Predicting Next Page Access by Markov Models and Association Rules on Web Log Data

Siriporn Chimphlee, Suan Dusit Rajabhat University, Thailand
Naomie Salim, University Technology of Malaysia
Mohd Salihin Bin Ngadiman, University Technology of Malaysia
Witcha Chimphlee, Suan Dusit Rajabhat University, Thailand
Surat Srinoy, Suan Dusit Rajabhat University, Thailand

Abstract

Mining user patterns of log file can provide significant and useful informative knowledge. A large amount of the research has been concentrated on trying to correctly predict the pages a user will request. This task requires the development of models that can predict a user's next request to a web server. In this paper, we propose a method for constructing first-order and second-order Markov models of Web site based on past visitor behavior compare with association rules technique. This algorithm has been used to cluster Web site with similar transition behaviors and compares the transition matrix to an optimal size for efficient used to further improve the efficiency of prediction. From this comparison we propose a best overall method and empirically test the proposed model on real web logs.

Keywords

Web mining, Markov Model, Association Rule, Prediction

1. Introduction

The rapid expansion of the World Wide Web has created an unprecedented opportunity to disseminate and gather information online. There is an increasing need to study web-user behavior to better serve the web users and increase the value of enterprises. One important data source for this study is the web-server log data that traces the user's web browsing actions. The web log data consists of sequences of URLs requested by different clients bearing different IP Addresses. Association rules can be used to decide the next likely web page requests based on significant statistical correlations. The result of accurate prediction can be used for recommending products to the customers, suggesting useful links, as well as pre-sending, pre-fetching and caching of web pages for reducing access latency (Yang et al., 2004). The work by Liu et al. (1998) and Wang et al. (2000) considered using association rules for prediction by

selecting rules based on confidence measures, but they did not consider the classifiers for sequential data (Yang et al., 2004). It has been observed that users tend to repeat the trails they have followed once (Pitkow and Pirolli, 1999). The better prediction of a user's next request could be made on the data pertaining to that particular user, not all the 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 website that has many casual users (Chakoula, 2000).

In the network system area, Markov chain models have been proposed for capturing browsing paths that occur frequently (Pitkow and Pirolli, 1999; Su et al., 2000). However, researchers in this area did not study the prediction models in the context of association rules, and they did not perform any comparison with other potential prediction models in a systematic way. As a result, it remains an open question how to construct the best association rule based prediction models for web log data (Yang et al., 2004).

This paper is organized as follows. In section 2, we discuss the background and review the past works in related research. In section 3, we present the experimental design. In section 4, we discuss the experimental result. We conclude our work in section 5.

2. Background of study

Nowadays, there are many commercial web site log analysis programmed that can summarize statistics, number of hits common into web site or overview of web page hits or accesses. It is useful information, but none progress toward to understanding of behavior of web users. This study can significant the behavior of web users and more specific draw the behavior based on discover models of their web u sage data, and use to predict the next access with high accuracy.

2.1 Web Mining

Web mining is the term of applying data mining techniques to automatically discover and extract useful information from the World Wide Web documents and services (Etzioni, 1996). The web mining system uses information determined from the history of the investigated web system. When valuable hidden knowledge about the system of interest has been discovered, this information can be incorporated into a decision support system to improve the performance of the system. Three major web mining methods are web content mining, web structure mining and web usage mining. Web content mining is the process of extracting knowledge from the content of documents or their descriptions. Web structure mining aims to generate structural summaries about web sites and web pages (Madria and Bhowmick, 1999). Web usage mining is to discover usage patterns from web data, in order to understand and better serve the needs of web-based application. It is an essential step to understand the users' navigation preferences in the study of the quality of an electronic commerce site. In fact, understanding the most likely access patterns of users allows the service provider to personalize and adapt the site's interface for the individual user, and to improve the site's structure.

