Rough Sets Clustering and Markov model for Web Access Prediction

Siriporn Chimphlee[1], Naomie Salim[2], Mohd Salihin Bin Ngadiman[3], Witcha Chimphlee[4], Surat Srinoy[5]

[1,4]*Faculty of Science and Technology*
Suan Dusit Rajabhat University, 295 Rajasrima Rd, Dusit, Bangkok, Thailand
Tel: (+66)-2445225, Fax: (+66) 6687136, Email: [1]siriporn_chi@dusit.ac.th,[4]witcha_chi@dusit.ac.th,[5]surat_sri@dusit.ac.th

[2,3]*Faculty of Computer Science and Information Systems,*
University Technology of Malaysia, 81310 Skudai, Johor, Malaysia
Tel: (607) - 5532070, Fax: (607) 5565044, Email: [2]naomie@fsksm.utm.my , [3]msn@fsksm.utm.my

Abstract—Discovering user access patterns from web access log is increasing the importance of information to build up adaptive web server according to the individual user's behavior. The variety of user behaviors on accessing information also grows, which has a great impact on the network utilization. In this paper, we present a rough set clustering to cluster web transactions from web access logs and using Markov model for next access prediction. Using this approach, users can effectively mine web log records to discover and predict access patterns. We perform experiments using real web trace logs collected from www.dusit.ac.th servers. In order to improve its prediction ration, the model includes a rough sets scheme in which search similarity measure to compute the similarity between two sequences using upper approximation.

Index Terms— Rough sets, Markov model, Prediction

1. Introduction

The web has become a primary conduit for the people to search information and communication. Because of the web is novel and dynamic. Most of the users use web browsers to navigate a web site and go into the web site follow the hyperlinks that they think relevant in the starting page and subsequent pages, until they have found the desired information in one or more pages [1] Web servers collect huge amount of data every day. Extracting the trails users actually follow and comparing them to intend site usage can help justification and optimization of a site link structure. Namely, it can suggest which links should be restructured. Being able to predict the next request from web server makes it possible to pre-send this web request to the user's computer. If predictions are frequently correct, the network latency perceived by users can be significantly reduced. Being able to extract users' browsing patterns and preferences can help facilitate and personalize users' web experience in terms of adaptive web pages and agent assisted navigation. It has been observed that users tend to repeat the trails they have followed once. So, better prediction of a users' next request could be made on the data pertaining to that particular user, not all of users. However, this would require reliable user identification and tracking users between sessions. This is usually achieved by sending cookies to a client browser, or by registering users. Both require user cooperation and might discourage some of potential site visitors. So many web sites choose not to use these means of user tracking. Also, building prediction models on individual data would require that users have accessed enough pages to make a prediction, which is not usually the case for a university web site that has many casual users.

This paper is organized as follows. In section 2, we discuss the background of study and review the past works in related research. In section 3, we present the rough sets clustering. In section 4, we discuss Markov model for prediction model. Section 5 shows experimental design. Section 6 evaluates our model through Experimental Setup and Results. We conclude our work in section 7.

2. Background of Study

A lot of previous work has focused on Web data clustering [2, 3]. Web data clustering is the process of grouping web data into "clusters" so that similar objects are in the same class and dissimilar objects are in different classes. Its goal is to organize data circulated over the web into groups. Vakali et al. [4] categorize web data clustering into two classes (I) users' sessions-based and (II) link-based. The former uses the web log data and tries to group together a set of users' navigation sessions having similar characteristics. In web log data provide information about activities performed by a user from the moment the user enters a web site to the moment the same user leaves it [5]. The records of users' actions within a web site are stored in a log file. Each record in the log file contains the client's IP address, the date and time the request is received, the requested object and some additional information -such as protocol of request, size of the

