A Web Usage Lattice Based Mining Approach for Intelligent Web Personalization

BAOYAO ZHOU, SIU CHEUNG HUI, ALVIS. C. M. FONG
School of Computer Engineering, Nanyang Technological University, BLK N4, #02a-32, Nanyang Avenue, Singapore 639798
zhouby@pmail.ntu.edu.sg, {asschui, ascmfong}@ntu.edu.sg

Received: January XX 2005; revised: November XX 2005

***Abstract*— With the explosive growth of information available on the World Wide Web, it has become much more difficult to access relevant information from the Web. One possible approach to solve this problem is web personalization. In this paper, we propose a novel WUL (Web Usage Lattice) based mining approach for mining association access pattern rules for personalized web recommendations. The proposed approach aims to mine a reduced set of effective association pattern rules for enhancing the online performance of web recommendations. We have incorporated the proposed approach into a personalized web recommender system known as *AWARS*. The performance of the proposed approach is evaluated based on the efficiency and the quality. In the efficiency evaluation, we measure the number of generated rules and the runtime for online recommendations. In the quality evaluation, we measure the quality of the recommendation service based on precision, satisfactory and applicability. This paper will discuss the proposed WUL-based mining approach, and give the performance of the proposed approach in comparison with the Apriori-based algorithms.**

***Index Terms*— Web Usage Mining, Web Usage Lattice, Association Access Pattern Rules, Web Recommendation**

I. INTRODUCTION

With the explosive growth of information available on the World Wide Web, it has become much more difficult to access relevant information from the Web. One possible approach to solve this problem is web personalization [1]. To support this, we can model the past access behavior of users on the Web. The acquired knowledge or model can then be used for predicting the access behavior of the current user. In personalized web recommendation, it aims to predict which web pages are more likely to be accessed next by the current user. Traditional techniques such as collaborative filtering [2], [3], [4] and hybrid approaches [5], [6], [7] have been used to support web recommendation. However, such techniques suffer from a major drawback in which most users surf websites anonymously via a proxy, and their identities are hidden and difficult to get.

Recently, web usage mining [8] techniques, which aim to discover interesting and frequent user access patterns from web browsing data that are stored in web server logs, proxy server logs or browser logs, have been widely investigated and applied for web recommendation. Association rule mining [9], [10], sequential pattern mining [11] and clustering [12], [13] discover different access patterns from web logs that can be modelled and used for web page recommendation. [14] gave an overview of the above data mining techniques from an application-oriented view. Among the different web access patterns, association access rules are most appropriate for web personalization applications as pointed out by Mobasher *et al* [15].

Apriori [16] is the classical algorithm for mining association rules. Some variants of the Apriori approach such as FP-growth [17] have also been developed for improving the efficiency of the mining process. Algorithms for providing efficient data initialization for mining association rules in data warehouses by concentrating on the measurement of aggregate data were proposed in [18]. Privacy-Preserving Distributed Association-Rule-Mining Algorithm proposed in [19] reduces the overall communication costs significantly. Lin *et al.* [9] and Mobasher *et al.* [10] have applied association rule mining for mining association access patterns for web personalization. In general, a typical web recommender system consists of mainly two processes: one is an offline knowledge discovery process for finding association access pattern rules in order to construct a recommendation model; and the other is an online recommendation process for generating recommendations to a user based on the recommendation model and the current user access activities. To build a successful recommender system, it is important for the online recommendation process to generate recommendations efficiently due to the online, real-time requirement and at the same time, providing high quality recommendation.

The runtime for generating recommendations is proportional to the number of rules available, from which appropriate rules will be searched. The use of the conventional Apriori algorithm and its variants has generated far too many rules, thereby requiring much longer time in searching appropriate rules for recommendation. As a result, the overall performance of a web recommender system will be adversely affected. To tackle this problem, we propose a novel approach for mining a reduced set of high quality association access patterns based on the Web Usage Lattice (WUL) for personalized web recommendation.

In the proposed WUL-based mining approach, we first construct a Web Usage Lattice from the original web logs based on Formal Concept Analysis [20], [21] and then apply our proposed WUL-mine algorithm to mine the most effective and useful set of association access patterns from the Web Usage Lattice. The advantage of the proposed WUL-based approach is that it can generate much fewer number of association access pattern rules without compromising much on quality for web personalization applications when compared with the Apriori-based algorithms [16]. As such, the reduced set of non-redundant WUL-based association access pattern rules can greatly improve the efficiency on generating effective rules for personalized online recommendation.

The rest of this paper is organized as follows. In section 2, we review techniques for intelligent web recommendation. Section 3 describes the system architecture of the proposed web recommender system. The proposed WUL-based mining approach for discovering association access pattern rules is then presented in section 4. The performance of the proposed web recommender system and the WUL-based approach is evaluated in section 5. Finally, the conclusions are given in section 6.

II. Related Work

For the past few years, various statistical and knowledge discovery techniques have been proposed and applied to web recommender systems. These techniques can be classified into collaborative filtering, hybrid approaches and web usage mining.

A. Collaborative Filtering

Collaborative filtering is one of the most successful and widely used recommendation techniques. Collaborative filtering works by building a database of user preferences. A current user is then matched against the database to discover similar ”neighbors”, which refers to other users with historically similar tastes as the current user. Items on which the neighbors like are then recommended to the current user, as he/she will probably also like them. GroupLens [2] is a system that uses purely a collaborative filtering approach to make recommendation of Usenet news. It helps people find desirable articles from a huge stream of news feed. Tapestry [3] and SIFT [4] are recommender systems that also use the collaborative filtering approach. However, the collaborative filtering approach suffers from a few drawbacks. The pure collaborative filtering approach is quite inefficient especially for large websites containing lots of pages. Furthermore, since some users anonymously surf websites via a proxy, their identities are hidden, which can lead to unreliable predictions.

