Available online at www.sciencedirect.com

ScienceDirect

International Journal of Approximate Reasoning
48 (2008) 47–61

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

Linguistic object-oriented web-usage mining ☆

Tzung-Pei Hong [a,*], Cheng-Ming Huang [b], Shi-Jinn Horng [c]

[a] *Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, ROC*
[b] *Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan, ROC*
[c] *Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan, ROC*

Accepted 14 June 2007
Available online 4 July 2007

Abstract

Web mining has become a very important research topic in the field of data mining due to the vast amount of world wide web services in recent years. The fuzzy and the object concepts have also been very popular and used in a variety of applications, especially for complex data description. This paper thus proposes a new fuzzy object-oriented web mining algorithm to derive fuzzy knowledge from object data log on web servers. Each web page itself is thought of as a class, and each web page browsed by a client is thought of as an instance. Instances with the same class (web page) may have different quantitative attribute values since they may appear in different clients. The proposed fuzzy mining algorithm can be divided into two main phases. The first phase is called the fuzzy intra-page mining phase, in which the linguistic large itemsets associated with the same classes (pages) but with different attributes are derived. Each linguistic large itemset found in this phase is then thought of as a composite item used in phase 2. The second phase is called the fuzzy inter-page mining phase, in which the large sequences are derived and used to represent the relationship among different web pages. Both the intra-page linguistic association rules and inter-page linguistic browsing patterns can thus be easily derived by the proposed algorithm at the same time. An example is given to illustrate the proposed algorithm. Experimental results also show the effects of the parameters used in the algorithm.

Keywords: Data mining; Web mining; Association rule; Browsing pattern; Object-oriented log data

1. Introduction

The rapid development of computer technology, especially increased capacities and decreased costs of storage media, has led businesses to store huge amounts of external and internal information in large databases at low cost. Mining useful information and helpful knowledge from these large databases has thus evolved into an important research area. Due to the vast amounts of data in websites, data mining has recently been used

☆ This is an expanded version of the paper "Linguistic Object-Oriented Web Mining", presented at The 2006 North American Fuzzy Information Processing Society International Conference (NAFIPS), 2006, Montreal, Canada.

[*] Corresponding author.
E-mail address: tphong@nuk.edu.tw (T.-P. Hong).

0888-613X/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2007.06.006

for world wide web applications to help provide better web services to users. Web mining can be divided into three classes: web-structure mining, web-content mining and web-usage mining [12]. Web-structure mining analyzes web structures from hyperlinks in web pages, web-content mining focuses on information discovery from sources across the world wide web, and web-usage mining emphasizes on automatic discovery of user access patterns from web servers.

In the past, several web-mining approaches for finding sequential patterns and user-interested information from the world wide web were proposed [6,7,10,11]. Chen and Sycara proposed the WebMate system to keep track of user interests from the contents of the web pages browsed. It could help users easily search data from WWW [7]. Chen et al. mined path-traversal patterns by first finding the maximal forward references form log data and then obtaining the large reference sequences according to the occurring numbers of the maximal forward references [6]. Cohen et al. sampled only portions of the server logs to extract user access patterns, which were then grouped as volumes [10]. Files in a volume could then be fetched together to increase the efficiency of a web server. Many researches about this topic are still in progress.

Recently, the fuzzy and the object concepts have been very popular and used in different applications, especially for complex data description. Fuzzy set theory was first proposed by Zadeh in 1965 [24]. It is primarily concerned with quantifying and reasoning using natural language in which words can have ambiguous meanings [13,16,22]. This can be thought of as an extension of traditional crisp sets, in which each element must either be in or not in a set. A special notation is often used in the literature to represent fuzzy sets. Assume that x_1 to x_n are the elements in fuzzy set A, and μ_1 to μ_n are, respectively, their grades of membership in A. A is then usually represented as follows:

$$A = \mu_1/x_1 + \mu_2/x_2 + \cdots + \mu_n/x_n.$$

The *scalar cardinality* of a fuzzy set A defined on a finite universal set X is the summation of the membership grades of all the elements of X in A. These concepts will be used in the proposed algorithm to find linguistic association rules. For example, the quantities and the prices of the items sold in a web page may be numerical or linguistic. It is thus very suitable to use fuzzy sets to represent them. As to the object concept, an object represents an instance with several related attribute values and methods integrated together. They have widely applied in the fields such as databases, software engineering, knowledge representation [8,9], geographic information systems, and even computer architecture [17,21].

In the past, web mining is usually performed for inducing association rules and sequential patterns from log data. In this paper, we will try to generalize it and propose an object-oriented fuzzy mining algorithm to derive linguistic knowledge from quantitative object log data on web servers. The browsed pages recorded in a log are used to analyze users' browsing behavior. Since each web page has several quantitative attributes, the object-oriented concepts are used here to process them and to form both intra-page association rules and inter-page browsing patterns. The proposed algorithm is divided into two main phases, one for linguistic intra-page association rules, and the other for linguistic inter-page association rules. The first phase is called the fuzzy intra-page mining phase, in which the association rules within the same classes (pages) are divided. Each large itemset found in this phase can be thought of as a composite item used in phase 2. The second phase is called the fuzzy inter-page mining phase, in which the browsing relations among different web pages are derived. Both the linguistic intra-page association rules and linguistic inter-page browsing patterns can thus be easily derived by the proposed algorithm at the same time. Two apriori-like [4] procedures are adopted to find the two kinds of knowledge. Experiments are also made to show the effect of the proposed algorithm.

The remaining parts of this paper are organized as follows. Related mining algorithms are reviewed in Section 2. The object-oriented concept is introduced in Section 3. The proposed object-oriented web mining algorithm for both linguistic intra-page association rules and linguistic inter-page browsing patterns is described in Section 4. An example to illustrate the proposed algorithm is given in Section 5. Experimental results are described in Section 6. Conclusion and future work are given in Section 7.

2. Review of related mining approaches

Data mining applies nontrivial procedures for identifying effective, coherent, potentially useful, and previously unknown patterns in large databases. Years of effort in data mining has produced a variety of efficient techniques. Depending on the types of databases to be processed, mining approaches may be classified as working on transactional databases, temporal databases, relational databases, and multimedia databases, among others. Depending on the classes of knowledge sought, mining approaches may be classified as finding association rules, classification rules, clustering rules, and sequential patterns, among others. Among them, finding useful association rules and sequential patterns is especially important to real applications.

An association rule is an expression $X \rightarrow Y$, where X is a set of items and Y is usually a single item. It means in the set of transactions, if all the items in X exist in a transaction, then Y is also in the transaction with a high probability. For example, assume whenever customers in a supermarket buy bread and drink, they will also buy fruit. From the transactions kept in the supermarkets, an association rule "*bread* and *drink* → *fruit*" will be mined out.

In the past, Agrawal and his co-workers proposed several mining algorithms based on the concept of large itemsets to find association rules in transaction data [1–4,14,18–20]. They divided the mining process into two phases. In the first phase, candidate itemsets were generated and counted by scanning the transaction data. If the ratio of an itemset appearing in the transactions was larger than a predefined threshold value (called minimum support), the itemset was considered a large itemset. Itemsets containing only one item were first processed. Large itemsets containing only single items were then combined to form candidate itemsets containing two items. This process was repeated until all large itemsets had been found. In the second phase, association rules were induced from the large itemsets found in the first phase. All possible association combinations for each large itemset were formed, and those with calculated confidence values larger than a predefined threshold (called minimum confidence) were output as association rules.

A sequential pattern is an expression $X_1 \rightarrow X_2 \rightarrow \cdots \rightarrow X_n$, where X_i is a set of items. It means in the given set of transactions, if a customer buys all the items in X_1 at some time, then he will buy all the items in X_2 at some other time with a high probability. Similarly, the customer will sequentially buy all the items in X_3 to X_n with a high probability.