object etc. Figure 1 presents a sample of a web access log file from www.dusit.ac.th .The web log files are cleaned in data processing, such as invalid data cleaning and session identification [6]. Data cleaning removes log entries (e.g. images, JavaScript etc.) that are not needed for the mining process. In order to identify unique users' sessions, heuristic methods are (mainly) used [5], based on IP address and session time-outs. In this context, it is considered that a new session is created when a new IP address is encountered or if the visiting page time exceeds a time threshold (e.g. 30 minutes) for the same IP-address. Then, the original web logs are transferred into user access session datasets for analysis.

```
1102801060.863 1897600 172.16.1.98 TCP_IMS_HIT/
304 203 GET http://asclub.net/images/main_r4_c11.jpg -
NONE/- image/jpeg
1102801060.863 1933449 172.16.1.183 TCP_MISS/404
526 GET http://apl1.sci.kmitl.ac.th/robots.txt DIRECT/
161.246.13.86 text/html
1102801060.863 1933449 172.16.1.183 TCP_REFRESH_
HIT/200 3565 GET ttp://apl1.sci.kmitl.ac.th/wichitweb/
spibigled/spibigled.html - DIRECT/161.246.13.86 text/html
```

Figure 1: A sample of web access log

3. Rough Set Clustering

Rough sets were introduced by Zdzislaw Pawlak [7, 8] to provide a systemic framework for studying imprecise and insufficient knowledge. Rough sets are used to develop efficient heuristics searching for relevant tolerance relations that allow extracting interesting patterns in data. An attribute-oriented rough sets technique reduces the computational complexity of learning processes and eliminates the unimportant or irrelevant attributes so that the knowledge discovery in database or in experimental data sets can be efficiently learned. Using rough sets has been shown to be effective for revealing relationships within imprecise data, discovering dependencies among objects and attributes, evaluating the classificatory importance of attributes, removing data re-abundances, and generating decision rules [9]. Some classes, or categories, of objects in an information system cannot be distinguished in term of available attributes. They can only be roughly, or approximately, defined. The idea of rough sets is based on equivalence relations which partition a data set into equivalence classes, and consists of the approximation of a set by a pair of sets, called lower and upper approximations. The lower approximation of a given sets of attributes, can be classified as certainly belonging to the concept. The upper approximation of a set contains all objects that cannot be classified categorically as not belonging to the concept. The upper approximation of a set contains all objects that cannot be classified categorically as not belonging to the concept. A rough set also is defined as an approximation of a set, defined as a pair of sets: the upper and lower approximation of a set [10].

In this we uses rough agglomerative approach to cluster web user transactions because robust clustering methods are needed when user's browsing patterns are buried in data with significant noise and outlier components [11]. The rough set theory is one of the tools that deals with these types of problems for handing vagueness and uncertainly inherent in decision situations.

Our clustering is based on that described by De and Krishna [12]. The presented algorithm is based on the agglomerative method of clustering. A user transaction is a sequence of items. Let there be m users and the user transactions be

$$T = \{t_1, t_2, t_3, ,..., t_m\}$$

Let U be the set of distinct n clicks (hyperlinks/URLs) clicked by users. Let

$U = \{hl_1, hl_2, hl_3, ,..., hl_n\}$. Here, each

$t_i \in T$ is a non-empty subset of U. Here, the temporal order of user clicks within transactions has not been taken into account. A user transaction $t \in T$ can be represented as a vector

$$t = < u_1^t, u_2^t, u_3^t, ..., u_n^t >, \text{ where}$$

$$U_i^t = \begin{cases} 1 \; if \; hl_i \in t, \\ 0 \text{ otherwise.} \end{cases}$$

Given two transactions t and s, the measure of similarity between t and s is given by

$$sim(t,s) = \frac{|t \cap s|}{|t \cup s|}.$$

$sim(t,s) \in [0,1]$, $sim(t,s) = 1$, when two transactions t and s are exactly identical, $sim(t,s) = 0$, when two transactions t and s have no items in common. In general, the users' access related to their common areas of interest. Actually the navigation of any two users may not be exactly identical but may have some common interesting items. Moreover, the same user can navigate the same pattern in different ways. Considering a small dissimilarity between the two users, the transactions are meaningless. De and Krishna [12] analyzed by using a binary relation R defined on T. For any threshold value $th \in [0,1]$ and for any two user

transactions t and $s \in T$, a binary relation R on T denoted as tRs *is* defined by $tRs \ iff \ sim(t,s) \geq th.$ This relation R is a tolerance relation as R is both reflexive and symmetric but transitive may not hold good always.

Definition 1. The similarity class of t, denoted by $R(t)$, is the set of transactions which are similar to t. It is given by $R(t) = \{s \in T, sRt\}$.

