Integrating Markov Model with Clustering for Predicting Web Page Accesses

Faten Khalil, Doctoral Student, Department of Mathematics & Computing, University of Southern Queensland [HREF1], Toowoomba, Australia. khalil@usq.edu.au

Jiuyong Li, Associate Professor, School of Computer and Information Science, University of South Australia [HREF2], Mawson Lakes, Australia. Jiuyong.Li@unisa.edu.au

Hua Wang [HREF3], Senior Lecturer, Department of Mathematics and Computing, University of Southern Queensland [HREF1], Toowoomba, Australia.wang@usq.edu.au

Abstract

Predicting the next page to be accessed by Web users has attracted a large amount of research work lately due to the positive impact of such prediction on different areas of Web based applications. Major techniques applied for this intention are Markov model and clustering. Low order Markov models are coupled with low accuracy, whereas high order Markov models are associated with high state space complexity. On the other hand, clustering methods are unsupervised methods, and normally are not used for classification directly. This paper involves incorporating clustering with low order Markov model techniques. The pre-processed data is divided into meaningful clusters then the clusters are used as training data while performing 2nd order Markov model techniques. Different distance measures of k-means clustering algorithm are examined in order to find an optimal one. Experiments reveal that incorporating clustering of Web documents according to Web services with low order Markov model improves the web page prediction accuracy.

Introduction

The ongoing increase of digital data on the Web has resulted in the overwhelming amount of research in the area of Web user browsing personalization and next page access prediction. It is rather a complicated issue since, until now, there is not a single theory or approach that can handle the increasing number of data with improved performance, efficiency and accuracy of Web page prediction [HREF4]. Two of the most common approaches used for Web user browsing pattern prediction are Markov model and clustering. Each of these approaches has its own shortcomings. Markov model is the most commonly used prediction model because of its high accuracy. Low order Markov models have higher accuracy and lower coverage than clustering. In order to overcome low coverage, all-kth order Markov models have been used [HREF5] where the highest order is first applied to predict a next page. If it cannot predict the page, it decreases the order by one until prediction is successful. This can increase the coverage, but it is associated with higher state space complexity. Clustering methods are unsupervised methods, and normally are not used for classification directly. However, proper clustering groups users' sessions with similar browsing history together, and this facilitates classification. Prediction is performed on the cluster sets rather than the actual sessions. Clustering accuracy is based on the selected features for partitioning. For instance, partitioning based on semantic relationships or contents [HREF6] or link structure [HREF7] usually provides higher accuracy than partitioning based on bit vector, spent time, or frequency. However, even the semantic, contents and link structure accuracy is limited due to the unidirectional nature of the clusters and the multidirectional structure of Web pages. This paper involves implementing a clustering algorithm to partition Web sessions into clusters and then applying Markov model techniques based on the clusters in order to achieve better accuracy and performance of next page access prediction. Section 2 looks at previous literature in the area of combininig clustering with Markov model techniques. Section 3 explains the process acquired to achieve better prediction. In section 4, we prove our new process experimentally and section 5 concludes our work.

Literature Review

Markov model and clustering are two frameworks used for predicting the next page to be accessed by the Web user. Many research papers addressed Web page prediction by using clustering, Markov model or a combination of both techniques. Kim et al. [HREF4] combine most prediction models (Markov model, sequential association rules, association rules and clustering) in order to improve the prediction recall. The proposed model proves to outperform classical Web usage mining techniques. However, the new model depends on many factors, like the existence of a Web site link structure and the support and confidence thresholds. These factors affect the order of the applied models and the performance of the new model. Cadez et al. [HREF8] on the other hand, combined first order Markov model with clustering using a different approach. They partitioned site users using a model-based clustering approach where they implemented first order Markov model using the Expectation-Maximization algorithm. After partitioning the users into clusters, they displayed the paths for users within each cluster. Our work is distance based and not model based and we used Markov model for prediction rather than clustering. Another paper that combines both Markov model and clustering techniques for Web page link prediction is [HREF7], where the authors construct Markov models from log files and they use co-citation and coupling similarities for measuring the conceptual relationships between Web pages. CitationCluster algorithm is then proposed to cluster conceptually related pages. A hierarchy of the Web site is constructed from the clustering results. The authors then combine Markov model based link prediction to the conceptual hierarchy into a prototype called ONE to assist users' navigation. The authors implement a hierarchical clustering technique that could lead to running time complexity with large Web log files. Although Web page prediction performance was improved by previous work, none of the papers showed an improvement in the Web page prediction accuracy. Kim et. al used a combination of models but their work improved recall but did not improve the Web page prediction accuracy [HREF4]. Our work proves to outperform previous work in terms of Web page prediction accuracy using a combination of clustering and Markov model techniques. We implement a simple clustering algorithm, k-means algorithm where using different distance measures can lead to different results. Distance measures were analyzed and an optimal one was chosen.