Agrawal and Srikant proposed the AprioriAll mining approach to mine sequential patterns from a set of transactions [3]. Five phases were included in this approach. In the first phase, the transactions were sorted first by customer ID as the major key and then by transaction time as the minor key. This phase thus converted the original transactions into customer sequences. In the second phase, the set of all large itemsets were found from the customer sequences by comparing their counts with a predefined support threshold. This phase was similar to the process of mining association rules. Note that when an itemset occurred more than one time in a customer sequence, it was counted only once for this customer sequence. In the third phase, each large itemset was mapped to a contiguous integer and the original customer sequences were transformed into the mapped integer sequences. In the fourth phase, the set of transformed integer sequences were used to find large sequences among them. In the fifth phase, the maximally large sequences were then derived and output to users.

As to fuzzy mining, Hong et al. proposed a fuzzy mining algorithm to mine fuzzy rules from quantitative data [15]. They transformed each quantitative item into a fuzzy set and used fuzzy operations to find fuzzy rules. Cai et al. proposed weighted mining to reflect different importance to different items [5]. 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. then extended their concepts to fuzzy item vectors [23]. Many related researches are still in progress.

In this paper, both linguistic intra-page association rules and linguistic inter-page browsing patterns are to be derived from quantitative object-oriented web pages. A fuzzy object-oriented mining algorithm is proposed, which extends the above concepts to solve the desired problem.

3. Concept of object-oriented data

A primitive object-oriented data is called an object or an instance, each inheriting its characteristics from a super object, called *class*. A class defines the basic structure of objects with common properties, including attributes, default values, and methods. The roles of classes and instances in an object-oriented data set are like those that schema and tuples play in a relational database.

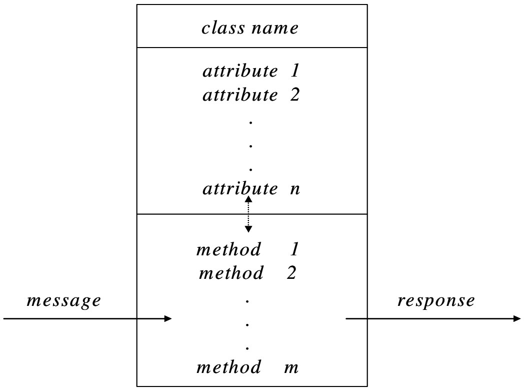

Fig. 1. Structure of a typical class.

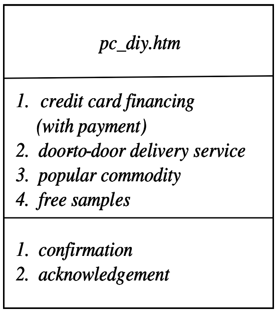

Fig. 2. An example of the class "pc_diy.htm".

A simple structure of a class is shown in Fig. 1, which includes at least three major components: the *class name*, the *attributes* and the *methods*. The *class name* is an identifier used to identify a class, the *attributes* are used to represent the characteristics of a class, and the *methods* are used to implement the operations and functions of a class.

In this paper, an object-oriented log data on a web server is a browsed web page, which is represented as an object or an instance. Note that each web page itself (or page filename) is thought of as a class, which have several attributes, and each web page browsed by a client is thought of as an instance. Instances with the same class (filename) may have different attribute values since they may come from different clients.

An example for a class (page) is given in Fig. 2 to illustrate the above concept. The web page is designed for the sale of DIY goods of PC. The class name is specified as "pc_diy.htm". It includes four attributes, respectively being credit card financing (with payments), door-to-door delivery service, popular commodity, and free samples. It also has two methods, confirmation and acknowledgement. Each client may input different values when buying goods from the web page.

4. The fuzzy object-oriented web mining algorithm

In this section, an algorithm is proposed for discovering both linguistic intra-page association rules and linguistic inter-page browsing patterns from quantitative objected-oriented web log data. The notation used in the algorithm is first explained below:*Notation*

D the set of extracted web log data
D_i the browsing sequence of the ith client
n the number of browsing sequences
w the number of web pages (classes)

m	the number of attributes
I_j	the jth web page (class), $1 \leqslant j \leqslant w$
A_k	the kth attribute, $1 \leqslant k \leqslant m$
$I_j\,.\,A_k$	the kth attribute of the jth web page
$\lvert I_j\,.\,A_k \rvert$	the number of fuzzy regions for $I_j\,.\,A_k$
R_{jkl}	the lth fuzzy region of $I_j\,.\,A_k$, $1 \leqslant l \leqslant \lvert I_j\,.\,A_k \rvert$
$v_{jk}^{(i)}$	the quantitative value of $I_j\,.\,A_k$ *in* D_i
$f_{jk}^{(i)}$	the fuzzy set converted from $v_{jk}^{(i)}$
$f_{jkl}^{(i)}$	the membership value of $v_{jk}^{(i)}$ in region R_{jkl}
$count_{jkl}$	the summation of $f_{jkl}^{(i)}, i = 1$ to n
$support_{jkl}$	the support of $f_{jkl}^{(i)}, i = 1$ to n
α	the predefined minimum support value
λ	the predefined minimum confidence value
C_r	the set of candidate itemsets with r intra-page items
L_r	the set of large itemsets with r intra-page items
C'_z	the set of candidate sequences with z inter-page composite items
L'_z	the set of large sequences with z inter-page composite items.

Quantitative log data in a web site are used to derive linguistic knowledge on that site. Only the log data with .asp, .htm, .html, .jva and .cgi are considered home pages and used to analyze the mining behavior. The other files such as .jpg and .gif are thought of as inclusion in home pages and are omitted. The number of files to be analyzed can thus be reduced.

Many fields exist in a log schema. Among them, the fields *date*, *time*, *client-ip* and *file-name* are used in the mining process. Besides, the attributes to be handled can be obtained from the log file or other files. For example, the attributes may be the quantities of the items to be sold in a web page, the item prices, the browsing time for a page, paying by credit cards or not, among others. Their attribute values may be binary or numeric. Binary values may be thought of as a special case of fuzzy values. In this paper, the attributes in each page (class) are assumed to be numeric, fuzzy concepts are used here to process them and to form linguistic terms. Since each web page has several attributes, the fuzzy object-oriented concepts are used here to process them and to form both fuzzy intra-page association rules and inter-page browsing patterns.

The proposed fuzzy object-oriented web mining algorithm can be divided into two main phases. The first phase is called the fuzzy intra-page mining phase, in which the linguistic large itemsets associated with the same classes (pages) but with different attributes are divided. The phase can find out the linguistic association relation within the same pages. Each large itemset found in this phase can then be thought of as a composite item used in phase 2. The second phase is called the fuzzy inter-page mining phase, in which the large itemsets from the composite items are obtained to get linguistic browsing relations among different web pages. Both the linguistic intra-page association rules and linguistic inter-page browsing patterns can thus be easily derived by the proposed algorithm at the same time. The details of the proposed algorithm are described below.

The fuzzy object-oriented web mining algorithm:

INPUT: A set of w pages (classes) with m attributes, a set of log data, each with some browsed pages and their attribute values, a set of membership functions, a predefined minimum support value α, and a predefined confidence value λ.

OUTPUT: A set of fuzzy intra-page association rules and inter-page browsing patterns.

STEP 1: Select the transactions with file names including .asp, .htm, .html, .jva .cgi and closing connection from the log data. Denote the resulting log data as D.

STEP 2: Form an object-oriented browsing sequence D_j for each client c_j by sequentially listing his/her n_j browsed pages with their attribute values, until a closing connection symbol is met.