For different threshold values we can get different similarity classes. A domain expert can choose the threshold based on this experience to get a proper similarity class. It is clear that for a fixed threshold $\in [0,1]$, a transaction form a given similarity class may be similar to an object of another similarity class.

Definition 2. Let $P \subset T$. For a fixed threshold $\in [0,1]$, a binary tolerance relation R is defined on T. The lower approximation of P, denoted by $\underline{R}(P)$ and the upper approximation of P, denoted by $\overline{R}(P)$ are respectively defined as follows:

$$\underline{R}(P) = \{t \in P, R(t) \subseteq P\} \text{ and } \overline{R}(P) = U_{t \in P} R(t).$$

A user transaction is a sequence of web pages. The training data of our example, shown in Figure 2, is from five user transactions. We use rough set to clustering web user transactions over the web.

S 1 : 100, 186, 194, 118
S 2 : 100, 168, 186
S 3 : 168, 186, 194, 118
S 4 : 194, 118, 119
S 5 : 168, 186, 118, 100

Figure 2:- User transaction from data set

4. Markov model and Sequential association rules for prediction model

We use technique that integrated Markov model and Sequential association rules for prediction model. The all k-th order Markov model has been proposed to intelligently combine the all k-th order Markov rules extracted from the historical training data set and provide predictions with a partial match method. In this approach, predictions are provided with the all k-th order Markov model by first matching the test instance against the highest order rules. If no matching rules are found, additional predictions are attempted by matching the latest consecutive subsequence of the test case with lower order rules, a cascading approach to matching. This cascading partial match prediction scheme, to improve both the prediction accuracy and coverage made Markov prediction model, has achieved considerable success in the web prefetching [13, 14, 15, 5].

The sequential association rules have been proposed for prediction model [16, 17]. A sequential association rule can be expressed as follow:

$X \Rightarrow Y$ [S, C] where S is the count support of the $X \cup Y$ and C is the confidence of the rule

$$C = \frac{Count\ X \cup Y}{Count(X)}.$$

However, the Markov model is overly restrictive because in reality the users may not follow a strict sequence while browsing a site. As a result, minor deviations in users' access (e.g. random access noise) may result in an unnecessary mismatch with the rules. By contrast, the sequential association rule based model does not predict a page that comes immediately after an access sequence. Instead an item set that is sequentially associated with an observed item set is predicted. Because the sequential association rules based model focuses on the sequentially related patterns of web page access instead of a consecutively accessed pattern, this approach is more flexible than the Markov model [18]. However, as the observed page items are loosely connected when providing predictions, less temporal properties (ie. the order of the observed access sequence) are considered within this sequential association rules based model. From the above discussion, we can see that combining the advantages of these two models would be a good approach to overcome their respective drawbacks. Wang [18] proposed hybrid prediction model, it is to combine the characteristics of all k-th order Markov and sequential association rules based prediction models. This hybrid model differs from the k-th order Markov model in that the predicted candidates do not need to have occurred consecutively after the k-length Markov sequences during the training sequence used to determine the rules. In other words, the predicted item is only sequentially associated with a k-length Markov sequence. A hybrid model emphasizes the effect that the user's ordered consecutive request sequence has on his or her future predicted access (a property inherited from the all k-th order Markov model). In addition, the approach provides more flexibility than the all k-th order Markov model when providing predictions (a property inherited through incorporating the association rules based prediction model). Chen [5] prediction candidate does not need to immediately follow the previous visited request sequence, is supported by the success of the popularity-based prediction model which uses a ranking of URL access patterns [5].

A rule derived from a model can be expressed as $X_1 \to X_2 \to ... \to X_k \Rightarrow X_l[S,C]$, where $X_1 \to X_2 \to ... \to X_k$ is a k-item Markov sequence,

X_l is the predicted item sequentially related with the k-item Markov sequence, S is the support of the rule and C is the confidence of the rule. The confidence of the hybrid Markov model is defined as,

$$Confidence(X_1 \to X_2 \to ...X_k \Rightarrow X_l) = \frac{count(X_1 \to X_2 \to ...X_k) \overset{u}{\cup} X_l)}{count(X_1 \to X_2 \to ...X_k)}$$

where $(X_1 \to X_2 \to ...X_k) \overset{u}{\cup} X_l)$, means $(X_1 \to X_2 \to ...X_k)$, and X_l occurred together with $(X_1 \to X_2 \to ...X_k)$ occurring before X_l.

