Communications of the IIMA

2006

Design and Implementation of a Web Usage Mining Model Based On Upgrowth and Preflxspan

Hengshan Wang
University of Shanghai

Cheng Yang
University of Shanghai

Hua Zeng
University of Shanghai

Follow this and additional works at: http://scholarworks.lib.csusb.edu/ciima

Part of the Management Information Systems Commons

Recommended Citation

Design and Implementation of a Web Usage Mining Model Based On Fpgrowth and Prefixspan

Hengshan Wang
Business School, University of Shanghai for Science and Technology
Shanghai 200093, P. R. China, wanghs@usst.edu.cn

Cheng Yang
Business School, University of Shanghai for Science and Technology
Shanghai 200093, P. R. China, ryan.yang@ebaotech.com

Hua Zeng
Business School, University of Shanghai for Science and Technology
Shanghai 200093, P. R. China, zeng_9830@163.com

ABSTRACT

Web Usage Mining (WUM) integrates the techniques of two popular research fields - Data Mining and the Internet. By analyzing the potential rules hidden in web logs, WUM helps personalize the delivery of web content and improve web design, customer satisfaction and user navigation through pre-fetching and caching. This paper introduces two prevalent data mining algorithms - FPgrowth and PrefixSpan into WUM and they are applied in a real business case. Maximum Forward Path (MFP) is also used in the web usage mining model during sequential pattern mining along with PrefixSpan so as to reduce the interference of "false visit" caused by browser cache and raise the accuracy of mining frequent traversal paths. Detailed analysis and application on the corresponding results are discussed.

Keywords : Data Mining, Web Usage Mining, Association Rules, Sequential Pattern, FPgrowth, PrefixSpan

INTRODUCTION

The World Wide Web is an immense source of data that can come either from the Web content, represented by the billions of pages publicly available, or from the Web usage, represented by the log information daily collected by all the servers around the world. Web Mining is that area of Data Mining which deals with the extraction of interesting knowledge from the World Wide Web (Etzioni, 1996). More precisely, Web Content Mining is that part of Web Mining which focuses on the raw information available in Web pages; source data mainly consist of textual data in Web pages (e.g., words, but also tags); typical applications are content-based categorization and content-based ranking of Web pages (Kosala et al., 2000). Web Structure Mining is that part of Web Mining which focuses on the structure of Web sites; source data mainly consist of the structural information present in Web pages (e.g., links to other pages); typical applications are link-based categorization of Web pages, ranking of Web pages through a combination of content and structure (Brin et al., 1998), and reverse engineering of Web site models. Web Usage Mining is that part of Web Mining which deals with the extraction of knowledge from server log files; source data mainly consist of the (textual) logs that are collected when users access Web servers and might be represented in standard formats (e.g., Common Log Format, Extended Log Format, LogML) (Punin et al., 2002); typical applications are those based on user modeling techniques, such as Web personalization, adaptive Web sites, and user modeling. Figure 1 shows the main application areas of WUM.
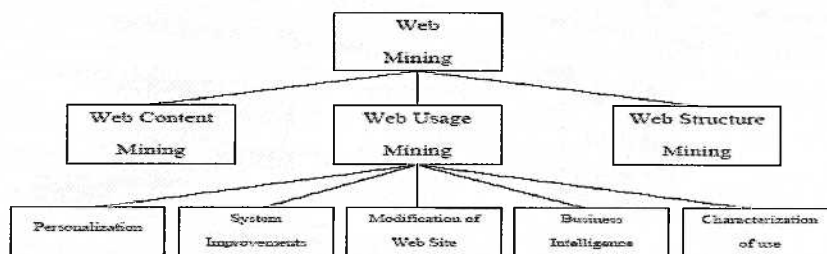
Figure 1: Application areas of web usage mining

Srivastava et al. (2000) systematically discuss the development of WUM and classify the content of WUM. Zhang and Liang (2004) emphasize the importance of data preprocessing in Web Usage Mining and present an algorithm called "USIA" which boasts high efficiency. Wang and Meinel (2004) point out that user behaviors recovery and pattern definition play more important roles in web mining than other applications so they give a new insight on behavior recovery and complicated pattern definition. As current Web Usage Mining applications rely exclusively on the web server log files, Guo et al. (2005) propose a system that integrates Web page clustering into log file association mining and use the cluster labels as Web page content indicators in the hope of mining novel and interesting association rules from the combined data source.

Association rules are probably the most elementary data mining technique and, at the same time, the most used technique in Web Usage Mining. Apriori algorithm proposed by Agrawal et al. (1994) is a classical algorithm in association rule mining. However, it has some problems in efficiency. Thus many improved algorithms are advanced, such as Direct Hashing and Pruning algorithm (Park et al., 1995); Partition algorithm (Savasere et al., 1995); Sampling (Toivonen, 1996); Dynamic Itemset Counting algorithm (Brin et al., 1997) and so on. In addition to these improved algorithms based on Apriori, a great deal of work has also been done on studying association-rule-mining methods based on completely different ideas and data expression. Shemoy at al. (2000) propose Vertical Itemset Partitioning for Efficient Rule-extraction algorithm which adopts a kind of vertical data format. Agarwal et al. (2001) propose a Tree Projection algorithm which employs Lexicographic tree to store data and projects database transaction into the tree. Han et al. propose a novel frequent-pattern tree (FP-tree) structure, which is an extended prefix-tree structure for storing compressed, crucial information about frequent patterns, and develop an efficient FP-tree based mining method, FP-growth (2000). Study shows that the FP-growth method is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than the Apriori algorithm and also faster than some other mining methods.