STEP 3: Transform the quantitative value $v_{jk}^{(i)}$ of each item attribute $I_j\,.\,A_k$ in the ith client's browsing sequence D_i into a fuzzy set $f_{jk}^{(i)}$ represented as:

$$\left(\frac{f_{jk1}^{(i)}}{R_{jk1}}+\frac{f_{jk2}^{(i)}}{R_{jk2}}+\cdots+\frac{f_{jkp}^{(i)}}{R_{jkp}}\right)$$

using the given membership functions, where $i = 1$ to n, I_j is the jth web page (class), $1 \leqslant j \leqslant w, A_k$ is the kth attribute, $1 \leqslant k \leqslant m, R_{jkl}$ is the lth fuzzy region of attribute $I_j . A_k, f_{jkl}^{(i)}$ is $v_{jk}^{(i)}$'s fuzzy membership value in region R_{jkl}, and p is the number of fuzzy regions for $I_j . A_k$.

STEP 4: Set the membership value $f_{jkl}^{(i)}$ of each fuzzy region R_{jkl} in a browsing sequence D_i as the maximum of all $f_{jkl}^{(i)}$ values if more than one R_{jkl} appear in D_i.

STEP 5: Calculate the scalar cardinality of each fuzzy attribute region R_{jkl} in all the browsing sequences as its count. That is:

$$count_{jkl}=\sum_{i=1}^{n} f_{jkl}^{(i)}.$$

STEP 6: Calculate the support of each attribute region R_{jkl} as:

$$support_{jkl}=count_{jkl}/n,$$

where n is the number of browsing sequences.

STEP 7: Check whether the $support_{jkl}$ of each fuzzy region R_{jkl} is larger than or equal to the predefined minimum support value α. If $support_{jkl}$ satisfies the condition, put R_{jkl} in the set of large 1-itemsets (L_1). That is,

$$L_1=\{R_{jkl}|support_{jkl}\geqslant\alpha, 1\leqslant l\leqslant p, 1\leqslant k\leqslant m, 1\leqslant j\leqslant w\}.$$

STEP 8: If L_1 is null, then exit the algorithm; otherwise, do the next step.

STEP 9: Set $r = 1$, where r is the number of items in the itemsets currently being processed.

STEP 10: Generate the candidate set C_{r+1} from L_r in a way similar to that in the *apriori* algorithm [4] except that $r - 1$ items of two itemsets to be processed in L_r must have the same classes (pages).

STEP 11: Do the following substeps for each newly formed (r+1)-itemset s with items $(s_1, s_2, \ldots, s_{r+1})$ in C_{r+1}:

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

$$f_s^{(i)}=f_{s_1}^{(i)}\cap f_{s_2}^{(i)}\cap\cdots\cap f_{s_{r+1}}^{(i)},$$

where $f_{s_j}^{(i)}$ is the membership value of the fuzzy region s_j in D_i and all the fuzzy regions in s must appear in the same transaction. If the minimum operator is used for the intersection, then:

$$f_s^{(i)}=\operatorname*{Min}_{j=1}^{r+1} f_{s_j}^{(i)}.$$

(b) Set the membership value $f_s^{(i)}$ of s in each browsing sequence D_i as the maximum of all $f_s^{(i)}$ values if more than one s appear in D_i.

(c) Calculate the scalar cardinality of s in all the browsing sequences as its count. That is:

$$count_s=\sum_{i=1}^{n} f_s^{(i)},$$

where n is the number of browsing sequences.

(d) Calculate the support of s as:

$$support_s=count_s/n.$$

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

STEP 12: If L_{r+1} is null, then do the next step; otherwise, set $r = r + 1$ and repeat STEPs 10 and 11.

STEP 13: Each large itemset found so far is then thought of as a composite item and is put in the fuzzy inter-page large 1-sequence (L'_1).

STEP 14: Set $z = 1$, where z is used to represent the number of composite items in the inter-page browsing sequences currently being processed.

STEP 15: Generate the candidate set C'_{z+1} from L'_z in a way similar to that in the *aprioriall* algorithm [4]. Restated, the algorithm generates the candidates under the condition that $z - 1$ items of two sequences in L'_z are the same and with the same orders. Different permutations represent different candidates. The algorithm then keeps in C'_{z+1} the sequences which have all their sub-sequences of length z existing in L'_z.

STEP 16: Do the following substeps for each newly formed fuzzy (z+1)-sequences s with composite items $(s_1, s_2, \ldots, s_{z+1})$ in C'_{z+1}:

(a) Calculate the fuzzy value of s in each client's browsing sequence data D_i as:

$$f_s^{(i)} = f_{s_1}^{(i)} \cap f_{s_2}^{(i)} \cap \cdots \cap f_{s_{z+1}}^{(i)},$$

where $f_{s_j}^{(i)}$ is the membership value of composite item s_j in D_i. If the minimum operator is used for the intersection, then:

$$f_s^{(i)} = \operatorname*{Min}_{j=1}^{z+1} f_{s_j}^{(i)}.$$

(b) Set the membership value $f_s^{(i)}$ of s in each browsing sequence D_i as the maximum of all $f_s^{(i)}$ values if more than one combination of s appear in D_i.

(c) Calculate the scalar cardinality of s in the customer sequences as its count:

$$count_s = \sum_{i=1}^{n} f_s^{(i)},$$

where n is the number of browsing sequences.

(d) Calculate the support of each s as:

$$support_s = count_s / n.$$

(e) If $support_s$ is larger than or equal to the predefined minimum support value α, put s in L'_{z+1} .

STEP 17: If L'_{z+1} is null, do the next step; otherwise, set $z = z + 1$ and repeat STEPs 15 and 16.

STEP 18: Derive the fuzzy intra-page association rules for any large q-itemset s with items $(s_1, s_2, \ldots, s_q), q \geqslant 2$, from the large itemsets L_2 to L_r, using the following substeps:

(a) Form all possible association rules as follows:

$$s_1 \cap \cdots \cap s_{t-1} \cap s_{t+1} \cap \cdots \cap s_q \rightarrow s_t, \quad t = 1 \text{ to } q.$$

(b) Calculate the confidence values of all association rules by:

$$\text{conf}_s = \frac{\sum_{i=1}^{n} f_{s_t}^{(i)}}{\sum_{i=1}^{n} (f_{s_1}^{(i)} \cap \cdots \cap f_{s_{t-1}}^{(i)} \cap f_{s_{t+1}}^{(i)} \cap \cdots \cap f_{s_q}^{(i)})}.$$

(c) Calculate the confidence values of all association rules by: Output the rules with confidence values larger than or equal to the predefined confidence threshold λ.

STEP 19: Derive the maximally large inter-page sequences from the large sequences L'_2 to L'_z as the browsing patterns.

After STEP 19, the two kinds of fuzzy intra-page association rules and inter-page browsing patterns are found from the given set of quantitative object-oriented web transactions.

5. An example

In this section, an example is given to illustrate the proposed fuzzy object-oriented web mining algorithm. This is a simple example to show how the proposed algorithm can be used to generate fuzzy intra-page association rules and inter-page patterns for clients' browsing behavior according to the log data in a web server. Assume a part of the log data are shown in Table 1.

Each transaction 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 transaction. For example, the user with the client-ip 203.66.10.128 browsed the page *component.htm* at 05:39:56 on October 1st, 2005. Also assume each web page has four quantitative attributes, represented as A_1 to A_4, which keep each client's related information to a web page. For example, the attributes may be the quantities of the items to be sold in a web page, the item prices, the browsing time for a page, paying by credit cards or not, among others. Their attribute values may be binary or numeric. Binary values may be thought of as a special case of fuzzy values. In this example, assume four quantitative attributes representing the quantities of four items are used to illustrate the proposed algorithm for simplicity. For the log data shown in Table 1, the fuzzy membership functions for the quantitative attribute values are shown in Fig. 3.

