International Journal of Engineering Research and Science & Technology

ISSN : 2319-5991
Vol. 2, No. 4
November 2013

www.ijerst.com

Email: editorijerst@gmail.com or editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 2, No. 4, November 2013

Research Paper

WEB USER PREDICTION BY: INTEGRATING MARKOV MODEL WITH DIFFERENT FEATURES

Sonal Vishwakarma[1]*, Shrikant Lade[1], Manish Kumar Suman[2] and Deepak Patel[1]

*Corresponding Author: ***Sonal Vishwakarma*** ✉ *vish.sonal@gmail.com*

Web page prefetching has been widely used to reduce the access latency problem of the Internet. However, if most prefetched web pages are not visited by the users in their subsequent accesses, the limited network bandwidth and server resources will not be used efficiently and may worsen the access delay problem. Therefore, it is critical that we have an accurate prediction method during prefetching. The technique like Markov models have been widely used to represent and analyze user's navigational behavior (usage data) in the Web graph, using the transitional probabilities between web pages, as recorded in the web logs. The recorded users' navigation is used to extract popular web paths and predict current users' next steps. In this paper, we analyze and study Markov model and all-Kth Markov model with webpage keywords as a feature to give more accurate results in Web prediction. Our experiments show the effectiveness of our modified Markov model in reducing the number of paths by propoer clustering without compromising accuracy.

Keywords: Information Extraction, Text Analysis, Ontology, Feature extraction, Text categorization, Clustering

INTRODUCTION

The web is an important source of information retrieval now-a days, and the users accessing the web are from different backgrounds. The usage information about users are recorded in web logs. Analyzing web log files to extract useful patterns is called web usage mining. Web usage mining approaches include clustering, association rule mining, sequential pattern mining, etc. To facilitate web page access by users, web recommendation model is needed. So the Interest in the analysis of user behavior on the Web has been increasing rapidly. This increase stems from the realization that added value for Web site visitors is not gained merely through larger quantities of data on a site, but through easier access to the required information at the right time and in the most suitable form. Estimates of

[1] Information and Technology Branch of RKDF Institute of Science and Technology, Bhopal.
[2] Computer Science and Engineering Branch of Millennium Institute of Technology, Bhopal.

Web usage expect the number of users to climb up to 945 million by 2004 (http://www.c-i-a.com). The majority of these users are non-expert and it difficult to keep up with the rapid development of computer technologies, while at the same time they recognize that the Web is an invaluable source of information for their everyday life. The increasing usage of the Web also accelerates the pace at which information becomes available online. In various surveys of the Web, e .g. (Chakrabarti, 2000), it is estimated that roughly one million new pages are added every day and over 600 GB of pages change per month. A new Web server providing Web pages is emerging every two hours. Nowadays, more than three billion Web pages are available online almost one page for every two people on the earth. In the above, one notices the emergence of a spiral elect, i.e., increasing number of users causing an increase in the quantity of online information, attracting even more users, and so on. This pattern is responsible for the 'explosion' of the Web, which causes the frustrating phenomenon known as 'information overload ' to Web users.

Web usage mining is valuable in many applications like online marketing, E- businesses, etc. The use of this type of web mining helps to gather the important information from customers visiting the site. This enables an in-depth log to complete analysis of a company's productivity flow. E-businesses depend on this information to direct the company to the most effective Web server for promotion of their product or service.

With the growing popularity of the World Wide Web. A large number of users access web sites in all over the world. When user access a websites, a large volumes of data such as addresses of users or URLs requested are gathered automatically by Web servers and collected in access log which is very important because many times user repeatedly access the same type of web pages and the record is maintained in log files. These series of accessed web pages can be considered as a web access pattern which is helpful to find out the user behavior. Through this behavior information, we can find out the accurate user next request prediction that can reduce the browsing time of web page thus save the time of the user and decrease the server load. In recent years, there has been a lot of research work done in the field of web usage mining, Future request prediction. The main motivation of this study is to know what research has been done on Web usage mining in future request prediction.

In Web prediction, main challenges are in both preprocessing and prediction. Preprocessing challenges include handling large amount of data that cannot fit in the computer memory, choosing optimum sliding window size, identifying sessions, and seeking/extracting domain knowledge. Prediction challenges include long training/prediction time, low prediction accuracy, and memory limitation.