Sequential pattern mining is a very complicated issue in data mining which aims at discovering frequent sub-sequences in a sequence database. The problem is more difficult than association rule mining because the patterns are formed not only by combinations of items but also by permutations of item-sets.

There are essentially two classes of algorithms that are used to extract sequential patterns: one includes methods based on association rule mining; the other one includes methods based on the use of tree structures and Markov chains to represent navigation patterns. Some well-known algorithms for mining association rules have been modified to extract sequential patterns. For instance, AprioriAll and GSP, two extensions of the Apriori algorithm are used for association rules mining (Huang et al., 2002; Han, 2001). Pei et al. (2000) argue that algorithms for association rule mining (e.g., Apriori) are not efficient when applied to long sequential patterns, which is an important drawback when working with Web logs. Accordingly, they propose an alternative algorithm in which tree structures (WAP-tree) are used to represent navigation patterns. The algorithm (WAP-mine) and the data structure (WAP-tree) are specifically tailored for mining Web access patterns. Mortazavi (2001) provides a comparison of different three sequential pattern algorithms applied to Web Usage Mining. The comparison includes (i) PSP+, an evolution of GSP, based on candidate generation and test heuristics, (ii) FreeSpan , based on the integration of frequent sequence mining and frequent pattern mining, and (iii) PrefixSpan that uses an approach based on data projection. The results of the comparison show that PrefixSpan outperforms the other two algorithms and offers very good performance even on long sequences.

In this paper, two prevalent data mining algorithms - FPgrowth and PrefixSpan are introduced into WUM so as to mind association rule and sequential pattern more efficiently and accurately. A web usage mining model with the students employment website at University of Shanghai for science and Technology (91.usst.edu.cn) is used as the research object. Detailed interpretation, analysis and application on the mining result are given.

The remainder of the paper is organized as follows: Section 2 provides a brief overview on FPgrowth and PrefixSpan, which is a prerequisite for the research. In Section 3, the design of the WUM model based on Fpgrowth and PrefixSpan is exemplified. In Section 4, the preprocessing work is introduced step by step. In Section 5, the association rule mining based on FPgrowth is conducted. In Section 5, the access sequence pattern mining based on PrefixSpan is conduct. In Section 6, the detailed pattern analysis is give. Section 7 concludes this paper with suggestions for future research.

BACKGROUND

Association Rule and FPgrowth

Given a set of transactions where each transaction is a set of items (itemset), an association rule implies the form $X \rightarrow Y$, where X and Y are itemset; X and Y are called the body and the head, respectively. The support for the association rule $X \Rightarrow Y$ is the percentage of transactions that contain both itemset X and Y among all transactions. The confidence for the rule $X \rightarrow Y$ is the percentage of transactions that contain itemset Y among transaction that contain itemset X. The support represents the usefulness of the discovered rule and the confidence represents certainty of the rule.

Association rule mining is the discovery of all association rules that are above a user-specified minimum support and minimum confidence. Apriori algorithm is one of the prevalent techniques used to find association rules (Agrawal, Imielinski, & Swami, 1993; Agrawal & Srikant, 1994). Apriori operates in two phases. In the first phase, all item-sets with minimum support (frequent item-sets) are generated. This phase utilizes the downward closure property of support. In other words, if an item-set of size k is a frequent item-set, then all the item-sets below (k - 1) size must also be frequent item-sets. Using this property, candidate item-sets of size k are generated from the set of frequent item-sets of size (k - 1) by imposing the constraint that all subsets of size (k - 1) of any candidate item-set must be present in the set of frequent item-sets of size (k - 1). The second phase of the algorithm generates rules from the set of all frequent item-sets.

The Apriori heuristic achieves good performance gained by (possibly significantly) reducing the size of candidate sets. However, in situations with a large number of frequent patterns, long patterns, or quite low minimum support thresholds, an Apriori-like algorithm may suffer from the following two nontrivial costs:

-- It is costly to handle a huge number of candidate sets. For example, if there are 104 frequent 1-itemsets, the Apriori algorithm will need to generate more than 107 length-2 candidates and accumulate and test their occurrence frequencies. Moreover, to discover a frequent pattern of size 100, such as {a1, . . . , a100}, it must generate $2100 - 2 \approx 1030$ candidates in total. This is the inherent cost of candidate generation, no matter what implementation technique is applied.

-- It is tedious to repeatedly scan the database and check a large set of candidates by pattern matching, which is especially true for mining long patterns.

In order to overcome the drawback inherited in Apriori, J.Han develop an efficient FP-treebased mining method, FP-growth,which contains two phases, where the first phase constructs an FPtree, and the second phase recursively projects the FPtree and outputs all frequent patterns.

Definition 1 (FP-tree). A frequent-pattern tree (or FP-tree in short) is a tree structure defined below.