B. Hybrid Approach

Since the pure collaborative filtering approach can be restrictive, some recommender systems such as WebWatcher [5] and Yoda [6] have proposed a hybrid approach that combines content-based approach with the collaborative filtering approach. For example, in WebWatcher [5], it is a web tour guide software agent that accompanies a user from page to page by recommending appropriate hyperlinks based on the content of the web pages the user has visited and a partial understanding of each user's interests. Yoda [6] is a web-based recommender system, which combines collaborative filtering with content-based querying to achieve good accuracy and scalability in real-time. It is designed as an adaptive model to be trained off-line and later deployed for real-time online recommendation. The L-R (Log-based Recommendation) system [7] constructs user models by classifying web access logs and extracting access patterns using transition probability of page accesses. It then recommends relevant pages to users based on both the user models and web content. Sato *et al.* [22] proposed an approach for improving retrieval effectiveness of a search engine by employing the website directory. To make a recommendation, the conceptual similarity between a web page unexamined by the user and the user query or the web pages examined by the user is calculated using the website directory. Approach was proposed in [23] to generate personalized mobile device compatible pages based on existing HTML web pages using dynamic Cascading Style Sheets (CSS). In [24], an approach for mining parallel patterns from mobile users was proposed.

C. Web Usage Mining

Recently, a number of recommender systems have adopted web usage mining techniques, which mine web logs for user models and recommendations. In other words, these recommender systems are still based on collaborative filtering. The web usage mining techniques proposed include association rule mining [9], [10], sequential pattern mining [11], clustering [12], [13], approximate reasoning [25], and Markov models [26]. Compared with the hybrid approaches, the web usage mining techniques can potentially make more accurate recommendation. Other recommender systems using web usage mining include Letizia [27], Syskill & Webert system [28] and Siteseer [29].

In [15], Mobasher *et al.* have pointed out that association rules are most appropriate for web personalization. In this paper, we focus on mining a reduced set of effective association pattern rules for enhancing the online performance of personalized web recommendation.

III. System Architecture

In this research, we have developed a personalized web recommender system known as *AWARS* (*A*ssociation *W*eb *A*ccess-based *R*ecommender *S*ystem) that uses the proposed WUL-based mining approach for personalized web recommendation. The system aims to help users to browse and access related web pages more efficiently and effectively.

Figure 1 shows the architecture of the proposed *AWARS* system which consists of two major processes: off-line mining and on-line recommendation.

In the off-line mining process, all the users' web access activities of a website are recorded by the Web server and stored into the Web Server Logs. Each user access record contains the client IP address, request time, requested URL, HTTP status code, etc. Users are treated as anonymous since

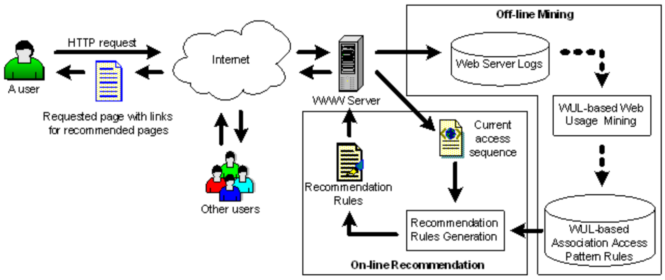
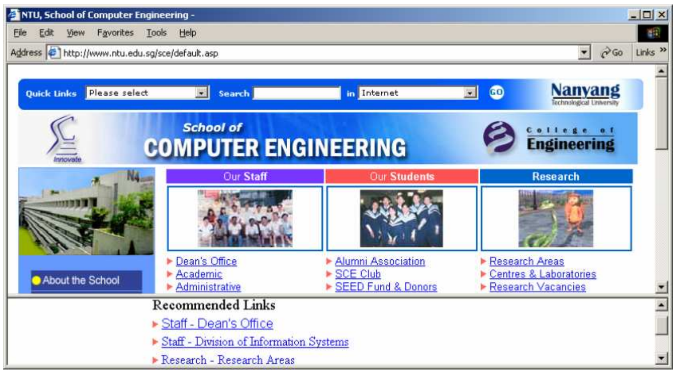

A user
HTTP request
Requested page with links for recommended pages
Internet
Other users
WWW Server
Off-line Mining
Web Server Logs
WUL-based Web Usage Mining
Current access sequence
Recommendation Rules
Recommendation Rules Generation
WUL-based Association Access Pattern Rules
On-line Recommendation

Fig. 1. The architecture of the *AWARS* system.


NTU, School of Computer Engineering -
File Edit View Favorites Tools Help
Address
Go
Links
Quick Links
Please select
Search
in
Internet
GO
Nanyang Technological University
School of COMPUTER ENGINEERING
Innovate
College of Engineering
Our Staff
Our Students
Research
About the School

http://www.ntu.edu.sg/sce/default.asp

- Dean's Office
- Academic
- Administrative
- Alumni Association
- SCE Club
- SEED Fund & Donors
- Research Areas
- Centres & Laboratories
- Research Vacancies

Recommended Links

- Staff - Dean's Office
- Staff - Division of Information Systems
- Research - Research Areas

Fig. 2. An example browser display of the recommended links from the *AWARS* system.

the IP addresses are not mapped to any user-identifiable profile database.The program of *WUL-based Web Usage Mining approach* is then applied to the Web Server Logs to construct a Web Usage Lattice. The WUL-based association access pattern rules are then mined from the Web Usage Lattice and stored in the rule database. The Web Usage Lattice can be updated or regenerated regularly to incorporate new access data. We will discuss the WUL-based mining approach in section 4.

In the on-line recommendation process, a user's HTTP requests in the current browsing session are recorded according to the order the user visited the website. The current access sequence is then constructed by an agent software installed in the Web server from the recorded access requests. By matching the patterns of the user's current access sequence with the WUL-based association access pattern rules, the *Recommendation Rules Generation*program will find and generate the most appropriate recommendation rules. The corresponding recommendation links will then be inserted into the current requested page dynamically for display. An example browser display is shown in Figure 2. The upper frame displays the original requested web page and the lower frame displays a list of recommended links.