RELATED WORK

Agrawal R and Srikant R (1994) proposed a method for constructing first-order and second-order Markov models of Web site access prediction based on past visitor behavior and compare it association rules technique. In these approaches, sequences of user requests are collected by the session identification technique, which distinguishes the requests for the same web page in different browses.

Ontology patterns were introduced by Blomqvist and Sandkuhl in 2005 (Pitkow and Pirolli, 1999). Later the same year, Gangemi

(2005) presented his work on ontology design patterns. Such patterns, encodings of best practices, were intended to reduce the need of extensive experience when developing ontologies.

In the TLPM, Chu -Hui Lee *et al*., (2011), in level one, Markov model is used to predict the next possible category which will be browsed by the user . In level two, Bayesian theorem is used to predict the next possible page which belongs to the predicted category of level one to archive the goal of reducing prediction scope more efficiently through the two-level framework. The experiment result proves that TLPM can archive the purpose and improve the efficiency of prediction by the way of finding out the important category in level one and decreasing the candidate page set in level two. Joachims *et al.* propose the WebWatcher which is a path-based recommender model based on *k* NN and reinforcement learning. The combination of previous tours of similar users and reinforcement learning is used in recommendations.

Nasraoui (Nasraoui O and Krishnapuram R 2002; Nasraoui O and Petenes C) propose a Web recommendation system using fuzzy inferences. Clustering is applied to group profiles using hierarchical unsupervised niche clustering. Context-sensitive URL associations are inferred using a fuzzy approximate reasoning-based engine.

Researchers have used various prediction models including k-nearest neighbor (kNN), ANNs (Nasraoui.O and Krishnapuram.R 2002),fuzzy inference (Nasraoui.O and Petenes.C 2003) SVMs, Bayesian model, Markov model and others. Mobasher et al. use the ARM technique in WPP and propose the frequent item set graph to match an active user session with frequent item sets and predict the next page that user is likely to visit. However, ARM suffers from well known limitations including scalability and efficiency.

Trilok Nath Pandey, Ranjita Kumari Dash, Alaka Nanda Tripathy, Barnali Sahu Pitkow J and Pirolli P (1999) proposed Integrating Markov Model with Clustering (IMC) approach for user future request prediction. In this paper author presented the improvement of markov model accuracy by grouping web sessions into clusters. The web pages in the user sessions are first allocated into categories according to web services that are functionally meaning full. Then k-means clustering algorithm is implemented using the most appropriate number of clusters and distance measure.

Sujatha and Punithavalli (2012) proposed the Prediction of User navigation patterns using Clustering and Classification (PUCC) from web log data. In the first stage PUCC focuses on separating the potential users in web log data, and in the second t-stage clustering process is used to group the potential users with similar interest and in the third stage the results of classification and clustering is used to predict the user future requests.

BACKGROUND

Markov Models for Predicting User's Actions

As discussed in the introduction, techniques derived from Markov models (Deshpande M and Karypis G (2004) have been extensively used for 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 |A| X |S| *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.* 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. A somewhat more complicated model computes the predictions by looking at the last two actions performed by the user. This is called the *second-order Markov* model, and its states correspond to all possible pairs of actions that can be performed in sequence.

This approach is generalized to the *Kth-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. For example, suppose the prediction of next page accessed by a user on a website is a problem. The input data for building Markov models consists of *web-sessions*, where each session consists of the sequence of the pages accessed by the user during his/her visit to the site. In this problem, the actions for the Markov model correspond to the different pages in the web site, and the states correspond to all consecutive pages of length *K* that were observed in the different sessions. In the case of first-order models, the states will correspond to single pages, in the case of second-order models, the states will correspond to all pairs of consecutive pages, and so on. Once the states of the Markov model have been identified, the transition probability matrix can then be computed. There are many ways in which the TPM can be built.