1. It consists of one root labeled as "null", a set of item-prefix subtrees as the children of the root, and a frequent-item-header table.

2. Each node in the item-prefix sub-tree consists of three fields: item-name, count, and node-link, where item-name registers which item this node represents, count registers the number of transactions represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree carrying the same item-name, or null if there is none.

3. Each entry in the frequent-item-header table consists of two fields, (1) item-name and (2) head of node-link (a pointer pointing to the first node in the FP-tree carrying the item-name).

Based on this definition, we have the following FP-tree construction algorithm.

Algorithm 1 (FP-tree construction).

Input: A transaction database DB and a minimum support threshold ξ .
Output: FP-tree, the frequent-pattern tree of DB.

Method: The FP-tree is constructed as follows.
1. Scan the transaction database DB once. Collect F, the set of frequent items, and the support of each frequent item. Sort F in support-descending order as FList, the list of frequent items.
2. Create the root of an FP-tree, T, and label it as "null". For each transaction Trans in DB do the following. Select the frequent items in Trans and sort them according to the order of FList. Let the sorted frequent-item list in Trans be [p | P], where p is the first element and P is the remaining list. Call insert tree ([p | P], T). The function insert tree ([p | P], T) is performed as follows. If T has a child N such that N.item-name = p.item-name, then increment N's count by 1; else create a new node N, with its count initialized to 1, its parent link linked to T , and its node-link linked to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert tree (P, N) recursively

Algorithm 2 (FP-growth: Mining frequent patterns with FP-tree by pattern fragment growth).

Input: A database DB, represented by FP-tree constructed according to Algorithm 1, and
a minimum support threshold ξ .
Output: The complete set of frequent patterns.

Method: call FP-growth (FP-tree, null).
Procedure FP-growth (Tree, α)
{
(1)   if Tree contains a single prefix path   // Mining single prefix-path FP-tree
(2)   then {
(3)   let P be the single prefix-path part of Tree;
(4)   let Q be the multipath part with the top branching node replaced by a null root;
(5)   for each combination (denoted as β) of the nodes in the path P do
(6)   generate pattern β $\cup$ α with support = minimum support of nodes in β;
(7)   let freq_pattern_set(P) be the set of patterns so generated; }
(8)   else let Q be Tree;
(9)   for each item ai in Q do {   // Mining multipath FP-tree

(10)   generate pattern β= ai □ α with support = ai .support;

(11)   construct β's conditional pattern-base and then β's conditional FP-tree Treeβ ;

(12)   :f Treeβ = □

(13)   then call FP-growth(Treeβ, β);

(14)   let freq_pattern_set(Q) be the set of patterns so generated; }

(15)   return (freq_pattern_set(P) □ freq_pattern_set (Q) □ (freq_pattern_set (P)×freq_pattern_set (Q)))

Sequential Pattern and PrefixSpan

In Web Usage Mining, sequential patterns are exploited to find sequential navigation patterns that appear in users_ sessions frequently. Sequential patterns might appear syntactically similar to association rules; in fact algorithms to extract association rules can also be used for sequential pattern mining. However, sequential patterns include the notion of time, i.e., at which point of the sequence a certain event happened.

PrefixSpan is an efficient algorithm for mining frequent sequences (Pei et al., 2001). PrefixSpan mines frequent sequences by intermediate database generation instead of the tradition approach of candidate sequence generation. PrefixSpan is shown to be efficient if sufficient amount of memory is available. Before we review the algorithm, let us first introduce several terminologies defined by Pei et al. (2001).

• **prefix**. Given two sequences $s_1 = <t_1, t_2, \ldots, t_n>$, $s_2 = <t_1', t_2', \ldots, t_m'>(m \leq n)$, s_2 is called a prefix of s_1 if (1) t_i = t_i' for $i \leq m - 1$; and (2) all items in $(t_m - t_m')$ are alphabetically ordered after those in t_m'.
For example, if $s_1 = <\{a\}, \{b, c, d\}, \{e\}>$, $s_2 = <\{a\}, \{b\}>$, $s_3 = <\{a\}, \{d\}>$, then s_2 is a prefix of s_1, but s_3 is not.

• **projection**. Given a sequence s_1 and one of its subsequences s_2 (i.e., $s_2 \sqsubseteq s_1$), a sequence p is called the projection

of s_1 w.r.t. prefix s_2 if (1) $P \sqsubseteq s_1$; (2) s_2 is a prefix of p; (3) p is the "maximal" sequence that satisfies conditions (1)

and (2), that is, $\not\exists$ p', s.t.($p \sqsubseteq p' \sqsubseteq s1$)   (p≠p')   ($s_2$ is a prefix of p').
For example, if $s1 = <\{a\}, \{b, c, d\}, \{e\}, \{ f \}>$, $s2 = <\{a\}, \{c, d\}>$, then $p = <\{a\}, \{c, d\}, \{e\}, \{ f \}>$ is the projection of s1 w.r.t. prefix s2.