IV. WUL-based Mining Approach

Web usage data [8] refers to data from Web server access logs, proxy server logs, browser logs, user profiles, registration data, cookies, user queries, bookmark data, mouse clicks and scrolls, or any other data as a result of user interactions. In this paper, we focus only on mining the web server logs for personalized web recommendation. Nevertheless, the proposed approach can also be applied to other web usage data in a similar manner.

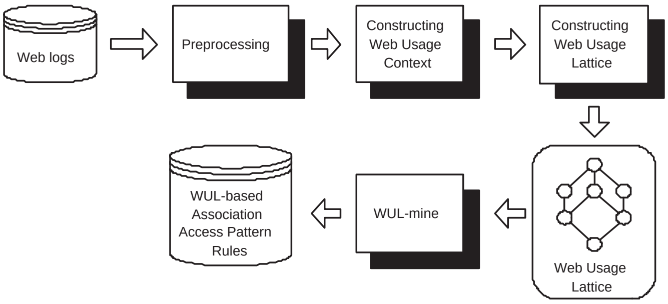

Web logs
Preprocessing
Constructing Web Usage Context
Constructing Web Usage Lattice
Web Usage Lattice
WUL-mine
WUL-based Association Access Pattern Rules

Fig. 3. The WUL-based web usage mining approach.

Figure 3 gives an overview of the proposed WUL-based mining approach for association access pattern rules. The proposed approach comprises the following steps: (1) Preprocessing; (2) Constructing Web Usage Context; (3) Constructing Web Usage Lattice; and (4) WUL-mine algorithm for mining association access pattern rules from Web Usage Lattice.

A. Preprocessing

The preprocessing step aims to preprocess the original web logs to identify all web access sessions. For web server logs, all users' access activities of a website are recorded by the Web server of the website. Each user access record contains the client IP address, request time, requested URL, HTTP status code, etc. Users are treated as anonymous since the IP addresses are not mapped to any user-identifiable profile database.

Generally, web logs can be regarded as a collection of sequences of access events from one user or session in timestamp ascending order. Preprocessing tasks [30] including data cleaning, user identification, and session identification can be applied to the original web log files to obtain all web access sessions.

Data cleaning: In the original web logs, not all records are valid for web usage mining. We only treat requested documents as access events when they are in HTML format. Therefore, apart from records such as URLs of HTML or extended HTML documents (e.g., ASP, PHP or JSP), all other records are discarded from the web logs. These include records containing URLs of GIF, JPG or BMP files. HTTP status codes are used to indicate the success or failure of the requested event. Only records with codes between 200 and 299 are considered as successful records, and others are discarded from the web logs.

User identification: For analyzing user access behavior, unique users must be identified. As mentioned earlier, users are treated as anonymous in most Web servers. We can simplify the user identification process to client IP identification. In other words, requests from the same IP address can be treated as from the same user and put into the same group under that user. In order to identify users more accurately, some other information from the web logs may be helpful. The agent filed in web logs records information on the client browser

TABLE I

A DATABASE OF WEB ACCESS SESSIONS

Session ID	Web Access Session
S1	P2, P1, P5, P2, P6
S2	P2, P6, P3
S3	P3, P4, P3, P6
S4	P1, P6, P2, P4, P1, P3
S5	P3, P4, P3

TABLE II

AN EXAMPLE WEB USAGE CONTEXT

	P1	P2	P3	P4	P5	P6
S1	X	X			X	X
S2		X	X			X
S3			X	X		X
S4	X	X	X	X		X
S5			X	X		

and operating system. Since different users may access the website through the same proxy server, the IP address may be the same. However, the agent type may not be the same in many cases, it is quite reasonable to assume that each different agent type for the same IP address represents a different user.

Session identification: For logs from one user that span a long period of time, it is very likely that the user has visited the website more than once. The goal of session identification is to divide web logs of each user into individual access sessions. The simplest method is to set a timeout threshold. If the difference between the requested time of two adjacent records from a user is greater than the timeout threshold, it could be considered that a new access session has started. Here, we use 30 minutes as the default timeout threshold.

Let M be a set of unique access events, which represents web resources, i.e. web pages, URLs, or topics, accessed by users. A web access session $S = \langle (e_1, t_1), (e_2, t_2), ..., (e_n, t_n) \rangle$ is a sequence of access events $e_i \in M$ for $1 \leq i \leq n$ with their requested time t_i. Note that it is not necessary that $e_i \neq e_j$ for $i \neq j$ in S, that is repeat of items is allowed. However, it is not necessary mean that all web pages accessed by the user are of interest, as some intermediate pages might need to be accessed first before reaching the targeted web pages. As such, we set a duration threshold d_{min} as a constraint to filter out non-targeted access events. The duration d_i of the access event e_i can simply be estimated as $d_i = (t_{i+1} - t_i)$. For the last access event e_n in each web access session that does not have "t_{n+1}" for estimating the duration, we have used the average duration of the relevant session as the estimated duration for the last access event, i.e. $d_n = (d_1 + d_2 + ... + d_{n-1})/(n-1)$. All access events whose durations are less than the predefined duration threshold d_{min} are regarded as not useful and are discarded. Table 1 shows an example database containing five web access sessions after the preprocessing step.

B. Constructing Web Usage Context

After the preprocessing step, we will obtain all the web access sessions from the original web logs. Then, we construct a Web Usage Context based on the web access sessions. Web Usage Context is defined as follows.

Definition 4.1 A *Web Usage Context* is a triple $K = (G, M, I)$, where G is a set of all web access sessions in web logs for a website, M is a set of all web resources in the website, and $I \subseteq G \times M$ is a binary relation between G and M, which indicates the web resources of the website that are accessed by the sessions. A web access session g in a relation I with a web resource m is denoted as $(g, m) \in I$.

If $(g, m) \in I$, it means that the user was interested in the web resource m in the web access session g.