2.2 Association rules

Association rules (Agraval and Srikant, 1994) were proposed to capture the co-occurrence of buying different items in a supermarket shopping. Association rule generation can be used to relate pages that are most often referenced together in a single server session (Srivastava et al., 2000). In the context of Web usage mining, association rules refer to set of pages that are accessed together with a support value exceeding some specified threshold. The association rules may also serves as a heuristic for prefetching documents in order to reduce user-perceived latency when loading a page from a remote site (Srivastava et al., 2000). These rules are used in order to reveal correlations between pages accessed together during a server session. Such rules indicate the possible relationship between pages that are often viewed together even if they are not directly connected, and can reveal associations between groups of users with specific interests. A transaction is a projection of a portion of the access log. In the work of Liu et al. (1998) and Wang et al. (2000) considered using association rules for prediction without consider sequential data. In the contrast, Lan et al. (1999) develops a specialized association rules mining algorithm to discover the prefetched documents. The counting of support is done differently than in Agrawal and Srikant (1994) since the ordering of documents is considered (Nanolopoulos et al., 2003). Only consecutive subsequences inside a user transaction are supported. For instance, the user transaction ABCD supports the subsequences: AB, BC, and CD. Yang et al. (2004) studied different association-rule based methods for web request prediction. In their analysis is based on a two dimensional picture and using real web logs as training and testing data. In this experiment found that the latest-substring produces the best prediction precision among all method and the pessimistic selection method is always the winner among the three rule-selection methods. In this work, we also used the latest-substring to represent the prediction rules.

2.3 Markov model

Markov models (Papoulis, 1991) have been used for studying and understanding stochastic processes, and were shown to be well-suited for modeling and predicting a user's browsing behavior on a web-site. In general, the input for these problems is the sequence of web-pages that were accessed by a user and the goal is to build Markov models that can be used to model and predict the web-page that the user will most likely access next. Padbanabham and Mogul (1996) use N-hop Markov models for improving pre-fetching strategies for web caches. Pitkow and Pirolli (1999) proposed a longest subsequence models as an alternative to the Markov model. Sarukkai (2000) use Markov models for predicting the next page accessed by the user. Cadez et al (2000) use Markov models for classifying browsing sessions into different categories. Markov model have been widely used to model user navigation on the web and predicting the action a user will take next given the sequence of actions he or she has already performed. For this type of problems, Markov models are represented by three parameters $< A, S, T >$, where A is the set of all possible *actions* that can be performed by the user; S is the set of all possible states for which the Markov model is built; and T is a $|S| \times |A|$ Transition Probability Matrix (TPM), where each entry t_{ij} corresponds to the probability of performing the action j when the process is in state i (Bestavros, 1996).

A Markov chain is a discrete-state random process in which the only state that influences the next state is the current state. To be more precise:

- Discrete-time Markov chain:

X_{n+1} depends only on X_n and not on any X_i, $1 \leq i < n$

$$\Pr[X_{n+1} = S_i \mid X_n = S_j, X_{n-1} = S_k, ..., X_1 = S_l] = \Pr[X_{n+1} = S_i \mid X_n = S_j] \quad (1)$$

This equation is referred to as the Markov property.

- Continuous-time Markov chain:

Consider a continuous-time random process in which the number of times the random variables X(t) change value (the process changes state) is finite or countable. Let $t_1,t_2,t_3,\ldots$be the times at which the process changes state. If we ignore how long the random process remains in a given state, we can view the sequence $\{X_{t_1}, X_{t_2}, X_{t_3}, ...\}$ as a discrete-time process embedded in the continuous-time process.

A continuous-time Markov chain is a continuous-time, discrete-state random process such that

(1) The embedded discrete-time process is a discrete-time Markov chain, and

(2) The time between state changes is a random variable with a memory-less distribution.

We describe a Markov chain as follows: We have a set of *states*, $S = \{s_1, s_2,\ldots,s_n\}$. The process starts in one of theses and moves successively from one state to another. Each move is called a *step*. If the chain is currently in state s_i, then it moves to state s_j at the next step with a probability denoted by p_{ij}, and this probability does not depend upon which states the chain was in before the current state.

The probabilities p_{ij} are called *transition probabilities*. The process can remain in the state it is in, and this occurs with probability p_{ii}. An initial probability distribution, defined on S, specifies the starting state. Usually this is done by specifying a particular state as the starting state.

3. Experimental design

In this paper, we study prediction models that prediction the user's next requests and compare prediction models by using association rules and Markov models for constructing the best prediction model. The prediction models that we build are based on web logs data that correspond with users' behavior. Therefore, they are used to make prediction for a general user and are not based on the data for a particular client. This 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 performance.