Methodology

Web page prediction involves anticipating the next page to be accessed by the user or the link the Web user will click at next when browsing a Web site. For example, what is the chance that a Web user visiting a site that sells computers will buy an extra battery when buying a laptop? Or, may be there is a greater chance the user will buy an external floppy drive instead. Users' past browsing experience is very fundamental in extracting such information. This is when modeling techniques come at hand. For instance, using clustering algorithms, we are able to personalize users according to their browsing experience. Different users with different browsing behavior are grouped together and then prediction is performed based on the users' link path in the appropriate cluster. Similar kind of prediction can be in effect using Markov models conditional probability. For instance, if 50% of the users access page D after accessing pages ABC, then there is a 50/50 chance that a new user that accesses pages ABC will access page D next. Our work improves the Web page access prediction accuracy by combining both Markov model and clustering techniques. It is based on dividing Web sessions into groups according to Web services and performing Markov model analysis using clusters of sessions instead of the whole data set. This process involves the following steps:

1. Preprocess the Web server log files in a manner where similar Web sessions are allocated to appropriate categories.
2. Analyze and calculate different distance measures and determine the most suitable distance measure.
3. Decide on the number of clusters (k) and partition the Web sessions into clusters according to the chosen distance measure.
4. For each cluster, return the data to its uncategorized and expanded state.
5. Perform Markov model analysis on the whole data set.
6. For each item in the test data set, find the approprite cluster the item belongs to.
7. Calculate 2-Markov model accuracy using the cluster data as the training data set.
8. Calculate the total prediction accuracy based on clusters.
9. Compare the Markov model accuracy of the clusters to that of the whole data set.

Feature Selection

Because of the overwhelming amount of Web data, it is very important to group data according to some features before applying clustering techniques. This will reduce the state space and will make the clustering task simpler. If the features are not selected appropriately, there is no way we can get good clusters no matter what type of clustering algorithm is used. Wang et al. [HREF9] presented different feature selections and metrics that form the base of E-commerce customer groupings for clustering purposes. They examined features like services request, navigation pattern and resource usage. The result of their experimentations proved that all features yield similar results and thus, grouping customers according to one of the features selected should do the job. For our purposes, we will group the pages, and not users, according to services requested since it is applicable to our log data and is simple to implement. Grouping pages according to services requested yields best results if it is carried out according to functionality [HREF9]. The grouping of Web pages according to functionality could be done either by removing the suffix of visited pages or the prefix. In our case, we cannot merge according to suffix because, for example, pages with suffix index.html could mean any default page like OWOW/sec4/index.html or OWOW/sec9/index.html or ozone/index.html. Therefore, merging will be according to a prefix. Since not all Web sites have a specific structure where we can go up the hierarchy to a suitable level, we had to come up with a suitable automatic method that can merge similar pages automatically. For our log file data, the chosen prefix will be delimited by slash, dot or space. A program runs and examines each record. It only keeps the delimited and unique word. A manual examination of the results also takes place to further reduce the number of categories by combining similar pages.

Clustering

According to Srivastava et al. [HREF10], clustering is a pattern discovery algorithm in the Web usage mining stage of Web mining. It is defined as the classification of patterns into groups (clusters) based on similarity in order to improve common Web activities. Clustering can be model-based or distance-based. With model-based clustering [HREF11], the model type is often specified a priori and the model structure can be determined by model selection techniques and parameters estimated using maximum likelihood algorithms, e.g., the Expectation Maximization (EM). Distance-based clustering involves determining a distance measure between pairs of data objects, and then grouping similar objects together into clusters. The most popular distance-based clustering techniques include partitional clustering and hierarchical clustering. A partitional method partitions the data objects into K groups and is represented by k-means algorithm. A hierarchical method builds a hierarchical set of nested clusters, with the clustering at the top level containing a single cluster of all data objects and the clustering at the bottom level containing one cluster for each data object. Model-based clustering has been shown to be effective for high dimensional text clustering [HREF11]. However, hierarchical distance-based clustering proved to be unsuitable for the vast amount of Web data. Although distance-based clustering methods are computationally more complex than model-based clustering approaches, they have displayed their ability to produce more efficient Web documents clustering results [HREF12, HREF13]. Clustering can also be supervised or unsupervised. The difference between supervised and unsupervised clustering is that with supervised clustering, patterns in the training data are labeled. New patterns will be labeled and classified into existing labeled groups [HREF14]. Unsupervised clustering can be classified as hierarchical or non-hierarchical [HREF15]. A common method of non-hierarchical clustering is the k-means algorithm that tends to cluster data into even populations. In this paper, we use a straightforward implementation of the k-means clustering algorithm. It is distance-based, unsupervised and partitional.