From the example, we can see that a hybrid model both emphases the users ordered consecutive request sequence's effect on his or her future predicted access and it is more flexible than the all k-th order Markov model.

5. Experimental Design

Data preparation includes tasks such as data cleaning, completeness, correctness, attribute creation, attribute selection, and discretization. Data conversion must be performed on the initial data into a form in which specific rough set can be applied. The problem is solved in two steps: 1) web usage clustering from measured data; 2) User transactions clusters based on selected feature. As reflected in Figure. 3, central to the present work is the creation of descriptive production rules from given user sessions clusters. This relies on the use of sequential Markov method.

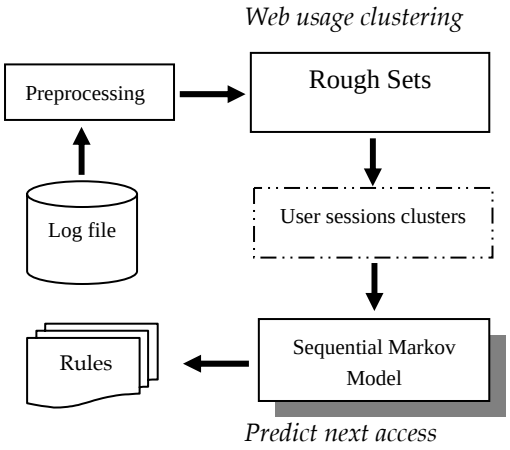

Figure 3- Our Model

The prediction models that we build are based on web log data that corresponds with users' behavior. They are used to make prediction for the general user and are not based on the data for a particular client. This prediction requires the discovery of a web users' sequential access patterns and using these patterns to make predictions of users' future access. We will then incorporate these predictions into the web prefetching system in an attempt to enhance the performance.

6. Experimental Setup and Results

The experiment used web data collected from www.dusit.ac.th web server (see example in Figure 1). The total number of web pages with unique URLs is equal to 314 URLs, and there are 13,062 web log records. These records are used to construct the user access sequences. The user session is split into training dataset and testing dataset. The training dataset is mined in order to extract rules, while the testing dataset is considered in order to evaluate the predictions made based on these rules.

Before processing to web mining on data set, raw data must be pre-processed in order to decrease the mining time and facilitate the effectiveness. Web log files contain a large amount of erroneous, misleading, and incomplete information. This step is to filter out irrelevant data and noisy log entries. Elimination of the items deemed irrelevant by checking the suffix of the URL name such as gif, jpeg, GIF, JPEG, jpg, JPG (preprocessing step in Figure 3). Since every time a Web browser downloads an HTML document on the Internet, several log entries such as graphics and script are downloaded too. In general, a user does not explicitly request all of the graphics that are in the web page, they are automatically down-loaded due to the HTML tags. Since web usage mining is interested in studying the user's behavior, it does not make sense to include file requests that a user does not explicitly request. The HTTP status code returned in unsuccessful requests because there may be bad links, missing or temporality inaccessible pages, or unauthorized request etc: 3xx, 4xx, and 5xx. Executions of CGI script, Applet, and other script codes. We also eliminated enough Meta data to map these requests into semantically meaningful actions, as these records are often too dynamic and contain insufficient information make sense to decision makers.

After the pre-processing, the log data are partitioned into user sessions based on IP and duration. Most users visit the web site more than once. The goal of session identification is to divide the page accesses of each user into individual sessions. The individual pages are grouped into semantically similar groups. A user session is defined as a relatively independent sequence of web requests accessed by the same user [19]. Fu et al. [20] identify a session by using a threshold idle time. If a user stays inactive for a period longer than the identified max_idle_time, subsequent page requests are considered to be in another episode,

thus another session. Most researchers use heuristic methods to identify the Web access sessions [21] based on IP address and a time-out not exceeding 30 minutes for the same IP Address. A new session is created when a new IP address is encountered after a timeout. Catledge and Pitkow [22] established a timeout of 25.5 minutes based on empirical data. In this research, we use IP address time-out of 30 minutes to generate a new session (Figure 2).