The experiment on the real data set is conducted to evaluate the performance of proposed framework. We used the web data, which collected from www.dusit.ac.th web server (figure 1) during 1 December 2004 – 31 December 2004. The total number of web pages with unique URLs is equal to 314 URLs. The web log records collected within 13,062 records are used to construct the user access sequences (figure 2). Each 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. We experimentally

evaluated the performance of the proposed approach: first-order markov model, second-order markov model, and association rule mining and construct the predictive model.

```
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
        http://apl1.sci.kmitl.ac.th/wichitweb/spibigled/spibigled.html - DIRECT/161.246.13.86 text/html
```

Figure 1: Web log data

3.1 Web log preprocessing

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. Because every time a Web browser downloads an HTML document on the Internet, several log entries since graphics and script are downloaded in HTML files 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. Web usage mining is interesting to study the user's behavior, it does not make sense to include file requests that the user did 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. Because it is not enough Meta data to map these requests into semantically meaningful actions, these records are often too dynamic and insufficiently information to make sense to decision makers.

3.2 Session identification

After removal, the log data are partitioned into user sessions based on IP and duration. The most of users visit the web site more than once time. 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 (Cooley et al., 2000). Fu et al. (1999) identify session by define the threshold idle time, if a user stays inactive for period longer than *max_idle_time*, subsequent page requests are considered to be in another episode, thus another session. Most of researcher use heuristic methods to identify the Web access sessions (Pallis et al., 2005) based on IP address and time-out does not exceed 30 minutes for the same IP Address, a new session is created when a new IP address is encountered or timeout. Catledge and Pitkow (1995) established a timeout of 25.5 minutes based on empirical data. In this research, we use IP address and time-out 30 minutes, if more than 30 minute then generate to new user (figure 2).

Session 1 : 900, 586, 594, 618
Session 2 : 900, 868, 586
Session 3 : 868, 586, 594, 618
Session 4 : 594, 618, 619
Session 5 : 868, 586, 618, 900

Figure2. User session from data set.

Assuming the access pattern of a certain type of user can be characterized by length of user transaction, and that the corresponding future access path is not only related to the last accessed URL. Therefore, users with relatively short transactions (e.g. 2-3 accesses per transaction) should be handled in a different way from users with long transactions (e.g. 10-15 accesses per transaction) (Wong et al., 2001). In this study, we propose a case definition design based on the transaction length. User transactions with lengths of less than 3 are removed because it too short to provide sufficient information for access path prediction (Wong et al., 2001).

3.3 Prediction using Association rules

The web log data is a sequence of entries recording which document was requested by a user. We extracted a prediction model based on the occurrence frequency and find the last-substring (Yang et al., 2004) of the W1. The last- substrings are in fact the suffix of string in W1 window. These rules not only take into account the order and adjacency information, but also the newness information about the LHS string. We used only the substring ending in the current time (which corresponds to the end of window W1) qualifies to be the LHS of a rule (Yang et al., 2004). For example, Table 1 shows the latest-substring rules.

Table 1. The latest-substring rules

W1	W2	The latest-substring rule
900, 586, 594	618	{594} ➔ 618
868, 586, 594	618	

From these rules, we extract sequential association rules of the form LHS□RHS from the session (Yang et al., 2004). The support and confidence are defined as follows:

$$supp = \frac{count\ (LHS \rightarrow RHS)}{number\ of\ sessions} \quad (2)$$

$$conf = \frac{count\ (LHS \rightarrow RHS)}{count\ (LHS)} \quad (3)$$

In the equation above, the function *count(Table)* returns the number of records in the log table, and *count(LHS)* returns the number of records that match the left-hand-side LHS of a rule.

3.4 Markov prediction model

The markov model has achieved considerable success in the web prefetching field (Pitkow and Pirolli, 1999; Deshpande and Karypis, 2001; Mobasher et al., 2002). However the limit of this approach in web prefetching is that only requested pages are considered. The state-space of the Markov model depends on the number of previous actions used in predicting the next action. The simplest Markov model predicts the next action by only looking at the last action performed by the user. In this model, also known as the first-order Markov model, each action that can be performed by a user corresponds to a state in the model (table 2). A somewhat more complicated model computes the predictions by looking at the last two actions performed by the user. It is called the second-order Markov model, and its states correspond to all possible pairs of actions that can be performed in sequence (table 3). This approach is generalized to the K^{th}-order Markov model, which computes the predictions by looking at the last K actions performed by the user, leading to a state-space that contains all possible sequences of K actions (Bestavros, 1996)

Table 2: Sample First-order Markov

1 st order	Support count					
Second item in sequence / First item in sequence	586	594	618	619	868	900
586	0	2	1	0	0	0
594	0	0	3	0	0	0
618	0	0	0	0	1	1
619	0	0	0	0	0	0
868	3	0	0	0	0	0
900	1	0	0	0	0	1