• postfix. If p is the projection of s1 w.r.t. prefix s2, then s3 obtained by removing the prefix s2 from p is called the postfix of s1 w.r.t. prefix s2.
For example, if $s1 = <\{a\}, \{b, c, d\}, \{e\}, \{ f \}>$, $s2 = <\{a\}, \{c, d\}>$, then $p = <\{a\}, \{c, d\}, \{e\}, \{ f \}>$, is the projection of s1 w.r.t. prefix s2 and the postfix of s1 w.r.t. prefix s2 is $<\{e\}, \{ f \}>$.
If s2 is not a subsequence of s1, then both the projection and the postfix of s1 w.r.t. s2 are empty.

There are three major steps of PrefixSpan.
• Find frequent length-1 sequences
In this step, PrefixSpan scans the database D once to find all frequent items. The set of frequent length-1 sequences is $L1 = \{<\{ \}>|i$ is a frequent item$\}$.
• Divide search space into smaller subspaces.The set of all frequent sequences can be divided into several groups, such that the sequences within a group share the same prefix item.
For example, if {A, B, E} is the set of frequent items discovered in the first step, then all the frequent sequences can be divided into three groups, corresponding to the three prefixes $<\{A\}>$, $<\{B\}>$, and $<\{E\}>$.
• Discover frequent sequences in each subspace.In this step, PrefixSpan finds frequent sequences in each sub-space.

DESIGN OF THE WUM MODEL BASED ON FPGROWTH AND PREFIXSPAN

The WUM model mainly consists of three functional models, namely data preparation and preprocessing model, pattern discovery model and pattern analysis model, as illustrated in figure 2.
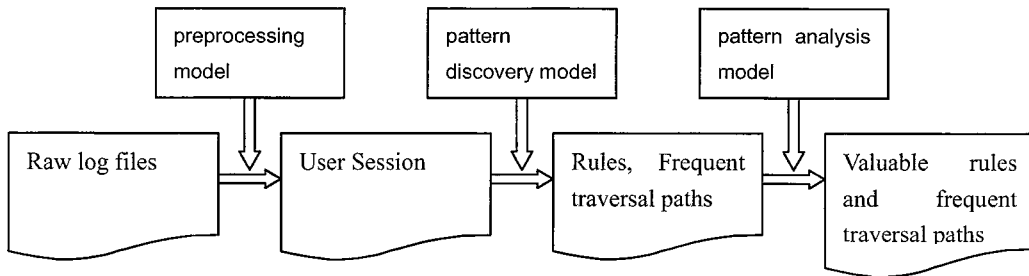

Figure 2: Main functional models in the WUM model

The functions of the three models are interpreted as follows:

Data preparation and preprocessing model: analyze the characteristics of page design; collect page content, web site linking structure and log records in server; preprocess log files, pages and structure; store processed data, which would be used as input in the next phase.

Pattern discovery model: conduct association rule mining based on FPgrowth so as to discover pertinence among pages; conduct sequential pattern mining based on PrefixSpan so as to discover frequent traversal paths from sequences of visited pages.

Pattern analysis model: find out valuable mining results and give a detailed interpretation according to the discovered association rules and frequent traversal paths and other analysis about the website.

PREPROCESSING

Preprocessing is a very crucial step during the whole process as its result has a direct impact on the rules and patterns generated by mining algorithms. The purpose of preprocessing is to transit various input such as content, structure and usage information into the format which data mining algorithms can handle easily (Han et al., 2000).

The logs we research are of W3C Extended Log File Format under IIS5.0 environment, containing attributes such as date, time, c-ip, cs-username, s-ip, s-port, cs-method, cs-uri-query, cs-uri-query, sc-status and cs(User-Agent). Log preprocessing includes data collection, data cleaning, user identification, session identification and path completion. We collect log files from the server of the students employment website at University of Shanghai for Science and Technology (91.usst.edu.cn) during the period between November 1 and 30, 2004. The total size of the Web log files is about 208 MB. We use database software Access and Java Programming Language to implement the preprocessing work.

Data Cleaning

This step consists of removing all the data that are useless for mining purposes e.g., images (bmp, gif, jpg); script (js, class); css (cascading style sheet) file and the irrelevant DHTML and multimedia information. As page content is the research object, we mainly analyze pages with the postfix such as html, htm and a few primary DHTML. In the experiment, we get 85429 cleaned log records from 797817 introduced ones by using SQL language to screen the records. All of these log records are in the form of html, tm and asp.

User Identification

Log records should be distinguished according to users for the analysis in the next stage, namely user identification. W3C Extended Log File Format under IIS5.0 environment gives the cs-ip and cs (User-Agent) fields to represent the visiting user's information. In the experiment, we get the cs-ip and cs (User-Agent) of all the visiting users by using SQL language. If the user's IP address is the same as the proxy information, we regard that's a single user. We filtrate illegal IP field, false user who can not be recognized and scarcely-visiting users (less than three times). In the experiment, we totally identify 10048 users through user identification.

However, the task of efficiently identifying users in the complicated web environment is overwhelming so the above rules can't ensure identifying users correctly.

Session Identification

A user may visit the same website many times during a certain period. Session identification aims at dividing the multi-visiting user sessions into single ones. The easiest solution is overtime technique. If the pages are requested at a time interval larger than a specified time, we think the user begins a new session. The default setting is 30 minutes. In our experiment, we adopt the default setting.

