Linguistic Object-Oriented Web Mining

Cheng-Ming Huang  
Department of Electrical Engineering  
National Taiwan University of  
Science and Technology  
Taipei, 106, Taiwan, R.O.C.  
apcmhl@ms8.hinet.net

Tzung-Pei Hong  
Department of Computer Science and  
Information Engineering  
National University of Kaohsiung  
Kaohsiung, 811, Taiwan, R.O.C.  
tphong@nuk.edu.tw

Shi-Jinn Horng  
Department of Computer Science and  
Information Engineering  
National Taiwan University of Science  
and Technology  
Taipei, 106, Taiwan, R.O.C.  
horng@mouse.ee.ntust.edu.tw

Abstract - The paper 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. 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. Experimental results also show the effects of the parameters used in the algorithm.

I. INTRODUCTION

Due to the vast amounts of data in websites, data mining has recently been used 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 and Goguen 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. 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 browsing patterns. 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.

II. REVIEW OF RELATED MINING APPROACHES

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 pre-defined 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.

Agrawal and Srikant proposed the AprioriAll mining approach to mine sequential patterns from a set of transactions
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 $\alpha$, and a predefined confidence value $\lambda$.

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 $D_c$ for each client $c$, by sequentially listing his/her $n_i$ browsed pages with their attribute values, until a closing connection symbol is met.

STEP 3: Transform the quantitative value $v^{(i)}_{jk}$ of each item attribute $I_iA_j$ in the $i$-th client’s browsing sequence $D_i$ into a fuzzy set $f^{(i)}_{jk}$ represented as:

III. 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.

IV. 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 object-oriented web log data. 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 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.
STEP 4: Set the membership value $f_{jkli}$ of each fuzzy region $R_{jkli}$ in a browsing sequence $D_r$ as the maximum of all $f_{jkli}$ values if more than one $R_{jkli}$ appear in $D_r$.

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

$$count_{jkli} = \sum f_{jkli}$$

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

$$support_{jkli} = \frac{count_{jkli}}{n}$$

where $n$ is the number of browsing sequences.

STEP 7: Check whether the support of each fuzzy region $R_{jkli}$ is larger than or equal to the predefined minimum support value $\alpha$. If $support_{jkli}$ satisfies the condition, put $R_{jkli}$ in the set of large 1-itemsets $(L_r)$. That is, $L_r = \{R_{jkli} | support_{jkli} \geq \alpha, 1 \leq l \leq p, 1 \leq k \leq m, 1 \leq j \leq w\}$.

STEP 8: If $L_r$ 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 $C_{z+1}$ by joining $L_z$ in a way similar to that in the apriori algorithm [4]. Restated, the algorithm first joins $L'_z$ and $L''_z$ under the condition that $z$-1 items in the two sequences 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 $C_{z+1}$.

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 $C_{z+1}$:

(a) Calculate the fuzzy value of $s$ in each client’s browsing sequence data $D_r$ as:

$$f_s^{(i)} = f_{s_1}^{(i)} \land f_{s_2}^{(i)} \land \ldots \land f_{s_{r+1}}^{(i)}$$

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

$$f_s^{(i)} = \min_{j=1}^{r+1} f_{s_j}^{(i)}$$

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

(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.

STEP 12: If $L_{z+1}$ 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 ($L_{z+1}$).

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 first joins $L'_z$ and $L''_z$ under the condition that $z$-1 items in the two sequences 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 $C_{z+1}$.

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 $D_r$ as:

$$f_s^{(i)} = f_{s_1}^{(i)} \land f_{s_2}^{(i)} \land \ldots \land f_{s_{z+1}}^{(i)}$$

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

$$f_s^{(i)} = \min_{j=1}^{z+1} f_{s_j}^{(i)}$$

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

(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 $s$ as:

$$support_s = \frac{count_s}{n}$$

(e) If $support_s$ is larger than or equal to the predefined minimum support value, 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 \((s_1, s_2, ..., s_q)\), \(q \geq 2\), from the large itemsets \(L_i\) to \(L_s\), using the following substeps:

(a) Form all possible association rules as follows:
\[ s_1 \Lambda \ldots \Lambda s_{i-1} \Lambda s_i \Lambda \ldots \Lambda s_q \rightarrow s_j, \quad t = 1 \text{ to } q. \]

(b) Calculate the confidence values of all association rules by:
\[ \text{conf} = \frac{\sum_{i=1}^{j} f_i^{(1)} \ldots f_i^{(q)} \Lambda f_i^{(q)} \Lambda \ldots \Lambda f_i^{(q)}}{\sum_{i=1}^{j} f_i^{(1)} \ldots f_i^{(q)} \Lambda f_i^{(q)} \Lambda \ldots \Lambda f_i^{(q)}}. \]

(c) Output the rules with confidence values larger than or equal to the predefined confidence threshold \(\lambda\).