The most commonly used approach is to use a *training* set of action-sequences and estimate each T*ji* entry based on the frequency of the event that action *ai* follows the state *sj.* For example consider the *web-session WS*2 (*P*3; *P*5; *P*2; *P*1; *P*4) shown in Figure 2. If they are using *first-order Markov model* then each state is made up of a single page, so the first page *P*3 corresponds to the state *s*3. Since page *p*5 follows the state *s*3 the entry *t*35 in the TPM will be updated. Similarly, the next state will be *s*5 and the entry *t*52 will be updated in the TPM. In the case of higher-order model each state will be made up of more than one actions, so for a second-order model the first state for the *web-session WS*2 consists of pages {*P*3; *P*5} and since the page *P*2 follows the state {*P*3; *P*5} in the web session the TPM entry corresponding to the state {*P*3; *P*5} and page *P*2 will be updated. Once the transition probability matrix is built making prediction for web sessions is straight forward. For example, consider a user that has accessed pages {*P*1; *P*5; *P*4}. If they want to predict the page that will be accessed by the user next, using a first-order model, we will first identify the state *s*4 that is associated with page *P*4 and look up the TPM to find the page *pi* that has the highest probability and predict it. In the case of our example the prediction would be page *P*5.

Web Sessions

*WS*1: {*P*3; *P*2; *P*1}

*WS*2: {*P*3; *P*5; *P*2; *P*1; *P*4}

*WS*3: {*P*4; *P*5; *P*2; *P*1; *P*5; *P*4}

*WS*4: {*P*3; *P*4; *P*5; *P*2; *P*1}

*WS*5: {*P*1; *P*4; *P*2; *P*5; *P*4}

Figure 1: Sample Web Sessions with the Corresponding 1st and 2nd Order Transition Probability Matrices

1st Order	P1	P2	P3	P4	P5
S1={P1}	0	0	0	2	1
S2={P2}	4	0	0	0	1
S3={P3}	0	1	0	1	1
S4={P4}	0	1	0	0	2
S5={P5}	0	3	0	2	0

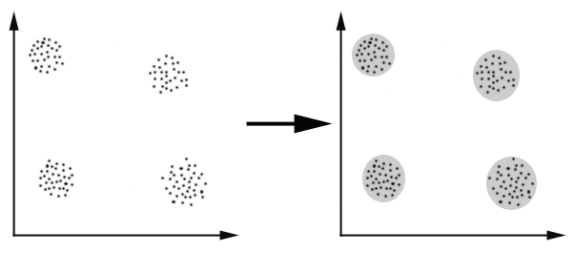
2nd Order	P1	P2	P3	P4	P5
{P1;P4}	0	1	0	0	0
{P1;P5}	0	0	0	1	0
{P2;P1}	0	0	0	1	1
{P2;P5}	0	0	0	1	0
{P3;P2}	1	0	0	0	0

2nd Order	P1	P2	P3	P4	P5
{P2;P5]	0	1	0	0	0
{P2;P4}	0	0	0	0	1
{P4;P5}	0	2	0	0	0
{P5;P2}	3	0	0	0	0
{P3;P4}	0	0	0	0	1

CLUSTERING

Clustering (Nasraoui O and Petenes C, 2003) is the most important unsupervised learning problem. So the simple definition of Clustering could be "the process of organizing objects into groups whose members are similar in some way". A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. We can show this with a simple example:

Figure 2: Representing Cluster



Feature Selection

As web pages have contents so each page have the identity as the number of specific keywords and user try to move as per content as well in some website, that may be use as related content. For example if user read some paragraph in the webpage then the next page it visit may be related to the content of the current page keywords. So Keywords of the page act as the feature of the webpage. Other feature of the webpage is the user session as the pattern of reaching any page. These features are use for prediction of next webpage as in terms of the keywords that the user find in preceding pages, it may look for some of the relative pages as well. So collection of keywords is done by reading the content of the webpages in Bag of Words (BOW), then preprocess the data by removing the stopwords. One more filter is require as the BOW may have large number of unnecessary words which get remove by setting the threshold for the frequency of word appear in the web page. Remove all words which are beyond the range of the threshold. So by this filter BOW finally contain keywords set.