A Web Usage Context can be represented by a cross table with rows labeled by web access sessions and columns labeled by web resources. A cross in row g and column m indicates a relation between the web access session g and the web resource m. Table 2 shows the Web Usage Context constructed from the web access sessions given in Table 1, which consists of five web access sessions, namely S1, S2, ..., S5 and six web resources, namely P1, P2, ..., P6. The relation between a web access session and a web resource is represented by a symbol "X", which means that the user is interested in the specified web resource from the corresponding web access session.

C. Constructing Web Usage Lattice

This step constructs a Web Usage Lattice based on a Web Usage Context.

Definition 4.2 Given a Web Usage Context $K = (G, M, I)$, we define the set of web resources common to the web access sessions in A as $A' = \{m \in M | \forall g \in A : (g, m) \in I\}$ for a set $A \subseteq G$ and the set of web access sessions which have all web resources in B as $B' = \{g \in G | \forall m \in B : (g, m) \in I\}$ for a set $B \subseteq M$.

Definition 4.3 A *web access activity* of a Web Usage Context $K = (G, M, I)$ is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. The sets A and B are called *web access session set* and *web resource set* of the web access activity (A, B) respectively.

Definition 4.4 Let (A_1, B_1) and (A_2, B_2) be two web access activities of a Web Usage Context $K = (G, M, I)$, (A_1, B_1) is called *sub-activity* of (A_2, B_2) and denoted as $(A_1, B_1) \leq (A_2, B_2)$, if and only if $A_1 \subseteq A_2 (\Leftrightarrow B_2 \subseteq B1)$. Equivalently, (A_2, B_2) is called *super-activity* of (A_1, B_1). The relation $\leq$ is called *hierarchical order* (or simply *order*) of web access activities.

Definition 4.5 A *Web Usage Lattice* of a Web Usage Context $K = (G, M, I)$ is a set of all web access activities of K with hierarchical order $\leq$, and is denoted as $\Re(G, M, I)$.

Figure 4 shows the Web Usage Lattice constructed from the Web Usage Context given in Table 2. Each node in the figure represents a web access activity in the lattice with the corresponding web resource set on the left and web access session set on the right. Each edge in the lattice represents a hierarchical relationship. For example, the node located at

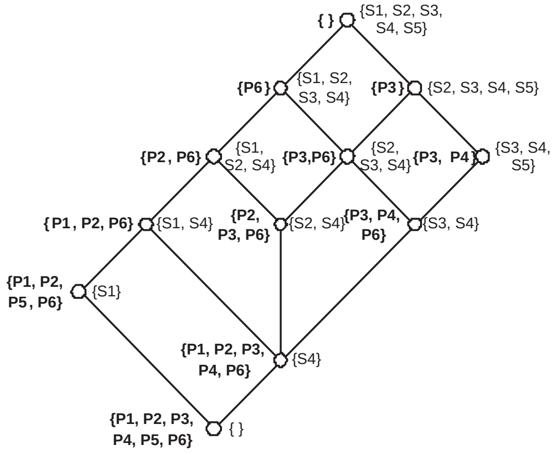


Fig. 4. The Web Usage Lattice constructed from the Web Usage Context given in Table 2.

the far right of the lattice represents the web access activity ({S3, S4, S5}, {P3, P4}), which is a sub-activity of ({S2, S3, S4, S5}, {P3}), and a super-activity of ({S3, S4}, {P3, P4, P6}). There are a total of 12 web access activities in the lattice including one upper activity ({S1, S2, S3, S4, S5}, {$\emptyset$}) that is the super-activity of all other activities, and one lower activity ({$\emptyset$}, {P1, P2, P3, P4, P5, P6}) that is the sub-activity of all other activities. The Web Usage Lattice can be treated as a conceptual model of web logs which can then be mined for discovering interesting and frequent user access patterns from web usage data.

D. WUL-mine Algorithm

Association rule mining searches for interesting relationships among items in a given data set. Given a set of items $I = \{I_1, I_2, ..., I_m\}$ and a database of transactions $D = \{t_1, t_2, ..., t_n\}$ where $t_i = \{I_{i1}, I_{i2}, ..., I_{ik}\}$ and $I_{ij} \in I$, an *association rule* [15] is an implication in the form $X \Rightarrow Y$ where $X, Y \subset I$ are sets of items called *itemsets* and $X \cap Y = \emptyset$. X is called *antecedent* and Y is called *consequent*. We are generally not interested in all implications but only those that are important. Here, two features called *support* and *confidence* are commonly used to measure the importance of association rules. The *support* for an association rule $X \Rightarrow Y$ is the percentage of transactions in the database that contain $X \cup Y$. The *confidence* for an association rule $X \Rightarrow Y$ is the ratio of the number of transactions that contain $X \cup Y$ to the number of transactions that contain X. Rules that satisfy a minimum support threshold ($MinSup$) and a minimum confidence threshold ($MinConf$) are called *strong* rules.

Association access pattern rules from web usage data can be discovered based on Web Usage Lattice. They are defined as follows.

Definition 4.6 Let M be a set of web resources of a Web Usage Context $K = (G, M, I)$. An association access pattern rule is a pair $X \Rightarrow Y$ with $X, Y \subseteq M$. The *support* of the rule is defined as

$$sup(X \Rightarrow Y) = \frac{|(X \cup Y)'|}{|G|}$$

and the *confidence* of the rule is defined as

$$conf(X \Rightarrow Y) = \frac{|(X \cup Y)'|}{|X'|}.$$

Definition 4.7 Let $B \subseteq M$ and $MinSup \in [0, 1]$. The *support* of the web resource set B in a Web Usage Context $K = (G, M, I)$ is defined as $sup(B) = \frac{|B'|}{|G|}$. B is said to be a *frequent* web resource set if $sup(B) \geq MinSup$. A web access activity is called *frequent activity* if its web resource set is *frequent*.

