 0.078, 0.111). Results for all the web pages are shown in Table 12.

Step 13: The given linguistic minimum support value is transformed into a fuzzy set of minimum supports.

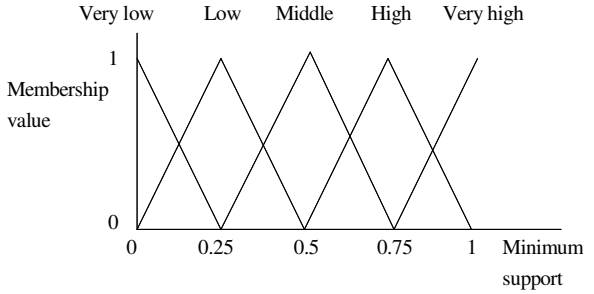

Fig. 3: The membership functions of minimum supports

Table 13: The set of fuzzy weighted large 1-sequences for this example

1-Sequence	Count
B.Short	4.4
C.Middle	3.2
D.Middle	4.2
E.Long	1.6

Assume the membership functions for minimum supports are given in Fig. 3.

Also assume the given linguistic minimum support value is Low. It is then transformed into a fuzzy set of minimum supports, (0, 0.25, 0.5), according to the given membership functions in Fig. 3.

Step 14: The fuzzy average weight of all possible linguistic terms of importance in Fig. 3 is calculated as:

$$I^{ave} = [(0, 0, 0.25)+(0, 0.25, 0.5)+(0.25, 0.5, 0.75) +(0.5, 0.75, 1)+(0.75, 1, 1)]/5 = (0.3, 0.5, 0.7).$$

The gravity of I^{ave} is then (0.3+0.5+0.7)/3, which is 0.5. The fuzzy weighted set of minimum supports for Low is then (0, 0.25, 0.5)×0.5, which is (0, 0.125, 0.25).

Step 15: The fuzzy weighted support of the representative region for each web page is compared with the fuzzy weighted minimum support by fuzzy ranking. Any fuzzy ranking approach can be applied here as long as it can generate a crisp rank. Assume the gravity ranking approach is adopted in this example. Take web page B as an example. The average height of the fuzzy weighted support for B.Short is (0.428+0.611+0.733)/3, which is 0.591. The average height of the fuzzy weighted minimum support is (0+0.125+0.25)/3, which is 0.125. Since 0.591>0.125, B.Short is thus a large fuzzy weighted 1-sequence. Similarly, C.Middle, D.Middle and E.Long are large fuzzy weighted 1-sequences. These 1-sequences are put in L_1 (Table 13).

Step 16: r is set at 1, where r is used to store the number of the linguistic items kept in the current sequences.

Step 17: The candidate set C_2 is first generated from L_1 as follows: (B.Short, B.Short), (B.Short, C.Middle), (B.Short, D.Middle), (B.Short, E.Long), (C.Middle, B.Short), (C.Middle, C.Middle), (C.Middle, D.Middle), (C.Middle, E.Long), (D.Middle, B.Short), (D.Middle, C.Middle), (D.Middle, D.Middle), (D.Middle, E.Long), (E.Long, B.Short), (E.Long, C.Middle), (E.Long, D.Middle), (E.Long, E.Long).

Step 18: The following substeps are done for each newly formed candidate sequences in C_2.

- The fuzzy weighted count of each candidate 2-sequence in each browsing sequence is first calculated. Here, the minimum operator is used for intersection. Take the linguistic browsing sequence (B.Short, C.Middle) for Client 4 as an example. There are three possible subsequences of (B.Short, C.Middle) in that browsing sequence. The average fuzzy weight of web page B is (0.583, 0.833, 1) and the average fuzzy weight of web page C is (0.417, 0.667,0.917) from Step 11. The fuzzy weighted count for the first possible subsequence (B.Short (1.0), C.Middle (0.6)) in the browsing sequence for Client 4 is calculated as: min(1.0*(0.583, 0.833, 1), 0.6*(0.417, 0.667,0.917)) = min ((0.583, 0.833,1), (0.25, 0.4, 0.55)) = (0.25, 0.4, 0.55). Since it has the largest fuzzy value among all the three possible sequences by fuzzy ranking, (0.25, 0.4, 0.55) is then the fuzzy weighted count for (B.Short, C.Middle) in this browsing sequence. The results all the clients for the sequence (B.Short, C.Middle) are shown in Table 14
- The fuzzy weighted count of each candidate 2-sequence in C_2 is then calculated. Results for this example are shown in Table 15. The fuzzy weighted support of each candidate 2-sequences is then calculated. Take (B.Short, C.Middle) as an example. The fuzzy weighted count of (B.Short, C.Middle) is (1, 1.6, 2.083) and the total number of the client is 6. Its fuzzy weighted support is then (1, 1.6, 2.083)/6, which is (0.167, 0.267, 0.347). All the fuzzy weighted supports of the candidate 2-sequences are shown in Table 16
- The fuzzy weighted support of each candidate 2-sequence is compared with the fuzzy weighted minimum support by fuzzy ranking. As mentioned above, assume the gravity ranking approach is adopted in this example. (B.Short, B.Short), (B.Short, C.Middle) and (E.Long, B.Short) are then found to be large 2-sequences. They are then put in L_2