STEP 19: Derive the maximally large inter-page sequences from the large sequences \(L_2\) to \(L_s\) 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.

V. 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 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. Also assume the browsing sequences after STEP 2 are shown in Table 1, where \(I_i.A_k.v_{ij}\) represents the page \(I_i\) is with the selected attribute \(A_k\) and \(A_k\)'s quantitative value is \(v_{ij}\).

Assume the fuzzy membership functions for the quantitative attribute values are shown in Fig. 2. 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. The quantitative values of the item attributes in each customer sequence are transformed into fuzzy sets. Take the item attribute \(I_i.A_k\) in the first customer sequence as an example. The value of the attribute \(A_k\) in Item \(I_i\) is 2, and is converted into a fuzzy set \((0.8/\text{Low} + 0.2/\text{Middle} + 0.0/\text{High})\) according to the given membership functions. This step is repeated for the other transactions and item attributes.

---

TABLE 1

<table>
<thead>
<tr>
<th>Client ID</th>
<th>Browsed web pages</th>
<th>Browsed web pages with their attributes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>(I_1, I_3, I_1) &amp; ((I_1.A_3:16, I_1.A_4:5, (I_1.A_1:10, I_4.A_6:6, (I_1.A_1:10, I_1.A_1:12)))</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(I_1, I_2) &amp; ((I_1.A_1:7, I_1.A_1:1, I_1.A_3:3, (I_4.A_1:10, I_4.A_1:7))</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(I_1, I_2) &amp; ((I_1.A_1:12, (I_4.A_1:1, I_3.A_2:10), (I_4.A_1:1, I_3.A_2:10, I_4.A_7:1))</td>
<td></td>
</tr>
</tbody>
</table>

---

Fig. 2 The membership functions used in the example.

The scalar cardinality of each fuzzy attribute region in all the browsing sequences is calculated as \(\text{count}\) value. 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 \(\alpha\). Assume in this example, \(\alpha\) is set at 40%. The large fuzzy regions are shown in Table 2.

TABLE 2

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I_1.A_1, \text{Low})</td>
<td>0.73</td>
</tr>
<tr>
<td>(I_1.A_1, \text{High})</td>
<td>0.77</td>
</tr>
<tr>
<td>(I_1.A_2, \text{Middle})</td>
<td>0.57</td>
</tr>
<tr>
<td>(I_1.A_1, \text{High})</td>
<td>0.8</td>
</tr>
<tr>
<td>(I_1.A_2, \text{Middle})</td>
<td>0.43</td>
</tr>
<tr>
<td>(I_1.A_1, \text{High})</td>
<td>0.8</td>
</tr>
</tbody>
</table>

In this example, the five 2-itemsets are shown in Table 3.

TABLE 3

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I_1.A_1, \text{Low}, I_1.A_1, \text{High})</td>
<td>0.67</td>
</tr>
<tr>
<td>(I_1.A_1, \text{Low}, I_1.A_1, \text{Middle})</td>
<td>0.57</td>
</tr>
<tr>
<td>(I_1.A_1, \text{High}, I_1.A_1, \text{Middle})</td>
<td>0.53</td>
</tr>
<tr>
<td>(I_1.A_1, \text{High}, I_1.A_1, \text{Middle})</td>
<td>0.43</td>
</tr>
<tr>
<td>(I_1.A_1, \text{High}, I_1.A_1, \text{Middle})</td>
<td>0.43</td>
</tr>
</tbody>
</table>