In the example, all the four attributes are assumed to be with the same membership functions for simplicity. Note that attributes with different membership functions can also be managed in a similar way. In Fig. 3, each attribute has three fuzzy regions: Low, Middle and High. Three fuzzy membership values are thus produced for each attribute according to the predefined membership functions. For the transaction data shown in Table 1, the proposed fuzzy object-oriented web mining algorithm proceeds as follows.

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

Date	Time	Client-ip	Server-ip	Server-port	File-name	...
01-10-2005	05:39:56	203.66.10.128	203.66.10.2	80	component.htm	...
01-10-2005	05:40:08	203.66.10.128	203.66.10.2	80	home-bg1.jpg	...
01-10-2005	05:40:10	203.66.10.128	203.66.10.2	80	line1.gif	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮
01-10-2005	05:40:26	203.66.10.128	203.66.10.2	80	waste.asp	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮
01-10-2005	05:40:52	203.66.10.82	203.66.10.2	80	peripheral.htm	...
01-10-2005	05:40:53	203.66.10.82	203.66.10.2	80	line1.gif	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮
01-10-2005	05:41:08	203.66.10.128	203.66.10.2	80	peripheral.htm	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮
01-10-2005	05:48:38	203.66.10.44	203.66.10.2	80	closing connection	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮
01-10-2005	05:48:53	203.66.10.20	203.66.10.2	80	peripheral.htm	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮
01-10-2005	05:50:13	203.66.10.20	203.66.10.2	80	storage.asp	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮
01-10-2005	05:53:33	203.66.10.20	203.66.10.2	80	closing connection	...

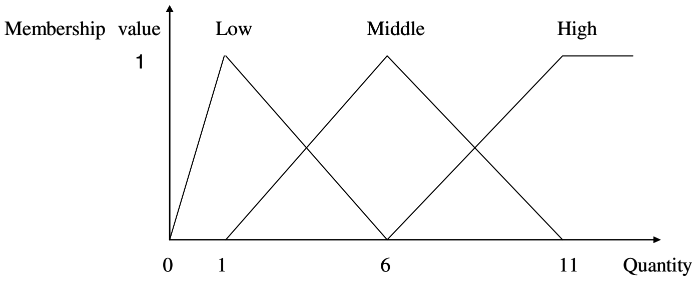

Fig. 3. The membership functions used in the example.

The transactions with file names of .asp, .htm, .html, .jva, .cgi and closing connection are selected for mining. Only the fields *date, time, client-ip* and *file-name* are kept. Besides, the attribute values to be handled can be obtained from the file or other files. Assume in this example, six clients log in the web server and five web pages including *pc_diy.htm*, *component.htm*, *storage.asp*, *peripheral.htm* and *waste.asp* are browsed. The web pages (with their attribute values) browsed by a client are listed as an object-oriented browsing sequence until a closing connection symbol is met. Assume the five web pages *pc_diy.htm*, *component.htm*, *storage.asp*, *peripheral.htm* and *waste.asp* are mapped into I_1 to I_5, and the client-ips are mapped into integers for convenience. Assume the resulting object-oriented browsing sequences after Step 2 are shown in Table 2, where I_j . A_k:v_{jk} represents that the page I_j is with the selected attribute A_k and $A_k's$ quantative value is v_{jk}.

The quantitative values of the item attributes in each customer sequence are transformed into fuzzy sets. Take the item attribute $I_2 \cdot A_1$ in the first customer sequence as an example. The value of the attribute A_1 in Item I_2 is 2, and is converted into a fuzzy set (0.8/Low + 0.2/Middle + 0.0/High) according to the given membership functions. This step is repeated for the other transactions and item attributes, with the results shown in Table 3.

If a fuzzy region appears more than once in a browsing sequence, then its maximum membership value in the browsing sequence is used. Take the fuzzy region $I_4 \cdot A_1 \cdot \text{High}$ for client 2 as an example. Its membership value is max(1.0, 0.8), which is 1.0. The scalar cardinality of each fuzzy attribute region in all the browsing sequences is calculated. Take the fuzzy region $I_2 \cdot A_1 \cdot \text{Low}$ as an example. Its scalar cardinality = $(0.8 + 1.0 + 0.0 + 0.8 + 1.0 + 0.8) = 4.4$.

Table 2
The object-oriented browsing sequences in this example

Client ID	Browsed web pages	Browsed web pages with their attributes
1	I_2, I_5, I_4, I_3	$(I_2 \cdot A_1 : 2, I_2 \cdot A_2 : 11, I_2 \cdot A_3 : 7), (I_5 \cdot A_2 : 5, I_5 \cdot A_3 : 16), (I_4 \cdot A_1 : 13, I_4 \cdot A_3 : 6),$ $(I_3 \cdot A_1 : 10, I_3 \cdot A_3 : 14)$
2	I_4, I_2, I_4, I_3	$(I_4 \cdot A_1 : 16, I_4 \cdot A_3 : 5), (I_2 \cdot A_1 : 1, I_2 \cdot A_2 : 10, I_2 \cdot A_3 : 6), (I_4 \cdot A_1 : 10, I_4 \cdot A_3 : 5),$ $(I_3 \cdot A_1 : 1, I_3 \cdot A_3 : 12)$
3	I_1, I_4	$(I_1 \cdot A_1 : 7, I_1 \cdot A_2 : 10, I_1 \cdot A_3 : 1, I_1 \cdot A_4 : 3), (I_4 \cdot A_1 : 10, I_4 \cdot A_3 : 7)$
4	I_2, I_3, I_4, I_2, I_3	$(I_2 \cdot A_1 : 2, I_2 \cdot A_2 : 15, I_2 \cdot A_3 : 5), (I_3 \cdot A_1 : 13, I_3 \cdot A_3 : 15, I_3 \cdot A_4 : 5), (I_4 \cdot A_1 : 11),$ $(I_2 \cdot A_1 : 2, I_2 \cdot A_2 : 15, I_2 \cdot A_3 : 5), (I_3 \cdot A_1 : 13, I_3 \cdot A_3 : 15, I_3 \cdot A_4 : 5)$
5	I_4, I_2, I_3	$(I_4 \cdot A_3 : 7), (I_2 \cdot A_1 : 1, I_2 \cdot A_2 : 10), (I_3 \cdot A_1 : 1, I_3 \cdot A_3 : 10, I_3 \cdot A_4 : 7)$
6	I_4, I_3, I_5, I_2	$(I_4 \cdot A_1 : 12), (I_3 \cdot A_3 : 15, I_3 \cdot A_4 : 6), (I_5 \cdot A_2 : 8, I_5 \cdot A_3 : 4),$ $(I_2 \cdot A_1 : 2, I_2 \cdot A_2 : 19, I_2 \cdot A_3 : 7)$