From Figure 2, we can be represented as vectors

$S_1 = \{1,1,1,0,0,1\}$; $S_2 = \{1,0,0,0,1,1\}$;
$S_3 = \{1,1,1,0,1,0\}$; $S_4 = \{0,1,1,1,0,0\}$;
$S_5 = \{1,0,1,0,1,1\}$;

The similarity classes are as follow:

$$sim(S_1, S_3) = \frac{3}{5} = 0.6,\ sim(S_1, S_4) = \frac{2}{5} = 0.4,$$
$$sim(S_1, S_5) = \frac{3}{5} = 0.6,\ sim(S_2, S_4) = \frac{0}{6} = 0,$$
$$sim(S_2, S_5) = \frac{3}{4} = 0.75,\ sim(S_3, S_4) = \frac{2}{5} = 0.4,$$
$$sim(S_3, S_5) = \frac{3}{5} = 0.6,\ \ sim(S_4, S_5) = \frac{1}{5} = 0.17$$

$\underline{R}(s_1) = \{s_1, s_3, s_5\}$, $\underline{R}(s_2) = \{s_2, s_5\}$,
$\underline{R}(s_3) = \{s_1, s_3, s_5\}$, $\underline{R}(s_4) = \{s_4\}$,
$\underline{R}(s_5) = \{s_1, s_2, s_3, s_5\}$

$\overline{R}(s_1) = \{s_1, s_3, s_5\}$, $\overline{R}(s_2) = \{s_2, s_5\}$,
$\overline{R}(s_3) = \{s_1, s_3, s_5\}$, $\overline{R}(s_4) = \{s_4\}$,
$\overline{R}(s_5) = \{s_1, s_2, s_3, s_5\}$

$\underline{\overline{R}}(s_1) = \{s_1, s_2, s_3, s_5\}$,
$\underline{\overline{R}}(s_2) = \{s_1, s_2, s_3, s_5\}$,
$\underline{\overline{R}}(s_3) = \{s_1, s_2, s_3, s_5\}$, $\underline{\overline{R}}(s_4) = \{s_4\}$,
$\underline{\overline{R}}(s_5) = \{s_1, s_2, s_3, s_5\}$

$\overline{\underline{R}}(s_1) = \{s_1, s_2, s_3, s_5\}$,
$\overline{\underline{R}}(s_2) = \{s_1, s_2, s_3, s_5\}$,
$\overline{\underline{R}}(s_3) = \{s_1, s_2, s_3, s_5\}$, $\overline{\underline{R}}(s_4) = \{s_4\}$,
$\overline{\underline{R}}(s_5) = \{s_1, s_2, s_3, s_5\}$

We see that two consecutive upper approximation for $\{s_1\},\{s_2\},\{s_3\},\{s_4\}$ *and* $\{s_5\}$ are same. We get the similarity upper approximation for $\{s_1\},\{s_2\},\{s_3\},\{s_4\}$ *and* $\{s_5\}$ as

$S_1 = \{s_1, s_2, s_3, s_5\}$, $S_2 = \{s_1, s_2, s_3, s_5\}$,
$S_3 = \{s_1, s_2, s_3, s_5\}$, $S_4 = \{s_4\}$,
$S_5 = \{s_1, s_2, s_3, s_5\}$

As $S_1 = S_2 = S_3 = S_5$ and $S_4 \neq S_j$ for i =1, 2, 3, 5, we get the two clusters: $\{s_1, s_2, s_3, s_5\}, \{s_4\}$. This denote that the user, who is visiting the hyperlinks as in s_1, may also visit the hyperlinks present in the sessions s_2, s_3 *and* s_5

7. Conclusions

Web servers keep track of web users' browsing behavior in web logs. From log file, one can builds statistical models that predict the users' next requests based on their current behavior. In this paper we studied different algorithm for web request prediction. In the future, we plan to use rough sets for prefetching to extract sequence rules.

Acknowledgement

This research was supported in part by grants from Suan Dusit Rajabhat University at Bangkok, Thailand.