Path Completion

Due to the limitation of mined log format, we don't introduce information field. So the main task in path completion is to find whether there is hyperlink between the previous page and following page. If there isn't any hyperlink, we have to complement the page which hyperlinks both of them according to hyperlink information. If there exist many pages which meet the requirement, we can choose the one which appears in the user's visiting record history with the most recent requesting time. If there is no page meeting the requirement, we regard it is a multi-user instance so the user session should be divided into two different ones.

In the experiment, we transit the log records into user session through session identification. After conducting data cleaning, user identification, session identification and path completion, we totally get 12418 user sessions

Data Integration and Store

All of the preprocessing work has been done in the Access database as we can store various data properly and handle data conveniently by making use of database system and database management system respectively. The content we store includes user session path datasheet, page information datasheet, page hyperlink structure datasheet, user information datasheet and so on, as illustrated in Figure 3.

ASSOCIATION RULE MINING

When a user browses a website, some non-linked pages may often be visited at the same time, which suggests some relationship may exist among these pages either in content or in structure. That is called pertinence of the pages. By analyzing the pertinence, we can improve page content and hyperlink structure so that a user can browse the website more conveniently. Association-rule model can help us precisely find out these kinds of pages.
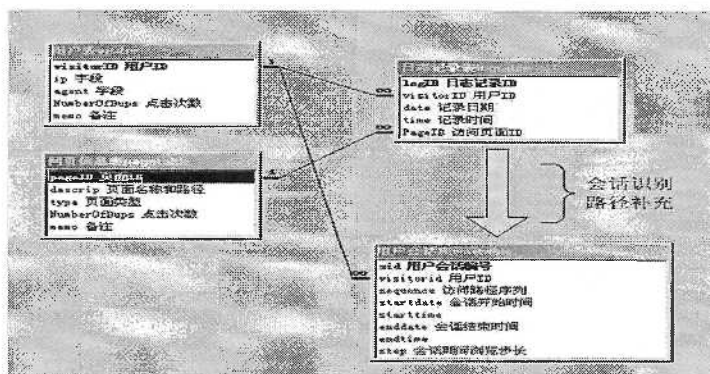
Figure 3: The Structure Design of Main Data Sheets

In the experiment, we use FPgrowth algorithm to implement association-rule mining. The input data is user session which records the browsed pages during each user's visiting period.

Data Preparation before Applying Algorithm

The preprocessed table still can't be used in association-rule mining as repetitive items should first be filtrated before data mining. Figure 4 shows the data files ready for mining algorithm after repetitive items have been filtrated.

Figure 4:   Data Files After Repetitive Items Have Been Filtrated

Experiment Step

Step 1: Input data to handle. After preprocessing, what the log database stores are the browsed page paths within each user session. Here repetitive pages should be filtrated.
Step 2: Mind all of the frequent item-sets which meet the given support by applying FPgrowth.
Step 3: Find out the frequent 2 item-set with the highest support and confidence, namely get the page set with certain pertinence. The step is realized by JAVA and SQL programming.

Mining Result

In FPgrowth, the threshold has a significant impact on the quantity of frequent item-set. Table 1 shows different mining results under different thresholds. In the experiment, we choose 500 as the threshold.

Item-set length	threshold = 120	threshold = 200	threshold = 500
1	49	40	22
2	210	115	32
3	269	106	12
4	125	34	0
5	19	2	0

Table 1: Statistics of Frequent Item-Set Result with Different Threshold

Figure 5 shows all of the 2 frequent item-sets and their confidence and support with 500 as the threshold. The result is hard to understand because we have not made value analysis. In 7.2, we will further handle the result so as to make it more understandable.

setID	item1	item2	abs_support
1	36	57	5317
2	36	69	2482
3	57	69	2393
4	36	60	1801
5	57	60	1038
6	60	69	559
7	36	100	1584
8	57	100	1218
9	69	100	558
10	7	36	1053
11	7	57	514
12	36	84	996
13	36	83	935
14	57	83	603
15	36	65	860
16	57	65	519
17	36	82	805
18	57	82	558
19	36	83	768
20	57	63	590
21	60	63	584
22	36	151	748
23	36	64	728
24	64	65	554
25	36	283	635
26	36	62	624

Figure 5:   Frequent 2 Item-Set Association Rules

ACCESS SEQUENCE PATTERN MINING

In user session, the browsed pages will be recorded in the log files according to the click sequence. This kind of information can be used to form Access Sequence after preprocessing. By analyzing the characteristics of these sequences, we can better understand users' browsing habits so as to predict users' next action and offer personalized website content and service based on corresponding forecast.

To reduce the interference of "false visit" caused by browser cache and raise the accuracy of mining frequent traversal paths, we introduce Maximum Forward Path (MFP) in the experiment.

MFP

MFP contains paths from the first page in user session to the previous page of "BACK". MFP algorithm is described as follows:

```
For each user session {
j=2; i=2;
Flag=true;
While (i<=m) {
If (xj==yk) for some i<=k<j {
        If (flag==true) output {y1,...yj-1} as MFP;
        j=k+1; i=i+1;
        Flag=false;
}    //for
else {
yj = xj;j=j+1;i=i+1;
flag = true;
}
}    //while
If (flag==true) output {y1,...yj-1} as MFP;
}    //For
```

This algorithm is used to find MFP in user session. Here we suppose $\{x1,...xm\}$ represents a user session, $\{y1,...,yj-1\}$ represents a sequence containing a potential MFP, flag indicates whether the traversal direction is backward or forward.