In this example, only the 3-itemset \((I_1.A_1, \text{Low}, I_1.A_1, \text{High}, I_1.A_1, \text{Middle})\) is large. No 4-itemsets are formed and \(L_s\) is null. STEP 13 then begins. Each large sequence found so far is thought of as a fuzzy intra-page composite item and is put in the fuzzy inter-page large 1-sequence \((L_2)\). \(C_2\) is first
generated from $L_1$. There are totally 169 candidates generated in $C_3$. Their supports are then calculated and checked against the predefined minimum support value 0.4. In this example, the twenty-four 2-sequences shown in Table 4 are large and kept in $L_2$. The large 3-sequences are shown in Table 5 and kept in $L_3$. No candidate 4-sequences are formed in this example. $L_4$ is thus null and STEP 18 then begins.

![Table 4: The inter-page large sequences in $L_2$](image)

<table>
<thead>
<tr>
<th>Sequences</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>[I,A,Low], [I,A,High]</td>
<td>0.57</td>
</tr>
<tr>
<td>[I,A,Low], [I,A,High]</td>
<td>0.43</td>
</tr>
<tr>
<td>[I,A,High], [I,A,High]</td>
<td>0.6</td>
</tr>
<tr>
<td>[I,A,High], [I,A,High]</td>
<td>0.47</td>
</tr>
<tr>
<td>[I,A,Middle], [I,A,High]</td>
<td>0.43</td>
</tr>
<tr>
<td>[I,A,Middle], [I,A,High]</td>
<td>0.43</td>
</tr>
<tr>
<td>[I,A,Low], [I,A,High]</td>
<td>0.53</td>
</tr>
<tr>
<td>[I,A,Low], [I,A,High]</td>
<td>0.4</td>
</tr>
<tr>
<td>[I,A,Low], [I,A,Middle]</td>
<td>0.43</td>
</tr>
<tr>
<td>[I,A,Low], [I,A,High]</td>
<td>0.43</td>
</tr>
<tr>
<td>[I,A,High], [I,A,Middle], [I,A,High]</td>
<td>0.4</td>
</tr>
<tr>
<td>[I,A,High], [I,A,Middle], [I,A,High]</td>
<td>0.4</td>
</tr>
<tr>
<td>[I,A,Low], [I,A,High], [I,A,Middle], [I,A,High]</td>
<td>0.4</td>
</tr>
<tr>
<td>[I,A,High], [I,A,Low]</td>
<td>0.43</td>
</tr>
<tr>
<td>[I,A,High], [I,A,High]</td>
<td>0.47</td>
</tr>
<tr>
<td>[I,A,Middle], [I,A,Middle]</td>
<td>0.43</td>
</tr>
<tr>
<td>[I,A,High], [I,A,Low], [I,A,Middle]</td>
<td>0.4</td>
</tr>
<tr>
<td>[I,A,Low], [I,A,High], [I,A,Middle]</td>
<td>0.4</td>
</tr>
<tr>
<td>[I,A,High], [I,A,Low]</td>
<td>0.43</td>
</tr>
<tr>
<td>[I,A,High], [I,A,High]</td>
<td>0.67</td>
</tr>
<tr>
<td>[I,A,Middle], [I,A,High]</td>
<td>0.43</td>
</tr>
</tbody>
</table>

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 2.6GHz 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 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. 3.

![Fig. 3: The relationship between numbers of rules or patterns and minimum support values](image)

Linguistic intra-page association rules with confidence values larger than or equal to $\lambda$ are derived from the large itemsets $L_2$ to $L_3$. In this example, $r = 3$. Assume the confidence $\lambda$ was set at 0.8 in this example. There are nine rules derived and output to users.

The maximally large inter-page sequences from the large sequences $L_2$ to $L_3$ are then found as the browsing patterns. In this example, $z = 3$. There are fifteen large browsing sequences are derived. After STEP 19, the two kinds of linguistic knowledge are found from the given set of quantitative object-oriented web log data.

VI. EXPERIMENTAL RESULTS

It can be observed from Fig. 3 that the number of rules or patterns decreased along with the increase of the minimum support value. It was consistent with the property of data mining. Note that the browsing patterns in Fig. 3 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.

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. 4. It is obvious from Fig. 4 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 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.

![Graph showing linguistic intra-page and inter-page rules]

Fig. 4 The relationship between execution times and customer numbers

VII. 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 linguistic 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