Most association rule mining algorithms employ a support-confidence framework. However, such approaches suffer from the problem in which a large number of rules are usually returned including redundant rules. In other words, despite using the minimum support and confidence thresholds to exclude uninteresting rules, many rules that are not interesting may still be generated. Mining association access pattern rules using the Web Usage Lattice can significantly reduce the number of rules without compromising much on quality. This approach extracts only a small subset of association access pattern rules, which is called *basis*, from which all other rules can be derived. The association access pattern rules mined from the Web Usage Lattice is referred to as WUL-based association access pattern rules, which are defined as follows.

Definition 4.8 Given a Web Usage Context $K = (G, M, I)$, *WUL-based association access pattern rules* consist of two kinds of rules: (1) WUL-based exact rules (i.e. rules with 100% confidence): $B_1 \Rightarrow B_2$, where B_1 and B_2 are two web resource sets of frequent and nonempty web access activities, and the activity (B_1', B_1) has activity (B_2', B_2) as its only immediate super-activity; (2) WUL-based approximate rules (i.e. rules with less than 100% confidence): $B_1 \Rightarrow B_2$, where B_1 and B_2 are two web resource sets of frequent and nonempty web access activities, and the activity (B_1', B_1) is an immediate sub-activity of (B_2', B_2). In addition, each rule, i.e. $B_1 \Rightarrow B_2 (B_1, B_2 \neq \emptyset)$, should satisfy with the conditions $sup(B_1 \Rightarrow B_2) \geq MinSup$ and $conf(B_1 \Rightarrow B_2) \geq MinConf$, where $MinSup$ and $MinConf$ are the given minimum support and confidence respectively.

From the above definition, it is obvious that each WUL-based exact rule corresponds exactly to one edge that connects the sub-activity with its only super-activity in Web Usage Lattice, and each WUL-based approximate rule corresponds exactly to one edge that connects the super-activity with one of its sub-activities in Web Usage Lattice. For example, in the Web Usage Lattice shown in Figure 3, the edge from the activity node {P2, P6} to {P6} represents an exact rule $P2 \Rightarrow P6$ with support = 60% and confidence = 100%, and the edge from the activity node {P6} to {P2, P6} represents an approximate rule $P6 \Rightarrow P2$ with support = 60% and confidence = 75%. The WUL-mine algorithm for mining association access pattern rules and the computation for the support and confidence are given as follows.

```
Algorithm:
WUL-mine for Mining Association Access
Pattern Rules

Input:
  1) $WUL$ - Web Usage Lattice based on a
     Web Usage Context $K = (G, M, I)$.
  2) $NL = \{N_1, N_2, ..., N_m\}$ - a set of activity
     nodes in $WUL$, where $N_i = \langle A_i, B_i, P_i \rangle$,
     $A_i \subseteq G$ is the web access session set
     of $N_i$ , $B_i \subseteq M$ is the web resource
     set of $N_i$, $P_i = \{N_{i1}, N_{i2}, ..., N_{ip}\} \subseteq NL$ is
     the immediate parent nodes of $N_i$.
  3) $MinSup$ - minimum support threshold.
  4) $MinConf$ - minimum confidence
     threshold.
Output:
  1) $ARS = \{AR_1, AR_2, ..., AR_n\}$ - a set of
     WUL-based association access pattern
     rules, where $AR_i = (X_i \Rightarrow Y_i, sup, conf)$,
     $X_i, Y_i \subset M$, and $X_i \cap Y_i = \emptyset$.
Process:
  1) Initialize $ARS = \emptyset$.
  2) For each $N_i \in NL$ with $P_i \neq \emptyset$ and $sup =
     \frac{|A_i|}{|G|} \geq MinSup$, do
       a) If $|P_i| = 1$ and $B_{i1} \neq \emptyset$, then
          Insert $((B_i - B_{i1}) \Rightarrow B_{i1}, sup, 100\%)$
          into $ARS$ as a WUL-based exact
          rule.
       b) For each $N_{ij} \in P_i$ with $B_{ij} \neq \emptyset$ and
          $conf = \frac{|A_i|}{|A_{ij}|} \geq MinConf$, do
          Insert $(B_{ij} \Rightarrow (B_i - B_{ij}), sup, conf)$ into
          $ARS$ as a WUL-based approximate
          rule.
  3) Return $ARS$.
```

Using the WUL-mine algorithm, all WUL-based association access pattern rules can be mined from the Web Usage Lattice given in Figure 4. Table 3 shows the mining results with $MinSup$ = 40% and $MinConf$ = 50%. A total of 12 WUL-based association access pattern rules including 3 exact rules and 9 approximate rules are generated. In addition, we have also mined the association access pattern rules using the Apriori-based algorithm [16] from the web access sessions given in Table 1. A total of 32 rules are generated. Among them, there are 9 exact rules and 23 approximate rules.

V. Performance Analysis

In this section, we evaluate the performance of the proposed WUL-based mining approach based on our personalized recommender system *AWARS*. The performance is measured in comparison with the Apriori-based association rule mining approach [16]. We evaluate the performance based on efficiency and quality. The efficiency evaluation measures the runtime of the online recommendation process. In quality evaluation, three performance measures, namely precision, satisfaction and applicability, are used to evaluate the quality of recommendation services provided by the recommender system.