According to the above description, we design a program to find out the MFP in wastable. In the experiment, we get a total of 26085 MFP. Thus the problem of mining frequent traversal path is transited into finding out sequence pattern in all of the MFP.

Experiment Step

Step 1: Apply MFP algorithm to the log files after preprocessing so as to find out all of the maximum forward paths and get the MFP datasheet.
Step 2: Apply PrefixSpan to discover the consistent subsequence whose support is bigger than the given value in all of the maximum forward paths in user session.

Mining Result

Similar with association rules, in PrefixSpan, the threshold determines the amount of the frequent traversal paths. Here we choose 120 as the threshold. We totally find 397 frequent traversal paths with length varying from 2 to 5. Table 2 shows a part of the mining result.

MINING RESULT ANALYSIS

Here we have two tasks: 1) identify and interpret the mining result, 2) analyze the practical value of rules and sequential patterns.

Identification and Interpretation On The Mining Result

Through Java programming, we introduce the mining result to the database and implement data transition. Through operations such as multi-sheet query in page database generated in preprocessing, we get understandable page association rules and data views of sequential pattern, as illustrated in Figure 6.

The analysis and interpretation on association-rule mining results are shown in table 3.Through SQL language and multi-sheet query, we can not only get the specific names of pages corresponding to the abstract numeric data (Figure 5) in mining result but also calculate the confidence and support between the associated pages X and Y. The

ID	Item-set	length	abs_support
1	00360057006000620069	5	190
2	00360057006000630100	5	156
3	00360060006100620063	5	154
4	00360004002001280057	5	137
5	00360057006300690100	5	121
6	00360057006900990100	5	146
7	00360057006000610062	5	174
8	0036005701530283	4	146
9	0036005700600066	4	120
10	0036028300250007	4	159
11	0036005700640283	4	130
12	0036006300650283	4	147
13	0036006800250007	4	198
14	0036005700620069	4	246
⋮	⋮	⋮	⋮

Table 2:   Frequent Traversal Paths (a part)

interpretation on frequent traversal path result clearly shows the name, path length and support of the pages, as illustrated in Table 4.

Value Analysis

After transition and interpretation, the mining results become more understandable. However, that's far from enough. Though parameters such as support and confidence in mining algorithm ensure the numeric value, further analysis is needed to ensure these rules and patterns have practical value in real instance.

In the experiment, we find the following mining results have little value:

(i) the result is too evident: e.g., record (3) in table 3. The support and confidence between index page and invite_info page are relatively high (56.49% and 59.94% respectively) but there originally exists a direct link between the two pages as the content of invite_info is information about companies and positions. This evident association rule has little practical value so it could be ignored in value analysis. Record (8) is another case in point.

(ii) Paths are too short: When traversing frequent traversal paths, very short paths can also be ignored as they are too simple and evident, with little research value. In the experiment, we only research the longest and sub-longest paths, namely paths with length of 4 or 5.
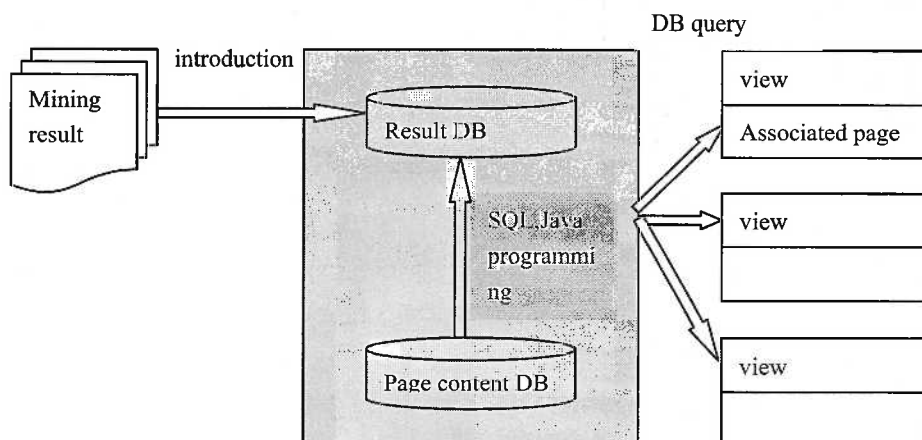

Figure 6: The Design for Result Transiting Process

(iii) The mining results are not practical: e.g.: page X and Y of record (10) in table 3 are both DHTML so their relationship can't be decided at all. Thus, there is no need for us to analyze the corresponding association rule. Record (5) is another case in point.

After filtrating the rules and patterns of no value, the remaining ones need to be further analyzed in order to serve as a basis for website designers when they analyze page using condition and optimize page structure.

(i) Pages with similar content could be combined into one: e.g.: the page meet/20031114_1.htm of record (7) is about campus employment information, which has a great correlation with the employing query page more_info.asp in content. Considering the browsing habit of users, the two pages could be integrated into one. By analogy, some information pages could be combined with dynamic query pages.