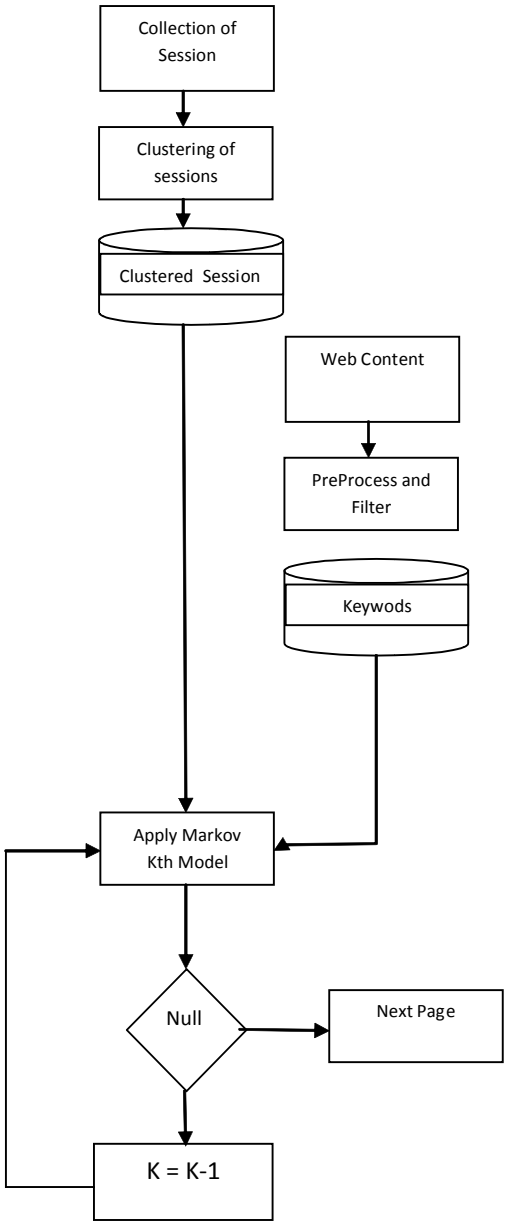
THE PROPOSED APPROACH

In this paper the different module for web page prediction is mention below.

Module 1

Web user session are collected in the form of session vector from the website whose web page prediction need to be done. As the web session are in the order when they are seen by the user or the timestamp order. So some clustering of the session are done which is Hierarchial approach as the sequence of web pages are use. In order to support markov model one cluster is done as the same first page of the session, then

Figure 3: Modules of the Web Page Prediction



second is create as the same first two page of the session, then third cluster is create as the same first three page of the session, in the similar fashion fourth cluster is also prepared. These clustering will reduce the data searching task, as the number of session is divide in each cluster so overall comparing time is reduce and the work will be more fine as the cluster continue more similar session for the prediction.

Module 2

For improving the web prediction probability one more feature is introduced, which is the collection of keywords of each page web content that will support the prediction of next page with the web user contents.

First collect the web page content then pre-process as, Data preprocessing reduces the size of the input text documents significantly. It involves activities like sentence boundary determination, natural language specific stop word elimination and stemming. Stop-words are functional words which occur frequently in the language of the text (for example, a, the, an, of etc., in English language), so that they are not useful for classification. Here we read whole project and put all words in the vector. Now again read the file which contain stop words then remove similar words from the vector. Once the data is pre-process then it will be the collection of the words that may be in the ontology list. For example let one paper of the image class is taken and its text vector is {a1, f1, s1, a2, s2, a3, a4, f2 ... an} and let the stop words collection is {a1, a2, a3, ... am}. Then the vector obtain after the Pre-Processing is {f1, s1, s2, f2, ... fx}.

Then pass that vector from the filter to refine the content of the vector by setting a upper limit and lower limit of the frequency of the words which appear in the content. So in this way that vector have the collection of the keywords pagewise. The user session which is need to predict is use for collecting a user interest words which is the collection of words that are obtain from the previous pages. Then the pages which are predicted by markov model are filtered by the page which have most of the words of user interest.

So in this way this feature support the web page prediction.