Table 3: Sample Second-order Markov

2 st order	Support count					
Second item in sequence / First item in sequence	586	594	618	619	868	900
586➔594	0	0	2	0	0	1
594➔619	0	0	0	1	0	0
868➔586	0	1	1	0	0	0
900➔868	1	1	0	0	0	0

Table 4: Prediction rule and confidence

Rule-selected	Prediction	confidence
586	594	2/3 = 67%
594	618	3/3 = 100%
868	586	3/3 = 100%
586➔594	618	2/3 = 67%

3.5 Rule –selection

In our goal is to output the best prediction on a class based on a given training set. In rule-representation methods, each session where the previous visited matches the case can give rise to more than one rule. Therefore, we need a way to select among all rules that apply. In a certain way, the rule-selection method compresses the rule set; if a rule is never applied, then it is removed from the rule set. The end result is that we will have a smaller rule set with higher quality. In this section, we will study a method for rule selection. In addition to the extracted rules, we also define a default rule, whose the predict page is the most popular page in the training web log and the previous is the empty set. When no other rules apply, the default rule is automatically applied. For a given set of rules and a given rule-selection method, the above rule set defines a classifier. With the classifier, we can make a prediction for any given case. For a test case that consists of a sequence of web page visits, the prediction for the next page visit is correct if the predict of the selected rule occurs in prediction set. In association rule mining area, a major method to construct a classifier from a collection of association rules is the most-confident selection method (Liu et al., 1998). The most confident selection method always chooses a rule with the highest confidence among all the applicable association rules, among all rules whose support values are above the minimum support threshold. For example, suppose that for a testing set and a previous sequence is (A, B, C). Using the most-confident rule selection method, we can find 3 rules which can be applied to this example, including:

Rule 1: (A, B, C) ➔ D with confidence 35%
Rule 2: (B, C ➔ E with confidence 60%
Rule 3: (C) ➔ F with confidence 50%

In this case, the confidence values of rule 1, rule 2 and rule 3 are 35%, 60% and 50%, respectively. Since Rule 2 has the highest confidence, the most-confident selection method will choose Rule 2, and predict E.

4. Experimental results & discussions

The most commonly used evaluation metrics are accuracy, precision, recall, and F-Score to evaluate their classifier. Deshpande and Karypis (2004) used several measures to compare different Markov model-based techniques for solving the next-symbol prediction problem: accuracy, number of states, coverage and model-accuracy. In Haruechaiyasak (2003) and Zhu et al. (2002) used precision and recall to evaluate the performance of method. The precision measure the accuracy of the predictive rule set when applied to the testing data set. The recall measures the coverage or the number of rules from the predictive rule set that match the incoming request (Haruechaiyasak, 2003). To evaluate classifiers used in this work, we apply precision and recall, which are calculated to understand the performance of the classification algorithm on the minority class. Based on the confusion matrix computed from the test results, several common performance metrics can be defined as in Table, where TN is the number of true negative samples; FP is false positive samples; FN is false negative samples; FP is true positive samples. Precision and recall can be defined in term of:

$$Precision = \frac{TP}{TP + FP} \qquad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

A confusion matrix (Kohavi and Provost, 1998) contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix. The following table shows the confusion matrix for a two class classifier.

The entries in the confusion matrix have the following meaning in the context of our study:

- *a* is the number of **correct** predictions that an instance is **negative**,
- *b* is the number of **incorrect** predictions that an instance is **positive**,
- *c* is the number of **incorrect** of predictions that an instance **negative**, and
- *d* is the number of **correct** predictions that an instance is **positive**.

Table 5: Confusion matrix

		Predicted	
		Negative	Positive
Actual	Negative	**a**	**b**
	Positive	**c**	**d**

Several standard terms have been defined for the 2 class matrix:

- The *accuracy (AC)* is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$AC = \frac{a + b}{a + b + c + d} \quad (6)$$

- The *recall* or *true positive rate (TP)* is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$TP = \frac{d}{c + d} \quad (7)$$

- The *false positive rate (FP)* is the proportion of negatives cases that were incorrectly classified as positive, as calculated using the equation:

$$TN = \frac{a}{a + b} \quad (8)$$

- The *false negative rate (FN)* is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation:

$$FN = \frac{c}{c + d} \quad (9)$$

- Finally, *precision (P)* is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$P = \frac{d}{b + d} \quad (10)$$

The accuracy determined using equation 6 may not be an adequate performance measure when the number of negative cases is much greater than the number of positive cases (Kubat et al., 1998).