TABLE III

WUL-based association access pattern rules mined from the Web Usage Lattice in Figure 4 ($MinSup$ = 40%, $MinConf$ = 50%)

No.	WUL-based Association Access Pattern Rules	Support	Confidence
1	$P2 \Rightarrow P6$	60%	100%
2	$P4 \Rightarrow P3$	60%	100%
3	$P1 \Rightarrow P2 \wedge P6$	40%	100%
4	$P6 \Rightarrow P2$	60%	75%
5	$P6 \Rightarrow P3$	60%	75%
6	$P3 \Rightarrow P6$	60%	75%
7	$P3 \Rightarrow P4$	60%	75%
8	$P2 \wedge P6 \Rightarrow P1$	40%	67%
9	$P2 \wedge P6 \Rightarrow P3$	40%	67%
10	$P3 \wedge P6 \Rightarrow P2$	40%	67%
11	$P3 \wedge P6 \Rightarrow P4$	40%	67%
12	$P3 \wedge P4 \Rightarrow P6$	40%	67%

A. Experimental Setup

In the experiment, the proposed web recommender system was implemented in C++. The experiment was conducted on a 1.6 GHz Intel Pentium 4 PC machine with 384 MB memory, running Microsoft Windows 2000 Professional. We have used two datasets from Microsoft Anonymous Web Data [31] for mining association access pattern rules and testing the web recommender system. These two datasets consist of a collection of sessions with each session containing a sequence of web page references. The Microsoft Anonymous Web Data records the pages within www.microsoft.com that each user visited in a one-week time frame during February 1998. The training dataset for constructing the Web Usage Lattice has a total of 5,000 web access sessions, with each session containing from 1 up to 35 page references from a total of 294 pages. Note that we only use the 2,213 valid web access sessions that have more than two items. The testing dataset has a total of 32,711 web access sessions including 8,969 valid web access sequences.

B. Efficiency Evaluation

In this experiment, we have mined the Apriori-based association access pattern rules and WUL-based association access pattern rules, with minimum support count set to 10, from the training dataset to generate recommendation rules for the testing dataset. The numbers of WUL-based association access pattern rules (WUL) and Apriori-based association access pattern rules (AAR) with different minimum confidence values (from 10% to 90%) are shown in Figure 5. As shown in the figure, the number of the mined WUL-based association access pattern rules generated is much less than that of the rules

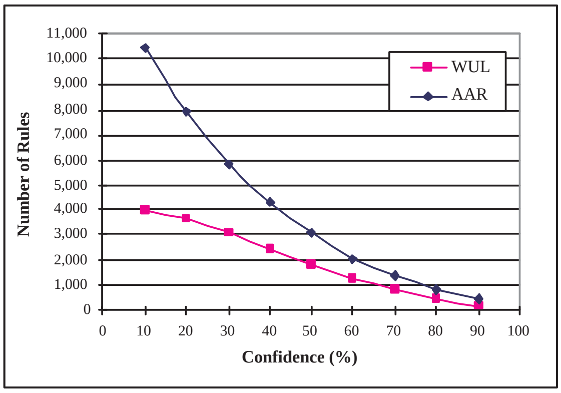


Fig. 5. Numbers of Apriori-based rules and WUL-based rules with different confidence values.

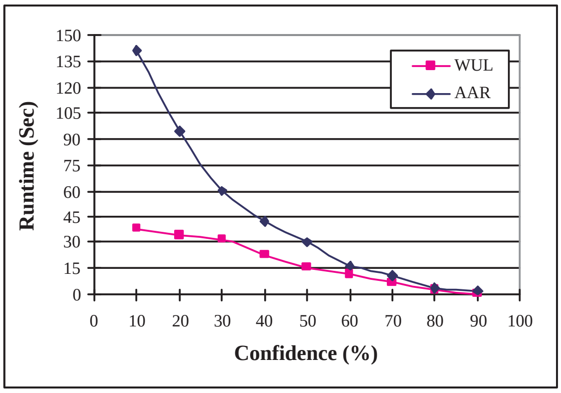


Fig. 6. Runtime of the recommendation rule generation using Apriori-based rules and WUL-based rules with different confidence values.

mined with the Apriori-based algorithm when the minimum confidence gets smaller.

In addition, the runtime of the recommendation rule generation process in the recommender system using WUL-based association access pattern rules and Apriori-based association access pattern rules with different minimum confidence values (from 10% to 90%) is shown in Figure 6. The experimental results have shown that the runtime based on Apriori-based rules increases sharply, when the confidence threshold decreases. Moreover, the runtime based on the WUL-based rules is always less than that based on the Apriori-based rules.

Comparing the results shown in Figure 5 and Figure 6, we have observed that the runtime for recommendation rule generation increases according to the number of association access pattern rules. In addition, the online recommendation process is much more efficient when using WUL-based rules, as the number of rules generated is much less than the Apriori-based rules, and the cost of matching the appropriate rules for recommendation is reduced significantly.

C. Quality Evaluation

For evaluating the quality of recommendation services provided by *AWARS*, we use the precision, satisfaction and applicability measures, which are defined as follows.

Definition 5.1 Let N be the total number of recommendation rules and N_c be the number of all correct recommendation rules, which include the immediate next page that the user has accessed. The precision measure of web recommendation is defined as

$$precision = \frac{N_c}{N}.$$

The precision measure evaluates how probable a user will access one of the recommended pages.

Definition 5.2 Let N_s be the number of all satisfactory recommendation rules, which include any pages that the user has accessed during subsequent browsing activities. The satisfaction of the web recommendation is defined as

$$satisfaction = \frac{N_s}{N}.$$

Satisfaction is a very important evaluation measure for web recommendation. Actually, the next web page accessed by a user may not be the target page that the user wants. In many situations, a user has to access some intermediate pages before reaching the target page. Therefore, it is inappropriate if we only use the precision measure to evaluate the performance of web recommendation. The satisfaction measure gives the precision that the recommended pages will be accessed in the near future.