Module 3

In this mode as all the steps of pre-processing then grouping is already done in previous modules so it simply work. All-K^{th} Markov model (Agarwal *et al.,* 1993; V Sujatha and Punithavalli, 2012), we generate all orders of Markov models and utilize them collectively in prediction. Note that the function *predict* (*x,mk*) is assumed to predict the next page visited of session *x* using the k^{th}-order Markov model *mk*. If the *mk* fails, the *mk*–1 is considered using a new session *x'* of length *k*–1 where *x'* is computed by stripping the first page ID in *x*. This process repeats until prediction is obtained or prediction fails. For example, given a user session *x* = <*P*1*, P*5*, P*6>, prediction of all-K^{th} model is performed by consulting third-order Markov model. If the prediction using third-order Markov model fails, then the second-order Markov model is consulted on the session *x_* = *x*–*P*1 = <*P*5*, P*6>. This process repeats until reaching the first-order Markov model. Therefore, unlike the basic Markov model, the all-K^{th}-order Markov model achieves better prediction Deshpande M and Karypis G (2004) and it only fails when all orders of the basic Markov models fail to predict.

Algorithm for web page prediction:

1. Collect User session x of length K
2. While k!=0
3. Pi←Predict_markov(x , K)
4. If Pi != 0 return Pi
5. K ← K–1
6. If Goto step 2

Algorithm for Predict_markov

1. Input x user session and modal number k
2. V←search(x) [Search next frequent page by Markov modal]
3. If V has more then one page
4. Key_vector←Keywords() [Collection of website keywords]
5. V←similar(Key_vector, V) [Select page have most similar keywords as previous page have]
6. Ifend
7. Return V

Predict_markov algorithm take session and modal number as input then find most frequent page. If it generate more then one page then, second feature will be predicted for the page selection which is keywords extracted from the web pages. There similar function take key_vector which is the collection of the keywords which is obtain from the previous page of the session, then compare the keywords of the pages in V vector. The most similar page will be the next target page of the session. This page is return to the function.

EXPERIMENT AND RESULTS

In order to predict new web user session page the data sets and preprocessing we considered. We considered two data sets, namely, the NASA data set, the one generate artificial by the website create for this work. In addition to many other items, the preprocessing of a data set includes the following: grouping of sessions, identifying the beginning and the end of each session, assigning a unique session ID for each session, and filtering irrelevant records detail of the dataset is mention in Table 1. In this experiments, the cleaning steps and the session identification techniques is done in two modules of clustering and keyword feature vector creation.

Table 1: Dataset Summary		
	NASA	**Artificial**
Total Session	50,000	20,000
Average Session Length	6.4	4.5
Number of Pages	2266	16
Data Set Time	Aug/1995	Self

Experimental Setup

Implementation of both the Markov and the keywords feature vector for web page prediction models is done as the web sessions are clustered in different groups, then collect keywords from the web content to make keywords feature vector. Once these step is done then train the model with different ratio of the Dataset for example 60% for training and 40% for the testing, or 67% for training and 33% for the testing of the modal.

To measure the accuracy, the generalization accuracy procedure by partitioning each data set randomly into a training set (two-thirds of the original set) and a testing set (one-third of the original set). The generalization accuracy is a standard procedure which is widely used to measure prediction models' accuracy against new examples that might not have been observed during training. We have run each experiment 10 times, applying random partitioning/sampling on the data set, for each different criterion. In other words, each point in any of the presented curves is the average value of 10 runs of the same parameters yet different partitioning/sampling.

Prediction Setup

Given a testing session (t) of length L, we conduct prediction using the (L – 1)-gram Markov model and obtain the prediction to evaluate the accuracy of the model. Recall that the last page of t is the final outcome that we will evaluate the correctness of the mode against; hence, we use (L " 1)- gram. In case t is longer than the highest N-gram used in the experiment, we apply a sliding window of size L on t. For example, suppose $t = p1, p2, p3, p4, p5$, if we use the third order Markov model, then we break t into $p1, p2, p3, p4$ and $p2, p3, p4, p5$.

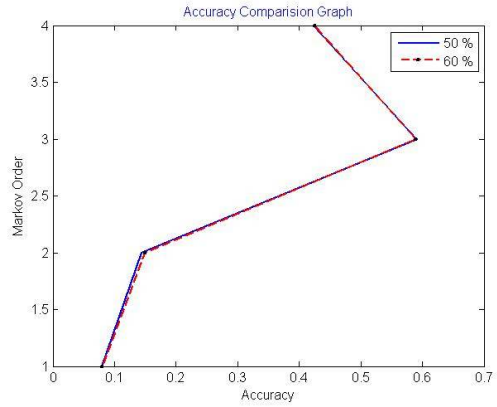
Here accuracy is the measuring parameter which is taken as on both the data set with different training set. In case of Nasa Dataset, the results are obtained without content features as the dataset contain some of the logs the are image, so retrieving content of that page is not possible because of this reason, Artificial dataset is introduced and used for both the case of the web features.