4.1 Results

The results are plotted in figure 3 and show comparison of different algorithms. The experiments were conducted on dataset from *www.dusit.ac.th* web server. We evaluated our models using similar method to cross validation. We divided the web log as training data and testing data. As can be seen from the figure 3, the first-order Markov model gives the best prediction performance.

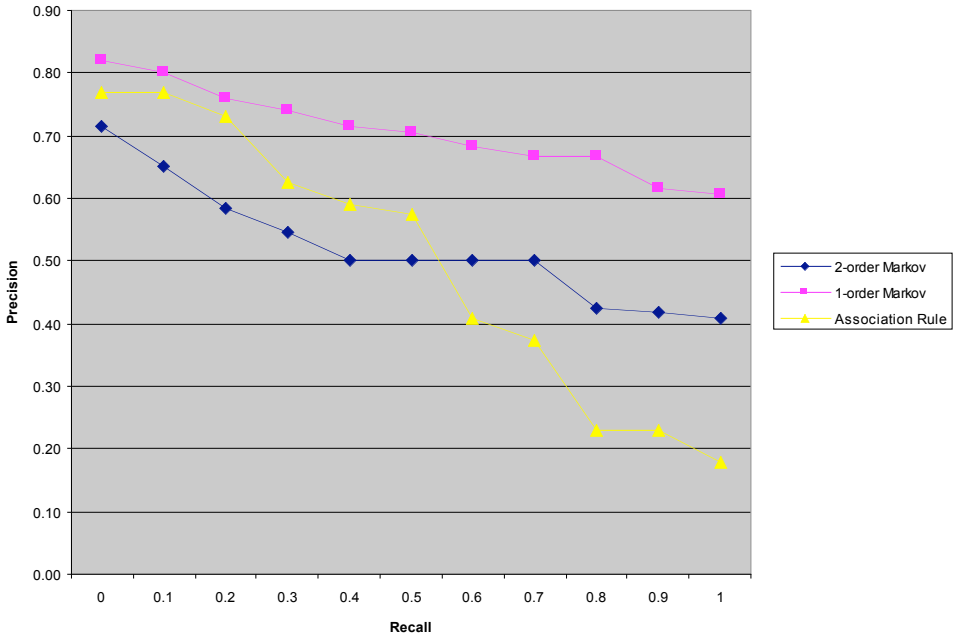

Figure 3. Result compare among three techniques

4.2 Discussions

Web usage mining is the application of data mining techniques to usages logs of large Web data repositories in order to produce results that can be used in the design tasks. In this experiment, the three algorithms are not successful in correct predicting the next request to be generated. This is because; in these models do not look far into the past to correctly discriminate the difference modes of the generative process.

5. Conclusions and future work

Today a large percentage of the Internet population accesses the World Wide Web. Users spend a lot of time impatiently waiting for the web pages to come up on screen. Web caching and Web prefetching are two important techniques used to reduce the noticeable response time perceived by users. Web prefetching technique utilizes the spatial locality of Web objects. In client's browser caching, Web objects are cached in the client's local disk. If the user accesses the same object more than one in a short time, the browser can fetch the object directly from the local disk, eliminating the repeated network latency.

Web servers keep track of web users' browsing behavior in web logs. Because of the increasing of users, World Wide Web is now suffering from the heavy traffic and the long latency. One way to reduce web traffic and speed up web accesses is through the use of prefetching. From log file, one can build statistical models that predict the users' next requests based on their current behavior. In this paper we studied different algorithm for web request prediction. We used proxy log that includes several kinds of client and contains a lot of client request from academic web servers. Our analysis based on three algorithm and using real web logs as training and testing data. Our conclusion is that the first-order Markov model is the best prediction among them. The Markov model construct of such an access model identifies the most likely hyperlink Web pages from the currently requested Web page. The first-order Markov model predicts the next action by only looking at the last action performed by the user.

In the future, we plan to use another algorithm and use the different method to extract sequence rules. However, each approach to prefetching assumes different environment, we must compare several approaches in each environment.

Acknowledgements

This work was supported in part by grants from Suan Dusit Rajabhat University, Thailand. http://www.dusit.ac.th

References

Agrawal, R. & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceeding of the 20th VLDB Conference ages*. Santiago. Chile, 487–499.

Bestavros, A. (1996). Speculative Data dissemination and service to reduce server load. Network Traffic and Service Time. *Proceeding IEEE Conference Data Eng (IEEE ICDE'96)*, February, 151-160.