Table 14: The fuzzy weighted count of the sequence (B.Short, C.Middle) in each client

Client	B.Short	C.Middle	(B.Short, C.Middle)
1	(0.467, 0.667, 0.8)	(0.333, 0.533, 0.733)	(0.333, 0.533, 0.733)
2	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)
3	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)
4	(0.583, 0.833,1)	(0.25, 0.4, 0.55)	(0.25, 0.4, 0.55)
5	(0.467, 0.667, 0.8)	(0.417, 0.667, 0.917)	(0.417, 0.667, 0.8)
6	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)

Table 15: The fuzzy weighted counts of the candidate sequences in C_2

Sequences	Count	Sequences	Count
(B.Short, B.Short)	(0.583, 0.833, 1)	(D.Middle, B.Short)	(0, 0.467, 1.167)
(B.Short, C.Middle)	(1, 1.6, 2.083)	(D.Middle, C.Middle)	(0, 0.433, 1.083)
(B.Short, D.Middle)	(0, 0.2, 0.5)	(D.Middle, D.Middle)	(0, 0.1, 0.25)
(B.Short, E.Long)	(0.4, 0.6, 0.8)	(D.Middle, E.Long)	(0, 0.167, 0.417)
(C.Middle, B.Short)	(0.25, 0.4, 0.55)	(E.Long, B.Short)	(0.8, 1.2, 1.6)
(C.Middle, C.Middle)	(0.167, 0.267, 0.367)	(E.Long, C.Middle)	(0.167, 0.267, 0.367)
(C.Middle, D.Middle)	(0, 0, 0)	(E.Long , D.Middle)	(0, 0, 0)
(C.Middle, E.Long)	0.25, 0.4, 0.55)	(E.Long, E.Long)	(0, 0, 0)

Table 16: The fuzzy weighted supports of the sequences in C_2

Sequences	Weight support	Sequences	Weight support
(B.Short, B.Short)	(0.097, 0.139, 0.167)	(D.Middle, B.Short)	(0, 0.078, 0.194)
(B.Short, C.Middle)	(0.167, 0.267, 0.347)	(D.Middle, C.Middle)	(0, 0.072, 0.181)
(B.Short, D.Middle)	(0, 0.033, 0.083)	(D.Middle, D.Middle)	(0, 0.017, 0.042)
(B.Short, E.Long)	(0.067,0.1, 0.133)	(D.Middle, E.Long)	(0, 0.028, 0.069)
(C.Middle, B.Short)	(0.042, 0.067, 0.092)	(E.Long, B.Short)	(0.133, 0.2, 0.267)
(C.Middle, C.Middle)	(0.028, 0.044, 0.061)	(E.Long, C.Middle)	(0.028, 0.044, 0.061)
(C.Middle, D.Middle)	(0, 0, 0)	(E.Long , D.Middle)	(0, 0, 0)
(C.Middle, E.Long)	(0.042, 0.067, 0.092)	(E.Long, E.Long)	(0, 0, 0)

Step 19: Since L_2 is not null, r = r+1 = 2. Steps 17 to 19 are repeated to find L_3. C_3 is then generated from L_2. In this example, C_3 is empty. L_3 is thus empty.

Step 20: The linguistic support values are found for each large r-sequence s (r>1). Take the sequential pattern (B.Short→C.Middle) as an example. Its fuzzy weighted support is (0.167, 0.267, 0.347). Since the membership function for linguistic minimum support region Middle is (0.25, 0.5, 0.75) and for High is (0.5, 0.75, 1), the weighted fuzzy set for these two regions are (0, 0.125, 0.25) and (0.125, 0.25,0.375). Since (0.125, 0.25,0.375)<(0.167, 0.267, 0.347)<(0.25, 0.375, 0.5) by fuzzy ranking, the linguistic support value for sequence (B.Short→C.Middle) is then Middle. The linguistic supports of the other two large 2-sequences can be similarly derived. All the three large linguistic browsing patterns are then output as:

- (B.Short→B.Short) with a low support
- (B.Short→C.Middle) with a middle support
- (E.Long→B.Short) with a low support

These three linguistic browsing patterns are thus output as the meta knowledge concerning the given log data.