References

[1] J. Zhu, Mining Web Site Link Structure for Adaptive Web Site Navigation and Search, PhD thesis, Faculty of Informatics, University of Ulster at Jordanstown, 2003.

[2]P. Baldi, P. Frasconi, and P. Smyth, Modeling the Internet and the Web Wiley, 2003.

[3]S. Chakrabarti, Mining the Web, Morgan Kaufmann, 2003.

[4] A. Vakali, J. Pokorny, and T. Dalamagas, An Overview of Web Data Clustering Practices, Proceeding of the EDBT Workshop on Cluster Web, Lecture Notes in Computer Science (LNCS) Series, Springer Verlag, Heraklion, Greece, March 2004, pp. 597-606.

[5] Z. Chen, A.Wai-Chee Fu, and F. Chi-Hung Tong, Optimal algorithms for finding user access sessions from very large Web logs. World Wide Web: Internet and Information Systems, 2003, pp. 259-279.

[6] R. Cooley, B. Mobasher, and J. Srivastava, Data preparation for mining World Wide Web browsing patterns Knowledge Information Systems, 1999, pp. 5-32.

[7] Z. Pawlak, Rough Sets, International Journal of

Information and Computer Science, Vol. 11, 1982, pp. 145-172.

[8]Z. Pawlak, Rough Sets-Theoretical Aspects of reasoning about Data, Kluwer Academic Publisher, Dordrecht, 1991.

[9] J. Stefanowski and K. Slowinski, Rough set as a tool for studying attribute dependencies in the urinary stones treatment data, In Rough Sets and Data Mining Analysis for Imprecise Data, London: Kluwer, 1997, pp. 177-195.

[10] K. Cios, Witold Pedrycz and Roman Swiniarski, Data Mining Method for Knowledge Discovery, London: Kluwer, 2000, pp. 27-66.

[11] A. Joshi, R. Krishnapuram, Robust fuzzy clustering methods to support web mining, Proceeding Workshop in Data Mining and Knowledge Discovery, SIGMOD, 1998, pp. 151–158.

[12] S. Kumar de and P. R. Krishna, Clustering Web Transactions using Rough Approximation, Journal of Fuzzy sets and systems, Vol. 148, 2004, pp. 131-138.

[13] T. Palpanas and A. Mendelzon, Web prefetching using partial match prediction , Proceedings of Web Caching Workshop, San Diego, California, March 1999.

[14] A. Nanopoulos, D. Katsaros, and U. Manolopoulos, Effective prediction of web-user accesses: A data mining approach, Proceeding of the Workshop WEBKDD, 2001.

[15] B. Mobasher, W. Dai, T. Luo, and M. Nakagawa, Using Sequential and Non-Sequential Patterns for Predictive Web Usage Mining Tasks, Proceedings of the IEEE International Conference on Data Mining (ICDM'2002), Maebashi City, Japan, December, 2002.

[16] J.Han, J. Pei, B. Mortazavi-Asi, Q. Chen, U. Dayal, and M.C. Hsu, Freespan: Frequent Pattern-Projected Sequential Pattern Mining, Proceedings of the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining, 2000, pp. 355—359.

[17] J. Pei, J. Han, B. Mortazavi-Asl, W. Pinto, Q. Chen, U. Dayal, and M. Hsu.
Prefixspan: Mining sequential patterns by prefix-projected growth In ICDE, 2001, pp. 215-224.

[18] Y.Wang, A Hybrid Markov Prediction Model for Web Prefetching, Master thesis, Department of Electrical and Computer Engineering, Calgary, Alberta, 2003.

[19] R.Cooley, P-N.Tan, J.Srivastava, Discovery of Interesting Usage Patterns from Web Data, In Springer-Verlag LNCS/LNAI series, 2000.

[20] Y. Fu, K. Sandhu, and M.Y. Shih, Clustering of Web Users Based on Access Patterns, Proc. of the 5th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Diego: Springer, 1999.

[21] G. Pallis, L. Angelis, and A. Vakali, Model-based cluster analysis for web users' sessions, Springer-Verlag Berlin Heideberg, 2005, pp. 219-227.

[22]L.Catledge, and J.E. Pitkow, Characterizing Browsing Behaviors on The World Wide Web, Computer networks and ISDN Systems, Vol.27, No.6, 1995.