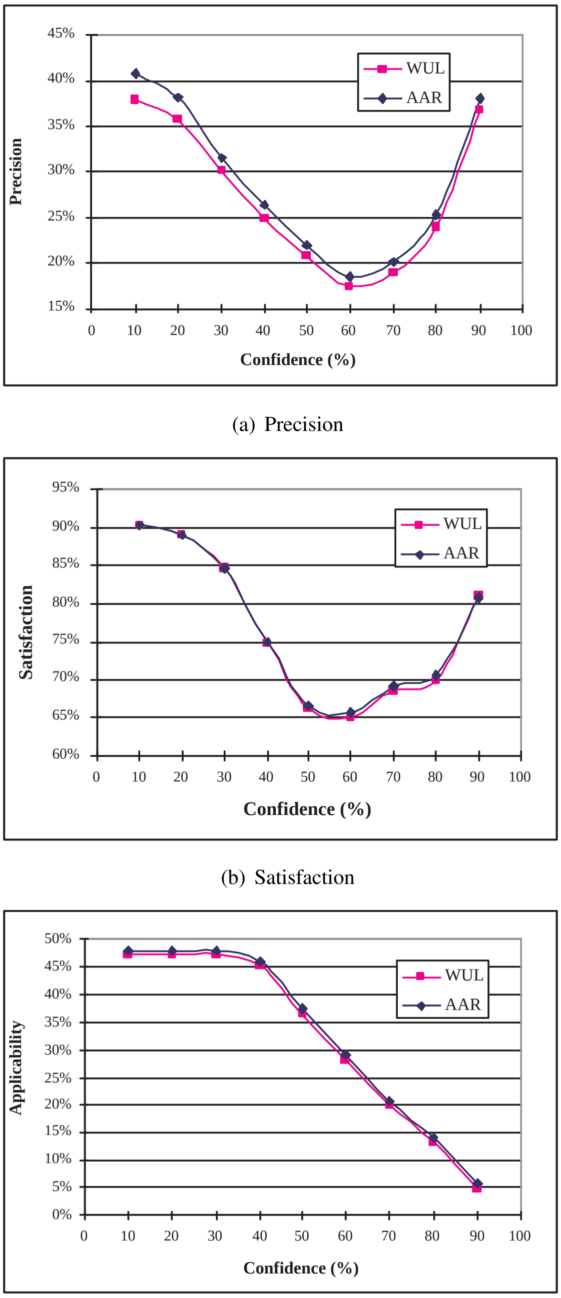
Definition 5.3 Let N_n be the number of all nonempty recommendation rules. The applicability measure of web recommendation is defined as

$$applicability = \frac{N_n}{N}.$$

The association access pattern rules only store web access event sets accessed frequently by users (with a support of at least $MinSup$ and a confidence of at least $MinConf$). If no matching rule is found during recommendation, then the generated recommendation rule set is empty. Therefore, the applicability measure evaluates how often recommendations will be generated. Some parameters such as $MinSup$ and $MinConf$ in the proposed approach can affect the applicability measure of web recommendation. Generally, the smaller the $MinSup$ and $MinConf$ are, the more applicable the web recommendation is. But, this comes at the expense of increased runtime on association access pattern rule mining.

Figure 7 gives the performance results in terms of the precision, satisfaction and applicability measures. Figure 7(a) shows the precision measure of recommendation using the WUL-based rules which is a bit lower than that of using the Apriori-based rules. Figure 7(b) shows the satisfaction measure of recommendation using both kinds of rules which give more or less the same values. Figure 7(c) shows the applicability measure of recommendation in which the WUL-based rules have slightly lower applicability than the Apriori-based rules.

In summary, the performance results have shown that the WUL-based association access pattern rules have comparable quality for web recommendation with the rules mined using the Apriori-based algorithm (as shown in Figure 7), but with much less number of rules generated (as shown in Figure 5). Therefore, the proposed WUL-based mining algorithm has



(a) Precision

(b) Satisfaction

(c) Applicability

Fig. 7. Quality measures of WUL-based rules and Apriori-based rules for web recommendation.

achieved better performance in terms of efficiency for online recommendations (as shown in Figure 6).

VI. Conclusions

In this paper, we have proposed a Web Usage Lattice-based mining approach for personalized web recommendation. The proposed approach constructs the Web Usage Lattice from the original web logs, and then uses the WUL-mine algorithm to discover association access pattern rules. We have incorporated the WUL-based mining approach into our personalized web recommender system known as *AWARS*. We have measured the performance of the proposed WUL-based mining approach in comparison with the Apriori-based algorithms based on efficiency and quality. In efficiency evaluation, we have shown that the proposed WUL-based approach has generated much less number of association access pattern rules and achieved faster runtime for online recommendation. The quality evaluation is based on measuring the precision, satisfactory and applicability on the *AWARS* recommender system. The results have shown that association access pattern rules generated by the WUL-based approach have achieved comparable quality with the Apriori-based access pattern rules.

Acknowledgment

We would like to thank Singapore Millennium Foundation Ltd (http://www.smf-scholar.org) for providing the SMF Scholarship to support Baoyao Zhou's research work.

References

[1] M. Eirinaki, and M. Vazirgiannis (2003) Web Mining for Web Personalization. *ACM Transactions on Internet Technology*, **3**(1): 1–27.

[2] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl (1997) GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, **40**(3): 77–87.

[3] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry (1992) Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, **35**(12): 61–70.

[4] T.W. Yan and H. Garcia-Molina (1999) The SIFT Information Dissemination System. *ACM Transactions on Database Systems*, **24**(4): 529–565.

[5] T. Joachims, D. Freitag, and T. Mitchel (1997) WebWatcher: A Tour Guide for the World Wide Web. *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pp. 770–775, Japan, 1997.

[6] C. Shahabi, F. Banaei-Kashani, Y. Chen, and D. McLeod (2001) Yoda: An Accurate and Scalable Web-based Recommendation System. *Proceedings of the 6th International Conference on Cooperative Information Systems*, pp. 418–432, Trento, Italy, 2001.

[7] H. Ishikawa, T. Nakajima, T. Mizuhara, S.Yokoyama, J. Nakayama, M. Ohta, and K.Katayama (2002) An Intelligent Web Recommendation System: A Web Usage Mining Approach. *Proceedings of the 13th International Symposium on Foundations of Intelligent Systems*, pp. 342–350, 2002.

[8] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan (2000) Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *ACM SIGKDD Explorations*, **1**(2): 12–23.