Rule NO.	pageX	page Y	XYconfidence	XYsupport

			%	%
1	/download/download.htm	/index.asp	97.32	11.19
2	/meet/20031121_1.htm	/meet/20031121_2.htm	72.32	5.89
3	/index.asp	/invite_info.asp	59.94	56.49
4	/download/download.htm	/invite_info.asp	47.50	5.46
5	/invite_info.asp	/more_info.asp	42.29	25.42
6	/meet/20031114_1.htm	/meet/20031114_4.htm	33.89	5.99
7	/meet/20031114_1.htm	/more_info.asp	33.59	5.94
8	/index.asp	/more_info.asp	27.98	26.37
9	/invite_info.asp	/probation/invite_info.asp	21.52	12.94
10	/more_info.asp	/probation/invite_info.asp	20.30	5.91
11	/invite_info.asp	/meet/20031114_1.htm	18.34	11.03
24	/index.asp	/student/index.htm	8.43	7.95
25	/index.asp	/meet/20031121_1.htm	8.21	7.73
26	/index.asp	/zhaopin/shile.htm	7.16	6.75

Table 3: Associated Page XY After Interpretation (a part)

Path number	Frequent traversal path	length	support%
1	index.asp→invite_info.asp→meet/20031114_1.htm→meet/20031114_3.htm →more_info.asp	5	0.73
2	index.asp→invite_info.asp→meet/20031114_1.htm→meet/20031114_4.htm→probation/invite_info.asp	5	0.59
3	index.asp→meet/20031114_1.htm→meet/20031114_2.htm→meet/20031114_3.htm→ meet/20031114_4.htm	5	0.59
4	index.asp→invite_info.asp→ company/index.asp→graduate/graduate.htm→spec/glgongye.htm	5	0.53
8	index.asp→invite_info.asp→meet/20031121_1.htm→zhaopin/shile.htm	4	0.57
10	index.asp→zhaopin/shile.htm→guide/guide.htm→download/download.htm	4	0.59
12	ndex.asp→meet/20031114_4.htm→meet/20031121_2.htm→zhaopin/shile.htm	4	0.56
13	index.asp→meet/more.htm→guide/guide.htm→download/download.htm	4	0.76

Table 4: Frequent Traversal Path After Interpretation (a part)

(ii) Associated pages with high support but no links could be linked together so as to attract more users to browse: e.g.: the pages about campus employment information meet/20031114_1.htm and meet/20031114_4.htm are often browsed by users at the same time so they could be linked together. Record (2) is another case in point.

(iii) Some pages in frequent traversal paths don't have direct hyperlink (e.g.: guide/guide.htm and download/download.htm in record (10) in table 4), but they are frequently visited by users so hyperlinks could be added. Thus, a user can browse them conveniently.

(iv) Page pre-fetching and caching mechanism could be set up by referring valuable frequent traversal sequences and applying other website designing tools. Take record (4) in table 4 is an example. When we find a user has visited index.asp, invite_info.asp, company/index.asp accordingly by website testing tool, we could preload the page graduate/graduate.htm and spec/glgongye.htm into cache, thus accelerating the speed of response of the server and enhancing user efficiency.

Besides the methods mentioned above, there are many other valuable assistant techniques such as data visualization, online analytical processing, which can also help analyze the mining results.

CONCLUSION AND FUTURE WORK

The general goal of Web Usage Mining is to gather interesting information about users navigation patterns (i.e., to characterize Web users). This information can be exploited later to improve the Website from the users' viewpoint. The results produced by the mining of Web logs can be used for various purposes: (i) to personalize the delivery of Web content; (ii) to improve user navigation through pre-fetching and caching; (iii) to improve Web design or e-commerce sites , and (iv) to improve the customer satisfaction.

After reviewing related theories about WUM technology, we design and implement a WUM model based on association rule and sequential pattern mining. Two prevalent mining algorithms, FPgrowth and PrefixSpan are used during the process so as to enhance efficiency and accuracy. Based on predetermined rules, we conduct a series of preprocessing work, introduce MFP during web access sequence pattern mining in order to reduce interference, analyze the mining result and offer valuable suggestions on the improvement of the website. As web access pattern becomes complicated and diversified, log recording format in the experiment is not complete and the log files on proxy and client ends are not available, which suggests that the user identification and path completion process still need improving.

Future research may focus on the following:

1) the research about data handling techniques on collection and structure-transition of web data of various kinds (Supriya et al., 2004).

2) the research about web mining methods and pattern identification techniques for self-adaptive websites and intelligent websites to provide personalized service and performance optimization.

3) the research about dynamic maintaining, updating of web knowledge base and comprehensive evaluating methods for various knowledge and patterns (MartAAn, 2006).

4) the research about the technology on intelligent searching engine with high efficiency and automatic navigation function based on web mining and information retrieving (Baowen,2001).

5) the research about highly-efficient mining algorithms for semi-structural or structural text data, graphic or image data, multimedia data.

6) the research about data mining language specially designed for knowledge discovery and its corresponding standardization.