From Table 2, include of both the feature results are more accurate then using the log session feature alone as the accuracy. Since data sets are different so the accuracy results have such a difference but method for both the data set for log feature is same.

Table 2: All K-order Values for the 60% Dataset Size			
	Artificial DataSet		**NASA DataSet**
Markov Orrder	**Logs**	**Logs + Content**	**Logs**
First	0.062	0.085	0.91
Second	0.1213	0.132	0.088
Third	0.564	0.543	0.014
Fourth	0.4102	0.208	0.138
Accuracy	0.1953	0.2050	0.086

For different training set it has been observed as the training data increase the accuracy also increase but by very small fraction. As after 50% of data set size accuracy of change by small value in different morkov order values which is shown in Figure 4.

Figure 4: Accuracy Graph for Different Size of Dataset 50% and 60%



CONCLUSION

World Wide Web has necessitated the users to make use of automated tools to locate desired information resources and to follow and asses their usage pattern. Web page prefetching has been widely used to reduce the user access latency problem of the internet; its success mainly relies on the accuracy of web page prediction. Markov model is the most commonly used prediction model because of its high accuracy. Low order Markov models have higher accuracy and lower coverage. The higher order models have a number of limitations associated with (i) Higher state complexity, (ii) Reduced coverage, (iii) Sometimes even worse prediction accuracy. Clustering is one of the best solutions for resolving the problem of worse prediction accuracy of Markov model. It is a powerful method for arranging users' session into clusters according to their similarity. This work develop the techniques to overcome the issues of web page prediction. However, research of the web page prediction is just at its beginning and much deeper understanding needs to be gained.

REFERENCES

1. Adami G, Avesani P and Sona D (2003), "Clustering documents in a web directory", WIDM'03, USA, pp. 66-73.
2. Agrawal R, Imielinski T and Swami A (1993), "Mining association rules between sets of items in large databases", *ACM SIGMOD Conference on Management of data,* pp. 207-216.
3. Agrawal R and Srikant R (1994), "Fast algorithms for mining association rules", VLDB'94, Chile pp. 487-499.
4. Chu -Hui Lee, Yu-lung Lo and Yu-Hsiang Fu (2011), "A novel rediction model based on hierarchical characteristic of web site", *Expert Systems with Applications,* Vol. 38.
5. Sujatha Punithavalli V (2012), "Improved User Navigation Pattern Prediction Technique From Web Log Data", *Procedia Engineering,* Vol. 30.
6. Blomqvist E and Sandkuhl K (2005), "Patterns in ontology engineering: Classication of ontology patterns", In: Proceedings of the 7th International Conference on Enterprise Information Systems, pp. 413-416
7. Gangemi A (2005), "Ontology design patterns for semantic web content", In: The Semantic WebISWC, Springer, pp. 262-276.
8. Nasraoui O and Petenes C (2003), "Combining Web usage mining and fuzzy inference for Website personalization," in Proc. WebKDD, pp. 37-46.
9. Nasraoui O and Krishnapuram R (2002), "One step evolutionary mining of context

sensitive associations and Web navigation patterns," in Proc. SIAM Int. Conf. Data Mining, Arlington, VA, April, p. 531.

10. Deshpande M and Karypis G (2004), "Selective Markov Models for Predicting Web-Page Accesses", *ACM Transactions on Internet Technology* (TOIT), Vol. 4, No. 2, pp. 163-184.
11. Pitkow J and Pirolli P (1999), "Mining longest repeating subsequences to predict www surfing", USENIX Annual Technical Conference, pp. 139-150.
12. Computer Industry Almanac, http://www.c-i-a.com
13. Chakrabarti S (2000), "Data mining for hypertext: A tutorial survey", ACM SIGKDD Explorations, Vol. 1, No. 2, pp. 1-11.

International Journal of Engineering Research and Science & Technology
Hyderabad, INDIA. Ph: +91-09441351700, 09059645577
E-mail: editorijlerst@gmail.com or editor@ijerst.com
Website: www.ijerst.com