Table 3
The fuzzy sets of the web page attributes transformed from Table 2

Client ID	Fuzzy sets
1	$\left(\frac{0.8}{I_2 \cdot A_1 \cdot \text{Low}} + \frac{0.2}{I_2 \cdot A_1 \cdot \text{Middle}}, \frac{1.0}{I_2 \cdot A_2 \cdot \text{High}}, \frac{0.8}{I_2 \cdot A_3 \cdot \text{Middle}} + \frac{0.2}{I_2 \cdot A_3 \cdot \text{High}}\right), \left(\frac{0.2}{I_5 \cdot A_2 \cdot \text{Low}} + \frac{0.8}{I_5 \cdot A_2 \cdot \text{Middle}}, \frac{1.0}{I_5 \cdot A_3 \cdot \text{High}}\right), \left(\frac{1.0}{I_4 \cdot A_1 \cdot \text{High}}, \frac{1.0}{I_4 \cdot A_3 \cdot \text{Middle}}\right),$ $\left(\frac{0.2}{I_3 \cdot A_1 \cdot \text{Middle}} + \frac{0.8}{I_3 \cdot A_1 \cdot \text{High}}, \frac{1.0}{I_3 \cdot A_3 \cdot \text{High}}\right)$
2	$\left(\frac{1.0}{I_4 \cdot A_1 \cdot \text{High}}, \frac{0.2}{I_4 \cdot A_3 \cdot \text{Low}} + \frac{0.8}{I_4 \cdot A_3 \cdot \text{Middle}}\right), \left(\frac{1.0}{I_2 \cdot A_1 \cdot \text{Low}}, \frac{0.2}{I_2 \cdot A_2 \cdot \text{Middle}} + \frac{0.8}{I_2 \cdot A_2 \cdot \text{High}}, \frac{1.0}{I_2 \cdot A_3 \cdot \text{Middle}}\right), \left(\frac{0.2}{I_4 \cdot A_1 \cdot \text{Middle}} + \frac{0.8}{I_4 \cdot A_1 \cdot \text{High}}, \frac{0.2}{I_4 \cdot A_3 \cdot \text{Low}} + \frac{0.8}{I_4 \cdot A_3 \cdot \text{Middle}}\right),$ $\left(\frac{1.0}{I_3 \cdot A_1 \cdot \text{Low}}, \frac{1.0}{I_3 \cdot A_3 \cdot \text{High}}\right)$
3	$\left(\frac{0.8}{I_1 \cdot A_1 \cdot \text{Middle}} + \frac{0.2}{I_1 \cdot A_1 \cdot \text{High}}, \frac{0.2}{I_1 \cdot A_2 \cdot \text{Middle}} + \frac{0.8}{I_1 \cdot A_2 \cdot \text{High}}, \frac{1.0}{I_1 \cdot A_3 \cdot \text{Low}}, \frac{0.6}{I_1 \cdot A_4 \cdot \text{Low}} + \frac{0.4}{I_1 \cdot A_4 \cdot \text{Middle}}\right), \left(\frac{0.2}{I_4 \cdot A_1 \cdot \text{Middle}} \frac{0.8}{I_4 \cdot A_1 \cdot \text{High}}, \frac{0.8}{I_4 \cdot A_3 \cdot \text{Middle}} + \frac{0.2}{I_4 \cdot A_3 \cdot \text{High}}\right)$
4	$\left(\frac{0.8}{I_2 \cdot A_1 \cdot \text{Low}} + \frac{0.2}{I_2 \cdot A_1 \cdot \text{Middle}}, \frac{1.0}{I_2 \cdot A_2 \cdot \text{High}}, \frac{0.2}{I_2 \cdot A_3 \cdot \text{Low}} + \frac{0.8}{I_2 \cdot A_3 \cdot \text{Middle}}\right), \left(\frac{0.8}{I_3 \cdot A_1 \cdot \text{Low}} + \frac{0.2}{I_3 \cdot A_1 \cdot \text{Middle}}, \frac{1.0}{I_3 \cdot A_3 \cdot \text{High}}, \frac{0.2}{I_3 \cdot A_4 \cdot \text{Low}} + \frac{0.8}{I_3 \cdot A_4 \cdot \text{Middle}}\right), \left(\frac{1.0}{I_4 \cdot A_1 \cdot \text{High}}\right),$ $\left(\frac{0.8}{I_2 \cdot A_1 \cdot \text{Low}} + \frac{0.2}{I_2 \cdot A_1 \cdot \text{Middle}}, \frac{1.0}{I_2 \cdot A_2 \cdot \text{High}}, \frac{0.2}{I_2 \cdot A_3 \cdot \text{Low}} + \frac{0.8}{I_2 \cdot A_3 \cdot \text{Middle}}\right), \left(\frac{0.8}{I_3 \cdot A_1 \cdot \text{Low}} + \frac{0.2}{I_3 \cdot A_1 \cdot \text{Middle}}, \frac{1.0}{I_3 \cdot A_3 \cdot \text{High}}, \frac{0.2}{I_3 \cdot A_4 \cdot \text{Low}} + \frac{0.8}{I_3 \cdot A_4 \cdot \text{Middle}}\right)$
5	$\left(\frac{0.8}{I_4 \cdot A_3 \cdot \text{Middle}} + \frac{0.2}{I_4 \cdot A_3 \cdot \text{High}}\right), \left(\frac{1.0}{I_2 \cdot A_1 \cdot \text{Low}}, \frac{0.2}{I_2 \cdot A_2 \cdot \text{Middle}} + \frac{0.8}{I_2 \cdot A_2 \cdot \text{High}}\right), \left(\frac{1.0}{I_3 \cdot A_1 \cdot \text{High}}, \frac{0.2}{I_3 \cdot A_3 \cdot \text{Middle}} + \frac{0.8}{I_3 \cdot A_3 \cdot \text{High}}, \frac{0.8}{I_3 \cdot A_4 \cdot \text{Middle}} + \frac{0.2}{I_3 \cdot A_4 \cdot \text{High}}\right)$
6	$\left(\frac{1.0}{I_4 \cdot A_1 \cdot \text{High}}\right), \left(\frac{1.0}{I_3 \cdot A_3 \cdot \text{High}}, \frac{1.0}{I_3 \cdot A_4 \cdot \text{Middle}}\right), \left(\frac{0.6}{I_5 \cdot A_2 \cdot \text{Middle}} + \frac{0.4}{I_5 \cdot A_2 \cdot \text{High}}, \frac{0.4}{I_5 \cdot A_3 \cdot \text{Low}} + \frac{0.6}{I_5 \cdot A_3 \cdot \text{Middle}}\right), \left(\frac{0.8}{I_2 \cdot A_1 \cdot \text{Low}} + \frac{0.2}{I_2 \cdot A_1 \cdot \text{Middle}}, \frac{1.0}{I_2 \cdot A_2 \cdot \text{High}}, \frac{0.8}{I_2 \cdot A_3 \cdot \text{Middle}} + \frac{0.2}{I_2 \cdot A_3 \cdot \text{High}}\right)$

The support of each fuzzy region is calculated as its count divided by 6. The support of each fuzzy region is then checked against the predefined minimum support value α. Assume in this example, α is set at 40%. The large fuzzy regions are shown in Table 4.

The candidate set C_{r+1} is formed from L_r such that $r-1$ items of two itemsets to be processed in L_r must have the same classes (pages). Note that for $r=1$, any two itemsets to be processed in L_1 must have the same classes. In this case, it can be explained as if the empty items (0 items) in the two 1-itemsets have the same classes. For this example, C_2 is thus generated from L_1 as follows: ($I_2 \cdot A_1 \cdot$ Low, $I_2 \cdot A_2 \cdot$ High), ($I_2 \cdot A_1 \cdot$ Low, $I_2 \cdot A_3 \cdot$ Middle), ($I_2 \cdot A_2 \cdot$ High, $I_2 \cdot A_3 \cdot$ Middle), ($I_3 \cdot A_3 \cdot$ High, $I_3 \cdot A_4 \cdot$ Middle) and ($I_4 \cdot A_1 \cdot$ High, $I_4 \cdot A_3 \cdot$ Middle). The following substeps are executed for each newly formed candidate itemset. The fuzzy membership value of a candidate 2-itemset in each client's browsing sequence is calculated. Here, the minimum operator is used for the intersection. Take ($I_2 \cdot A_1 \cdot$ Low, $I_2 \cdot A_2 \cdot$ High) in the first client's browsing sequence as an example. Its membership value is calculated as: $\min(0.8, 1.0) = 0.8$. The results for all the other client's browsing sequences and candidate 2-itemsets can be derived in a similar fashion. If a candidate 2-itemset appears more than once in a browsing sequence, then its maximum membership value in the browsing sequence is used. Take the sequence ($I_4 \cdot A_1 \cdot$ High, I$_4 \cdot A_3 \cdot$ Middle) as an example. Its membership value in the second browsing sequence is calculated as: $\max[\min(1.0, 0.8), \min(0.8, 0.8)] = 0.8$ since there are two sub-sequences of ($I_4 \cdot A_1 \cdot$ High, I$_4 \cdot A_3 \cdot$ Middle) in that client's browsing sequence. The scalar cardinality (count) of each candidate 2-itemset is then found from all the browsing sequences. Results for this example are shown in Table 5.