CONCLUSION AND FUTURE WORK

In this research, we have proposed a new fuzzy weighted web-mining algorithm, which can process web-server logs to discover useful users' browsing behaviors from the time durations of the paged browsed. In the log data, each transaction contains only one web page. The mining process can thus be simplified when compared to that for multiple-item transactions in Agrawal and Srikant 's mining approach[1]. Since the time durations are numeric, fuzzy concepts are used here to process them and to form linguistic terms. Besides, different web pages may have different importance. The importance of web pages are evaluated by managers as linguistic terms, which are then transformed and averaged as fuzzy sets of weights. Each linguistic term is then weighted by the importance for its page. Only the linguistic term with the maximum cardinality for a page is chosen in later mining processes, thus making the number of fuzzy regions to be processed the same as the number of original web pages. The algorithm therefore focuses on the most important linguistic terms, which reduces its time complexity. The minimum support is also given linguistic. Fuzzy operations including fuzzy ranking are then used to find fuzzy weighted browsing patterns. Compared to previous mining approaches, the proposed one has linguistic inputs and outputs, which are more natural and understandable for human beings.

Although the proposed method works well in fuzzy weighted web mining and can effectively manage linguistic minimum supports, it is just a beginning. There is still much work to be done in this field. Our method assumes that the membership functions are known in advance. In[6,12], we proposed some fuzzy learning methods to automatically derive the membership functions. In the future, we will attempt to dynamically adjust the membership functions in the proposed web-mining algorithm to avoid the bottleneck of membership function acquisition.

REFERENCES

1. Agrawal, R. and R. Srikant, 1995. Mining sequential patterns. The 11th International Conference on Data Engineering, pp: 3-14.
2. Blishun, A.F., 1987. Fuzzy learning models in expert systems. Fuzzy Sets Syst., 22: 57-70.
3. Cai, C.H., W.C. Fu, C.H. Cheng and W.W. Kwong, 1998. Mining association rules with weighted items. The International Database Engineering and Applications Symposium, pp: 68-77.
4. de Campos, L.M. and S. Moral, 1993. Learning rules for a fuzzy inference model. Fuzzy Sets and Systems, 59: 247-257.
5. Chan, K.C.C. and W.H. Au, 1997. Mining fuzzy association rules. The 6th ACM International Conference on Information and Knowledge Management, pp: 10-14.
6. Chen, C.H., T.P. Hong and V.S.M. Tseng, 2006. A cluster-based fuzzy-genetic mining approach for association rules and membership functions. The 2006 IEEE International Conference on Fuzzy Systems, pp: 6971-6976.
7. Cooley, R., B. Mobasher and J. Srivastava, 1997. Grouping web page references into transactions for mining world wide web browsing patterns. Knowledge and Data Engineering Exchange Workshop, pp: 2 -9.
8. Cooley, R., B. Mobasher and J. Srivastava, 1997. Web mining: information and pattern discovery on the world wide web. Ninth IEEE International Conference on Tools with Artificial Intelligence, pp: 558-567.
9. Delgado, M. and A. Gonzalez, 1993. An inductive learning procedure to identify fuzzy systems. Fuzzy Sets and Systems, 55: 121-132.
10. Gonzalez, A., 1995. A learning methodology in uncertain and imprecise environments. Int. J. Intel. Syst., 10: 57-371.

11. Graham, I. and P.L. Jones, 1988. Expert Systems-Knowledge, Uncertainty and Decision, Chapman and Computing, Boston, pp: 117-158.
12. Hong, T.P., C.H. Chen, Y.L. Wu and Y.C. Lee, 2006. A GA-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions. Soft Comput., 10 (11) 1091-1101.
13. Hong, T. P., M.J. Chiang and S.L. Wang, 2002. Mining from quantitative data with linguistic minimum supports and confidences. The 2002 IEEE International Conference on Fuzzy Systems, Honolulu, Hawaii, pp: 494-499.
14. Hong, T.P., C.S. Kuo and S.C. Chi, 1999. Mining association rules from quantitative data. Intelligent Data Analysis, 3 (5): 363-376.
15. Kandel, A., 1992. Fuzzy Expert Systems, CRC Press, Boca Raton, pp: 8-19.
16. Kuok, C.M., A.W.C. Fu and M.H. Wong, 1998. Mining fuzzy association rules in databases. The ACM SIGMOD Record, 27 (1): 41-46.
17. Mamdani, E.H., 1558. Applications of fuzzy algorithms for control of simple dynamic plants. IEEE Proceedings, pp: 1585-1588.
18. Rives, J., 1990. FID3: fuzzy induction decision tree. The First International symposium on Uncertainty, Modeling and Analysis, pp: 457-462.
19. Yue, S., E. Tsang, D. Yeung and D. Shi, 2000. Mining fuzzy association rules with weighted items. The IEEE International Conference on Systems, Man and Cybernetics, pp: 1906-1911.
20. Zadeh, L.A., 1988. Fuzzy logic. IEEE Computer, pp: 83-93.
21. Zadeh, L.A., 1965. Fuzzy sets. Information and Control, 8 (3): 338-353.