Cadez, I., Heckerman, D., Meek, C., Smyth, P., & White, S. (2000). Visualization of navigation patterns on a web site using model based clustering. Technical Report MSR-TR-00-18. Microsoft Research.

Catledge, L. & Pitkow, J.E. (1995). *Characterizing browsing behaviors on the World Wide Web*. Computer networks and ISDN Systems, 27(6).

Chakoula O. (2000). *Predicting users' next access to the web server*. Master thesis. Department of Computer Science, University of British Columbia.

Cooley, R., Tan, P-N., & Srivastava, J. (2000). Discovery of interesting usage patterns from Web data. Lecture Notes in Computer Science, 1836, 163-182.

Deshpande, M. & Karypis, G. (2001). Selective Markov models for predicting web page accesses. *In Workshop on Web Mining at the First SIAM International Conference on Data Mining*.

Deshpande, M. (2004). *Prediction/Classification technique for sequence and graphs*. Doctoral dissertation, University of Minnesota.

Etzioni, O. (1996). *The World Wide Web: Quagmire or Gold Mine*. Communications of the ACM, 39(11), 65-68.

Fu, Y., Sandhu, K., and Shih, M.Y. (1999). Clustering of Web users based on access patterns. *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. San Diego.

Haruechaiyasak, C. (2003). *A Data mining and semantic web framework for building a web-based recommender system*". Doctoral dissertation, University of Miami.

Lan, B., Bressan, S., Ooi, B.C., & Tay, Y. (1999). Making web servers pushier. *Proceeding Workshop Web Usage Analysis and User Profiling (WEBKDD '99)*.

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating classification and association mining. *Proceeding of the Fourth International Conference on Knowledge Discovery and Data Mining*, 80-86.

Madria, S.K., Bhowmick, S.S., Ng, W.K. & Lim, E. (1999). Research Issues in web data mining. *Proceeding First International Conference on Data Warehousing and Knowledge Discovery*. Italy, Florence, 303-312.

Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002). Using sequential and non-sequential patterns in predictive web usage mining tasks. *Second IEEE International Conference on Data Mining (ICDM'02)*, 669-672.

Nanolopoulos A., Katsaros, D., & Manolopoulos, Y. (2003). A Data mining algorithm for generalized web prefetching. *IEEE Transactions on Knowledge and Data Engineering*, 5(5), September-October.

Padmanabhan, N. & Mogul, J.C. (1996). *Using predictive prefetching to improve World Wide Web latency*. *ACM SIGCOMM Computer Communication Review*, 26(3), 22-36.

Pallis, G., Angelis, L., & Vakali, A. (2005). *Model-based cluster analysis for web users sessions*. *Foundations of Intelligent Systems, 15th International Symposium (ISMIS 2005)*, Saratoga Springs, NY, USA, May 25-28.

Papoulis, A. (1991). *Probability, Random Variables, and Stochastic Processes*, NY: McGraw Hill.

Pitkow, J. and Pirolli, P. (1999). Mining longest repeating subsequences to predict World Wide Web surfing. *In Second USENIX Symposium on Internet Technologies and Systems*, 139-150.

Sarukkai, R.R. (2000). Link prediction and path analysis using Markov chain. *Proceeding of the 9th international World Wide Web conference on Computer networks: The international journal of computer and telecommunications networking*. Amsterdam. Netherlands, 377-386.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P. (2000). Web usage mining: discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2), 12-23.

Su, Z. Yang, Q., Lu, Y., & Zhang H. (2000). What next: A prediction system for web requests using *N*-gram sequence models. *Proceeding of the First International Conference on Web Information Systems and Engineering Conference*, Hong Kong, pp.200–207.

Wang, K., Zhou, S.Q., & He, Y. (2000). Growing decision trees on association rules. *Proceedings of the International Conference of Knowledge Discovery in Databases*, 265-269.

Wong, C., Shiu, S., & Pal, S. (2001). Mining fuzzy association rules for web access case adaptation. *Proceeding of the workshop Programme at the fourth International Conference on Case-Based Reasoning*, Harbor Center in Vancouver, British Columbia, Canada.

Yang, Q., Li, T., &Wang, K. (2004). Building association-rule based sequential classifiers for Web-Document Prediction. *Journal of Data Mining and Knowledge Discovery*, 8, 253-273.

Zhu, J., Hong, J., & Hughes, J.G. (2002). Using Markov chains for link prediction in adaptive web site". *Proceeding of Soft-Ware 2002: First International Conference on Computing in an Imperfect World.* Lecture Notes in Computer Science, Springer, Belfast, April, 60-73.