The support of each itemset in C_2 is calculated as its count divided by 6. The supports of the candidate 2-itemsets are then checked against the predefined minimum support value 40%. In this example, the five 2-itemsets, ($I_2 \cdot A_1 \cdot$ Low, $I_2 \cdot A_2 \cdot$ High), ($I_2 \cdot A_1 \cdot$ Low, $I_2 \cdot A_3 \cdot$ Middle), ($I_2 \cdot A_2 \cdot$ High, $I_2 \cdot A_3 \cdot$ Middle), ($I_3 \cdot A_3 \cdot$ High, $I_3 \cdot A_4 \cdot$ Middle) and ($I_4 \cdot A_1 \cdot$ High, $I_4 \cdot A_3 \cdot$ Middle) are large and kept in L_2 (Table 6).

Table 4
The set of large 1-itemsets L_1 for this example

Itemset	Support
($I_2 \cdot A_1 \cdot$ Low)	0.73
($I_2 \cdot A_2 \cdot$ High)	0.77
($I_2 \cdot A_3 \cdot$ Middle)	0.57
($I_3 \cdot A_3 \cdot$ High)	0.8
($I_3 \cdot A_4 \cdot$ Middle)	0.43
($I_4 \cdot A_1 \cdot$ High)	0.8
($I_4 \cdot A_3 \cdot$ Middle)	0.57

Table 5
The counts of the 2-itemsets in C_2

Itemset	Count
($I_2 \cdot A_1 \cdot$ Low,$I_2 \cdot A_2 \cdot$ High)	4.0
($I_2 \cdot A_1 \cdot$ Low,$I_2 \cdot A_3 \cdot$ Middle)	3.4
($I_2 \cdot A_2 \cdot$ High,$I_2 \cdot A_3 \cdot$ Middle)	3.2
($I_3 \cdot A_3 \cdot$ High,$I_3 \cdot A_4 \cdot$ Middle)	2.6
($I_4 \cdot A_1 \cdot$ High,$I_4 \cdot A_3 \cdot$ Middle)	2.6

Table 6
The large itemsets and their supports in L_2

Itemset	Support
($I_2 \cdot A_1 \cdot$ Low,$I_2 \cdot A_2 \cdot$ High)	0.67
($I_2 \cdot A_1 \cdot$ Low,$I_2 \cdot A_3 \cdot$ Middle)	0.57
($I_2 \cdot A_2 \cdot$ High,$I_2 \cdot A_3 \cdot$ Middle)	0.53
($I_3 \cdot A_3 \cdot$ High,$I_3 \cdot A_4 \cdot$ Middle)	0.43
($I_4 \cdot A_1 \cdot$ High,$I_4 \cdot A_3 \cdot$ Middle)	0.43

C_3 is then generated from L_2, and only the 3-itemset ($I_2 \cdot A_1 \cdot$ Low, $I_2 \cdot A_2 \cdot$ High, $I_2 \cdot A_3 \cdot$ Middle) is formed. Its support is calculated as 0.53, larger than 0.4. It is thus put in L_3. Since L_3 contains only one itemset, no 4-itemsets are formed and L_4 is null. STEP 13 then begins. Each large sequence found so far is thought of as a composite item and is put in the fuzzy inter-page large 1-sequence (L'_1). Table 7 shows the results.

z is set at 1, where z is used to represent the number of composite items in the inter-page sequences currently being processed. The candidate set C'_{z+1} is generated from L'_z under the condition that z-1 items in the two sequences to be processed in L'_z are the same and with the same orders. Different permutations represent different candidates. The algorithm then keeps in C'_{z+1} the sequences which have all their sub-sequences of length z existing in L'_z . In this example, C'_2 is first generated from L'_1 as follows: $[(I_2 \cdot A_1 \cdot \text{Low}), (I_2 \cdot A_1 \cdot \text{Low})]$, $[(I_2 \cdot A_1 \cdot \text{Low}), (I_2 \cdot A_2 \cdot \text{High})], \ldots, [(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle}), (I_4 \cdot A_1 \cdot \text{High}, I_4 \cdot A_3 \cdot \text{Middle})]$, $[(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle}), (I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle})]$. There are totally 169 candidates generated in C'_2.

The following substeps are executed for each newly formed candidate sequence. The fuzzy membership value of a candidate 2-sequence in each client's browsing sequence is first calculated. Here, the minimum operator is used for intersection. Take $[(I_2 \cdot A_1 \cdot \text{Low}), (I_3 \cdot A_3 \cdot \text{High})]$ in the first client's browsing sequence as an example. Its membership value is calculated as min(0.8, 1.0), which is 0.8. The results for all the other clients' browsing sequences and for all the other candidate 2-sequences can be derived in a similar fashion. If a candidate 2-sequence appears more than once in a browsing sequence, then its maximum membership value in the browsing sequence is used. Take the candidate 2-sequence $[(I_4 \cdot A_1 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$ in the second browsing sequence as an example. There are two sub-sequences of $[(I_4 \cdot A_1 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$ in the browsing sequence. Its membership value is calculated as max[min(1.0, 1.0), min(0.8, 1.0)], which is 1.0. The scalar cardinality (count) of each candidate 2-sequence in all the browsing sequences is calculated. The support of each candidate 2-sequence is calculated as its count divided by 6. These supports are then checked against the predefined minimum support value 0.4. In this example, the twenty-four 2-sequences shown in Table 8 are large and kept in L_2.

C'_3 is then generated from L'_2 . Each candidate 3-sequence is then checked against the minimum support 0.4. The large 3-sequences are shown in Table 9 and kept in L'_3 . No candidate 4-sequences are formed in this example. L'_4 is thus null and STEP 18 then begins.

Linguistic intra-page association rules with confidence values larger than or equal to λ are derived from the large itemsets L_2 to L_r. In this example, $r = 3$. Assume the confidence λ was set at 0.8 in this example. There are nine intra-page association rules output to users. The maximally large inter-page sequences from the large sequences L'_2 to L'_z are then found as the browsing patterns. In this example, $z = 3$. There are fifteen large inter-page browsing sequences found. After STEP 19, the two kinds of linguistic knowledge are found from the given set of quantitative object-oriented web log data. The above rules and sequences obtained can then be explained in a comprehensible way. For example, the association rule "If ($I_2 \cdot A_1 =$ Low and $I_2 \cdot A_2 =$ High), then $I_2 \cdot A_3 =$ Middle " with a confidence factor of 0.8 can be explained as "If item I_2 has a low amount of A_1

Table 7
The set of large inter-page composite 1-sequence L'_1

Sequence	Support
$(I_2 \cdot A_1 \cdot \text{Low})$	0.73
$(I_2 \cdot A_2 \cdot \text{High})$	0.77
$(I_2 \cdot A_3 \cdot \text{Middle})$	0.57
$(I_3 \cdot A_3 \cdot \text{High})$	0.8
$(I_3 \cdot A_4 \cdot \text{Middle})$	0.43
$(I_4 \cdot A_1 \cdot \text{High})$	0.8
$(I_4 \cdot A_3 \cdot \text{Middle})$	0.57
$(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High})$	0.67
$(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_3 \cdot \text{Middle})$	0.57
$(I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle})$	0.53
$(I_3 \cdot A_3 \cdot \text{High}, I_3 \cdot A_4 \cdot \text{Middle})$	0.43
$(I_4 \cdot A_1 \cdot \text{High}, I_4 \cdot A_3 \cdot \text{Middle})$	0.43
$(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle})$	0.53