7) the research about multi-tier data system structure and intelligent integrated system based on web with corresponding query language and optimizing and maintaining mechanism.

8) the research about the improvement on the existing data mining methods and techniques , the extension toward web data, the adaptive and timing performance of mining algorithms.

9) the research about pattern discovery within web files and its application in information extracting and text analysis (Toshiko,2004).

10) the research about the application of related web mining techniques in the field of e-commerce   (Asharfa et al., 2004).

REFERENCES

Agarwal, R C, Aggarwa, C C, Prasad VVV. (2001). A tree project ion algorithm for generation of frequent item sets. Journal of Parallel and Distributed Computing, March, 61 (3), 350-371.

Agrawal, R and Srikant, R. (1994). Fast algorithms for mining association rules, Proc. of the 20th international Conference on very large database, Chile, 487-499.

Asharfa ,S and Narasimha, M. (2004). A rough fuzzy approach to web usage categorization. Fuzzy Sets and Systems, 119-130

Brin S, Motwani R, and Ullman J D. (1997). Dynamic item set counting and implication rules for market basket data. SIGMOD Record (ACM Special Interest Group on Management of Data), 26 (2), 255.

Brin, S and Page,L. (1998). The anatomy of a large-scale hyper-textual Web search engine. Computer Networks and ISDN Systems, 30 (1–7), 107–117.

Etzioni, O. (1996). The world-wide Web: quagmire or gold mine? Communications of the ACM, 39 (11), 65–68.

Guo, Jiayun, Ke Šelj, Vlado, and Gao, Qigang. (2005). Integrating Web Content Clustering into Web Log Association Rule Mining, Advances in Artificial Intelligence: 18th Conference of the Canadian Society for Computational Studies of Intelligence, Canada, May 9-11, pp. 182

Han, J, Pei, J, and Yin, Y. (2000). Mining Frequent Patterns without Candidate Generation, in proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD' 00), Dallas, TX, May, 1-12.

Han, J, Pei, J, Mortazavi-Asl, B, Chen, Q, Dayal, U, and Hsu, M. (2000). FreeSpan: frequent pattern-projected sequential pattern mining, Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, August, 355-359.

Han, J, Pei, J, Mortazavi-Asl, B, Pinto, H, Chen, Q, and Dayal, U. (2004). Mining sequential patterns by patterngrowth: the PrefixSpan Approach, IEEE Transactions on Knowledge and Data Engineering, November, 1424-1440.

Han, J, Pei, J, Mortazavi-Asl, B, and Zhu, H. (2000). Mining access patterns recently from web logs, Pacifc-Asia Conference on Knowledge Discovery and Data Mining, pp. 396–407.

Huang, X and Cercone, N. (2002). Comparison of interestingness functions for learning web usage patterns, Proceedings of the Eleventh International Conference on Information and Knowledge Management, ACM Press, pp. 617–620.

J.R. Punin, M.S. Krishnamoorthy, and M.J. Zaki. (2001). LOGML - Log Markup Language for Web Usage

Mining, in WEBKDD Workshop 2001: Mining Log Data Across All Customer TouchPoints (with SIGKDD01), San Francisco, August, pp. 88–112.

Kosala, R and   Blockeel, H. (2000). Web mining research: a survey. SIGKDD Explorations, July, 2 (1), 1-15.

Long Wang. (2004). Christoph Meinel. Behaviour Recovery and Complicated Pattern Definition in Web Usage Mining. Web Engineering: 4th International Conference, ICWE 2004, Munich, Germany, July 26-30, pp. 531 – 543.

MartAAn-Guerrero. (2006). Studying the feasibility of a recommender in a citizen web portal based on user modeling and clustering algorithms. Expert Systems With Applications, 30 (2), 299-312

Park J S, Chen M -S, and Yu P S. (1995). An effective Hash-based algorithm for mining association rules, Proceedings of 1995 ACM-SIGMOD International Conference on Management of Data (S IGMOD'95). San Jo se, CA, 175-186.

Savasere A ,Omiecinski E, and Navathe S. (1995). An efficient algorithm for mining association rules in large databases . VLDB'95 , 432-443.

Shenoy P, Haritsa J R, Sudarshan S, et al. (2000). Turbo-charging vertical mining of large databases. SIGMOD Conference , 22-33.

Srivastava J , Cooley R , and Mukund Deshpanda. (2000). Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. SIGKDD Explorations, 1 (2), 12-23.

Supriya Kumar De, and Radha Krishna, P. (2004). Clustering web transactions using rough approximation. Fuzzy Sets and Systems, 131 ~ 138.

Toivonen H. (1996). Sampling Large Databases for Association Rules, Proceedings of 22th VLDB Conf. Bombay, India, 134-145.

Toshiko Wakaki. (2004). Rough Set-Aided Feature Selection for Automatic Web-Page Classification. IEEE WI, 70-76.

Xu, Baowen. (2001). A rough set based self-adaptive Web search engine. IEEE NSFC, October, 377-382.

Zhang Huiying and Liang, Wei. (2004). An intelligent algorithm of data pre-processing in Web usage mining, Proceedings of the World Congress on Intelligent Control and Automation (WCICA), v 4, WCICA, p3119-3123.