[9] W. Lin, S.A. Alvarez, and C. Ruiz (2000) Collaborative Recommendation via Adaptive Association Rule Mining. *Proceedings of the Web Mining for E-Commerce Workshop (WebKDD 2000)*, Boston, 2000.

[10] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa (2001) Effective Personalization Based on Association Rule Discovery from Web Usage Data. *Proceedings of the 3rd International Workshop on Web Information and Data Management*, pp. 9–15, Atlanta, Georgia, USA, 2001.

[11] B.Y. Zhou, S.C. Hui, and K.Y. Chang (2004) An Intelligent Recommender System using Sequential Web Access Patterns. *Proceedings of 2004 IEEE Conference on Cybernetics and Intelligent Systems*, Singapore, 2004.

[12] B. Mobasher (1999) A Web Personalization Engine based on User Transaction Clustering. *Proceedings of the 9th Workshop on Information Technologies and Systems*, 1999.

[13] D.S. Phatak, and R. Mulvaney (2002) Clustering for Personalized Mobile Web Usage. *Proceedings of the IEEE FUZZ'02*, pp. 705-710, Hawaii, 2002.

[14] S.Y. Chen, X. Liu (2005) Data Mining from 1994 to 2004: An Application-orientated Review. *International Journal of Business Intelligence and Data Mining*, **1**(1): 4–21.

[15] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa (2002) Using Sequential and Non-sequential Patterns for Predictive Web Usage Mining Tasks. *Proceedings of the IEEE International Conference on Data Mining*, Maebashi City, Japan, 2002.

[16] R. Agrawal, and R. Srikant (1994) Fast Algorithms for Mining Association Rules. *Proceedings of ACM International Conference on Very Large Database*, pp. 478–499, 1994.

[17] J. Han, J. Pei, and Y. Yin (2000) Mining Frequent Patterns without Candidate Generation. *Proceedings of ACM International Conference on the Management of Data*, pp. 1–12, 2000.

[18] H.C. Tjioe, and D. Taniar (2005) Mining Association Rules in Data Warehouses. *International Journal of Data Warehousing and Mining*, **1**(3): 28–62.

[19] M.Z. Ashrafi, D. Taniar, and K.A. Smith (2005) PPDAM: Privacy Preserving Distributed Association Rule Mining Algorithm. *International Journal of Intelligent Information Technologies*, **1**(1): 49–69.

[20] R. Wille (1982) *Restructuring Lattice Theory: An Approach based on Hierarchies of Concepts, Ordered sets*, pp. 455–470. Boston-Dordrecht: Reidel.

[21] B. Ganter, and R. Wille (1999) *Formal Concept Analysis: Mathematical Foundations*, Springer, Heidelberg.

[22] K. Sato, A. Ohtaguro, M. Nakashima, and T. Ito (2005) The Effect of a Website Directory When Employed in Browsing the Results of a Search Engine. *International Journal of Web Information Systems*, **1**(1): 43–51.

[23] H. Stormer (2005) Personalized Websites for Mobile Devices using Dynamic Cascading Style Sheets. *International Journal of Web Information Systems* **1**(2): 83–88.

[24] J. Goh, and D. Taniar (2005) Mining Parallel Patterns from Mobile Users. I*nternational Journal of Business Data Communications and Networking*, **1**(1): 50–76.

[25] O. Nasraoui, and C. Petenes (2003) An Intelligent Web Recommendation Engine Based on Fuzzy Approximate Reasoning. *Proceedings of the IEEE International Conference on Fuzzy Systems - Special Track on Fuzzy Logic and the Internet*, St. Louis, MO, 2003.

[26] R. Sarukkai (2000) Link Prediction and Path Analysis Using Markov Chains. *Proceedings of the 9th International World Wide Web Conference*, 2000.

[27] H. Lieberman (1995) Letizia: An Agent That Assists Web Browsing. *Proceedings of 1995 International Joint Conference on Artificial Intelligence*, ontreal, CA, USA, 1995.

[28] M. Pazzani, J. Muramatsu, and D. Billsus (1996) Syskill & Webert: Identifying Interesting Web Sites. *Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 54–61, 1996.

[29] J. Rucker and M.J. Polano (1997) Siteseer: Personalized Navigation for the Web. *Communications of the ACM*, **40**(3): 73–25.

[30] R. Cooley, B. Mobasher, and J. Srivastava (1999) Data Preparation for Mining World Wide Web Browsing Patterns. *Journal of Knowledge and Information Systems*, **1**(1).

[31] S. Hettich, and S. D. Bay (1999) The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science. Available: http://kdd.ics.uci.edu/databases/msweb/msweb.html

Baoyao Zhou Baoyao Zhou is a postgraduate student by research for Ph.D. degree in the School of Computer Engineering at Nanyang Technoligical University, Singapore. His current research interests include web usage mining, Semantic Web, data mining, and AI. He received his B.Eng. degree in Automation in 1999 and M.Sc. Degree in Control Theory and Control Engineering in 2002 from Tsinghua University, China. He was an intern (visiting student) in Media Management Group, Microsoft Research Asia from Jun. 2000 to Jan. 2002.

Siu Cheung Hui S. C. Hui is an Associate Professor in the School of Computer Engineering at Nanyang Technological University, Singapore. His current research interests include data mining, Internet technology, and multimedia systems. Previously, he worked in IBM China / Hong Kong as a system engineer from 1987 to 1990. He received his B.Sc. degree in Mathematics in 1983 and a D. Phil degree in Computer Science in 1987 from the University of Sussex, UK. Dr. Hui is a member of IEEE and ACM.

Alvis C. M. Fong A.C.M. Fong is currently Assistant Professor in the School of Computer Engineering at Nanyang Technological University, Singapore. His research interests include various aspects of Internet technology, information theory, and video and image signal processing. Previously, he was with the Motorola Corporate Research and Technology Center. He received his degrees from the University of Auckland and Imperial College, London. Dr. Fong is a member of IEEE and IEE, and is a Chartered Engineer.