Table 8
The inter-page large sequences in L'_2

Sequences	Support
$[(I_2 \cdot A_1 \cdot \text{Low}), (I_3 \cdot A_3 \cdot \text{High})]$	0.57
$[(I_2 \cdot A_1 \cdot \text{Low}), (I_4 \cdot A_1 \cdot \text{High})]$	0.43
$[(I_2 \cdot A_2 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$	0.6
$[(I_2 \cdot A_2 \cdot \text{High}), (I_4 \cdot A_1 \cdot \text{High})]$	0.47
$[(I_2 \cdot A_3 \cdot \text{Middle}), (I_3 \cdot A_3 \cdot \text{High})]$	0.43
$[(I_2 \cdot A_3 \cdot \text{Middle}), (I_4 \cdot A_1 \cdot \text{High})]$	0.43
$[(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$	0.53
$[(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High}), (I_4 \cdot A_1 \cdot \text{High})]$	0.4
$[(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_3 \cdot \text{Middle}), (I_3 \cdot A_3 \cdot \text{High})]$	0.43
$[(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_3 \cdot \text{Middle}), (I_4 \cdot A_1 \cdot \text{High})]$	0.43
$[(I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle}), (I_3 \cdot A_3 \cdot \text{High})]$	0.4
$[(I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle}), (I_4 \cdot A_1 \cdot \text{High})]$	0.4
$[(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle}), (I_3 \cdot A_3 \cdot \text{High})]$	0.4
$[(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle}), (I_4 \cdot A_1 \cdot \text{High})]$	0.4
$[(I_4 \cdot A_1 \cdot \text{High}), (I_2 \cdot A_1 \cdot \text{Low})]$	0.43
$[(I_4 \cdot A_1 \cdot \text{High}), (I_2 \cdot A_2 \cdot \text{High})]$	0.47
$[(I_4 \cdot A_1 \cdot \text{High}), (I_2 \cdot A_3 \cdot \text{Middle})]$	0.43
$[(I_4 \cdot A_1 \cdot \text{High}), (I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High})]$	0.4
$[(I_4 \cdot A_1 \cdot \text{High}), (I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_3 \cdot \text{Middle})]$	0.43
$[(I_4 \cdot A_1 \cdot \text{High}), (I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle})]$	0.4
$[(I_4 \cdot A_1 \cdot \text{High}), (I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle})]$	0.4
$[(I_4 \cdot A_1 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$	0.67
$[(I_4 \cdot A_3 \cdot \text{Middle}), (I_3 \cdot A_3 \cdot \text{High})]$	0.43

Table 9
The inter-page large sequences in L'_3

Sequences	Support
$[(I_2 \cdot A_1 \cdot \text{Low}), (I_4 \cdot A_1 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$	0.43
$[(I_2 \cdot A_2 \cdot \text{High}), (I_4 \cdot A_1 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$	0.47
$[(I_2 \cdot A_3 \cdot \text{Middle}), (I_4 \cdot A_1 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$	0.43
$[(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High}), (I_4 \cdot A_1 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$	0.4
$[(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_3 \cdot \text{Middle}), (I_4 \cdot A_1 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$	0.43
$[(I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle}), (I_4 \cdot A_1 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$	0.4
$[(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_2 \cdot \text{High}, I_2 \cdot A_3 \cdot \text{Middle}), (I_4 \cdot A_1 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$	0.4

and a high amount of A_2, then I_2 will also have a middle amount of A_3" with a confidence factor of 0.8. The browsing pattern $[(I_2 \cdot A_1 \cdot \text{Low}, I_2 \cdot A_3 \cdot \text{Middle}), (I_4 \cdot A_1 \cdot \text{High}), (I_3 \cdot A_3 \cdot \text{High})]$ can be explained as when the web page I_2 is browsed with a low amount of A_1 and a middle amount of A_3, then the web page I_4 will be next browsed with a high amount of A_1. Next, the web page I_3 will be browsed with a high amount of A_3.

6. Experimental results

The section reports on experiments made to show the effects of the parameters on the proposed algorithm for linguistic intra-page association rules and linguistic inter-page browsing patterns. They were implemented in JAVA on a Pentium-IV 2.6 GHz personal computer with 1 GB memory. There were 100 object-oriented web pages, and each web page had four quantitative attributes. Data sets with different numbers of customers were run by the proposed algorithm. In each data set, the numbers of browsed web pages in customer sequences were first randomly generated. The web pages and their attribute values were then randomly generated.

Experiments were first performed to find the relationships between numbers of rules or patterns and minimum supports when the minimum customer number was set at 800, the minimum confidence was 0.3 and the average number of object-oriented web pages browsed by a customer was 12. The results for both kinds of intra-page association rules and inter-page browsing patterns are shown in Fig. 4.

It can be observed from Fig. 4 that increasing thresholds decreased the number of frequent sets and associations discovered as is consistent with the properties of data mining. Note that the browsing patterns in Fig. 4 do not include L_1' . The number of linguistic inter-page browsing sequences was much larger than that of linguistic intra-page association rules when the minimum support was smaller than about 0.04. This was because the attribute number existing in a web page was less than the page number in the experiments. This situation usually occurs in real applications. Linguistic intra-page association rules are internal relations within a web page and linguistic inter-page browsing sequences are external relations among web pages.

Experiments were then performed to compare the results of different numbers of customers. The relationship between numbers of linguistic intra-page association rules or linguistic inter-page browsing patterns and minimum support values along with different numbers of customers is shown is Fig. 5.

From Fig. 5, it is easily seen that the numbers of rules or patterns were nearly the same for different numbers of customers since the minimum support and the minimum confidence were set at ratios and independent of customer numbers.

Fig. 6 then shows a comparison of the numbers of different large itemsets for intra-page association rules along with customer numbers. Fig. 7 shows a comparison of the numbers of different inter-page large sequences along with customer numbers. All the lines in these two figures are nearly constant.

At last, the execution time for linguistic intra-page rules and linguistic inter-page patterns with the minimum support value set at 0.05 along with different numbers of customers for an average number of 10 quantitative object-oriented web pages in a client's browsing sequence and a minimum confidence value set at 0.3 is shown in Fig. 8. It is obvious from Fig. 8 that the execution time increased along with the increase of customer numbers. Besides, finding linguistic inter-page browsing patterns spent much more time than finding linguistic

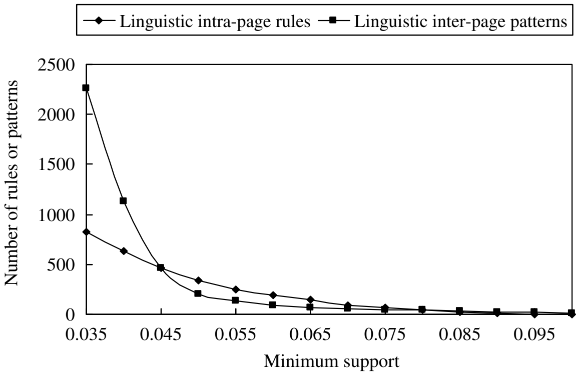

Fig. 4. The relationship between numbers of rules or patterns and minimum support values.

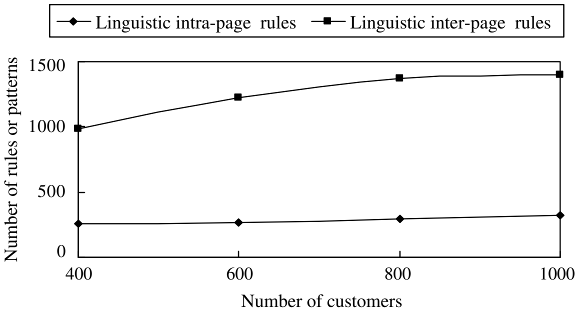

Fig. 5. The relationship between numbers of rules or patterns and numbers of customers.

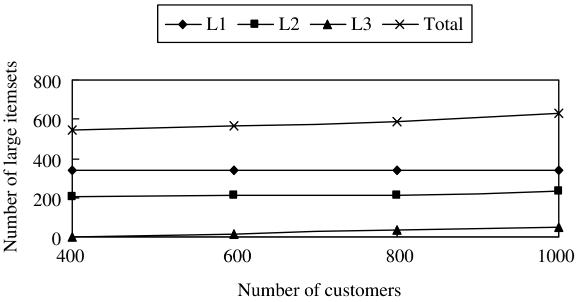

Fig. 6. The numbers of different intra-page large itemsets along with customer numbers.

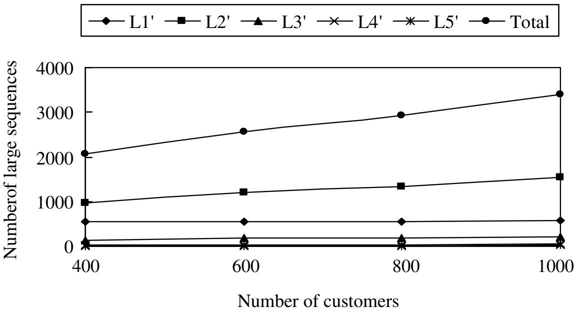

Fig. 7. The numbers of different inter-page large sequences along with customer numbers.

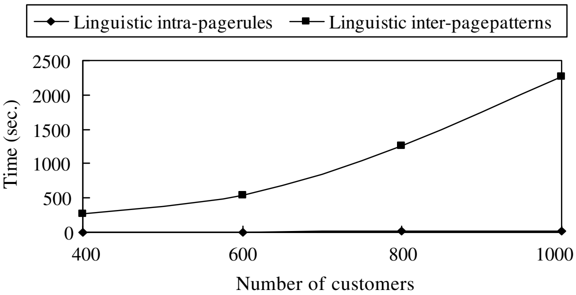

Fig. 8. The relationship between execution times and customer numbers.

intra-page association rules. This was because the number of web pages was usually larger than that of the attributes. The second phase is thus the bottleneck of the proposed algorithm.

7. Conclusion and future work

This paper has proposed a new fuzzy object-oriented web-mining algorithm, which can process quantitative web server logs to discover linguistic intra-page association rules and linguistic inter-page browsing patterns. The proposed fuzzy algorithm is divided into two main phases. The first phase is called the fuzzy intra-page mining phase, in which linguistic large itemsets associated with the same pages but with different quantitative attributes are derived. The second phase is called the fuzzy inter-page mining phase, in which the large itemsets derived from the composite items are used to represent the relationship among different web pages. Both the linguistic intra-page association rules and the linguistic inter-page browsing patterns can thus be easily derived by the proposed algorithm at the same time. An example has also been given to illustrate the algorithm in detail. Experimental results have shown the effects of the parameters on the proposed algorithm. The numbers of linguistic intra-page association rules are often smaller than those of linguistic inter-page browsing patterns because the attribute number is usually less than the web page number in real applications. Finding linguistic inter-page browsing patterns thus spends more time than finding linguistic intra-page association rules. In the future, we will further generalize our approach to manage other different mining problems.

References

[1] R. Agrawal, T. Imielinksi, A. Swami, Mining association rules between sets of items in large database, in: The 1993 ACM SIGMOD International Conference on Management of Data, vol. 22, 1993, pp. 207–216.

[2] R. Agrawal, T. Imielinksi, A. Swami, Database mining: a performance perspective, IEEE Transactions on Knowledge and Data Engineering 5 (6) (1993) 914–925.

[3] R. Agrawal, R. Srikant, Mining sequential patterns, in: The Eleventh International Conference on Data Engineering, 1995, pp. 3–14.

[4] R. Agrawal, R. Srikant, Fast algorithm for mining association rules, in: The International Conference on Very Large Databases, 1994, pp. 487–499.

[5] C.H. Cai, W.C. Fu, C.H. Cheng, W.W. Kwong, Mining association rules with weighted items, The International Database Engineering and Applications Symposium, 1998, pp. 68–77.

[6] M.S. Chen, J.S. Park, P.S. Yu, Efficient data mining for path traversal patterns, IEEE Transactions on Knowledge and Data Engineering 10 (1998) 209–221.

[7] L. Chen, K. Sycara, Webmate: a personal agent for browsing and searching, in: The ACM Second International Conference on Autonomous Agents, 1998, pp. 132–139.

[8] C. Clair, C. Liu, N. Pissinou, Attribute weighting: a method of applying domain knowledge in the decision tree process, in: The Seventh International Conference on Information and Knowledge Management, 1998, pp. 259–266.

[9] P. Clark, T. Niblett, The CN2 induction algorithm, Machine Learning 3 (1989) 261–283.

[10] E. Cohen, B. Krishnamurthy, J. Rexford, Efficient algorithms for predicting requests to web servers, in: The Eighteenth IEEE Annual Joint Conference on Computer and Communications Societies, vol. 1, 1999, pp. 284–293.

[11] R. Cooley, B. Mobasher, J. Srivastava, Grouping web page references into transactions for mining world wide web browsing patterns, in: Knowledge and Data Engineering Exchange Workshop, 1997, pp. 2–9.

[12] R. Cooley, B. Mobasher, J. Srivastava, Web mining: information and pattern discovery on the world wide web, in: The Ninth IEEE International Conference on Tools with Artificial Intelligence, 1997, pp. 558–567.

[13] A. Famili, W.M. Shen, R. Weber, E. Simoudis, Data preprocessing and intelligent data analysis, Intelligent Data Analysis 1 (1) (1997) 1–28.

[14] W.J. Frawley, G. Piatetsky-Shapiro, C.J. Matheus, Knowledge discovery in databases: an overview, in: The AAAI Workshop on Knowledge Discovery in Databases, 1991, pp. 1–27.

[15] T.P. Hong, J.B. Chen, Processing individual fuzzy attributes for fuzzy rule induction, Fuzzy Sets and Systems 112 (1) (2000) 127–140.

[16] A. Kandel, Fuzzy Expert Systems, CRC Press, Boca Raton, 1992, pp. 8–19.

[17] T.D. Kimura, Object-oriented dataflow, in: The Eleventh IEEE International Symposium on Visual Languages, 1995, pp. 180–186.

[18] H. Mannila, Methods and problems in data mining, in: The International Conference on Database Theory, 1997, pp. 41–55.

[19] R. Srikant, R. Agrawal, Mining quantitative association rules in large relational tables, in: The 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Canada, 1996, pp. 1–12.

[20] R. Srikant, Q. Vu, R. Agrawal, Mining association rules with item constraints, The Third International Conference on Knowledge Discovery in Databases and Data Mining, Newport Beach, CA, August 1997, pp. 67–73.

[21] K. Won, Object-oriented databases: definition and research directions, IEEE Transactions on Knowledge and Data Engineering 2 (3) (1990) 327–341.

[22] Y. Yuan, M.J. Shaw, Induction of fuzzy decision trees, Fuzzy Sets and Systems 69 (1995) 125–139.

[23] S. Yue, E. Tsang, D. Yeung, D. Shi, Mining fuzzy association rules with weighted items, The IEEE International Conference on Systems, Man and Cybernetics (2000) 1906–1911.

[24] L.A. Zadeh, Fuzzy sets, Information and Control 8 (3) (1965) 338–353.