Distance Measures

The clustering algorithm chosen for this work is K-means clustering algorithm that is a simple and popular form of cluster analysis. It has been widely used in grouping Web user sessions. It is distance based as opposed to complex model based algorithms. It involves the following:

1. Define a set of sessions (n-by-p data matrix) to be clustered.
2. Define a chosen number of clusters (k).
3. Randomly assign a number of sessions to each cluster.

The k-means clustering repeatedly performs the following:

1. Calculate the mean vector for all items in each cluster.
2. Reassign the sessions to the cluster whose center is closest to the session.

Until there is no change for all cluster centers.

Because the first clusters are created randomly, k-means runs different times each time it starts from a different point giving different results. The different clustering solutions are compared using the sum of distances within clusters. The clustering solution with the least sum of distances is considered. Therefore, k-means clustering depends greatly on the number of clusters (k), the number of runs and the distance measure used. The output is a number of clusters with a number of items in each cluster. Distances or similarities between items are a set of rules that serve as a method for grouping or separating items. The distance measured between items in each cluster plays a vital role in forming the clusters. Due to different units of measure in different dimensions, the Euclidean distance measure may not be an adequate measure of closeness even though it is commonly assumed to be. It is important to mention that other non-Euclidean distance measures have been proposed [HREF12] and can be useful for the same purpose. In this paper, we examine five distance measures: Euclidean and Squared Euclidean, City Block, Cosine, Pearson Correlation and Hamming.

Euclidean: This is the most straightforward and the most commonly chosen type of distance. It forms the actual geometric distance in the multidimensional space. It is computed as follows:

$$\text{Euclidean}(x, y) = \sqrt{\sum (x_i - y_i)^2} \qquad (1)$$

If greater weight needs to be assigned on items that are further apart, Squared Euclidean distance is used instead and it is computed as follows:

$$\text{Squared Euclidean}(x, y) = \sum (x_i - y_i)^2 \qquad (2)$$

City Block: Also known as Manhattan distance is another common distance measure and it yields results that are similar to the Euclidean distance results. It is only different in that it lessens the outliers effect. It is simply computed by finding the average difference between dimensions:

$$\text{City Block}(x, y) = \sum |x_i - y_i| \qquad (3)$$

Hamming: For real valued vectors, the Hamming distance is equivalent to the City Block distance. It is commonly used to compare binary vectors because of its simplicity. The Hamming distance measures the number of substitutions required to change one string into the other. It can be performed with an exclusive OR function, XOR. It is defined as follows:

$$\text{Hamming}(x, y) = \sum |x_i - y_i| \qquad (4)$$

The Hamming distance is an unsuitable distance measure for our data set, because data items have to be converted to binary data. This means that the weights we placed on the pages to specify the number of their occurrences will be eliminated.

Cosine: It determines similarity by the cosine of the angle between two vectors [HREF12]. Cosine distance measure is the most popular measure for text documents since the similarity does not depend on the length and it allows documents with the same composition but different totals to be treated identically. The Cosine distance is given by:

$$\text{Cosine}(x, y) = \frac{\sum (x_i y_i)}{\sqrt{\sum (x_i)^2 \sum (y_i)^2}} \qquad (5)$$

Pearson Correlation: It is mostly used in collaborative filtering to predict a feature from a highly similar mentor group of objects whose features are known [HREF12]. It is defined as follows:

$$\text{Correlation}(x, y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2)}\sqrt{\sum (y_i - \bar{y})^2}} \qquad (6)$$

K-means computes centroid clusters differently for different k-means supported distance measures. Therefore, a normalization step was necessary for Cosine and Correlation distance measures for comparison purposes. The points in each cluster, whose mean forms the centroid of the cluster, are normalized to unit Euclidean length. According to Strehl et al. [HREF12] and Halkidi et al. [HREF16], Cosine distance measure which is a direct application of the extended Jaccard coefficient, yields better clustering results than Pearson Correlation and the Euclidean distance measure [HREF12, HREF16]. Because different distance measures have been applied for different purposes, there is no apparent one clustering validation measure we can rely on to test our clusters in terms of their proximity. The importance of the validation measure is significant in order to form the most appropriate clusters to be used in conjunction with Markov model. The most common clustering validation technique is entropy [HREF12, HREF17, HREF9]. Entropy is defined as follows:

$$\Lambda^{(E)}(C_l) = \sum \frac{n_l^{(\lambda)}}{n_l} \log\left(\frac{n_l^{(\lambda)}}{n_l}\right) \qquad (7)$$

Entropy measures the purity of the clusters with respect to the given class labels. For our data sets, entropy is measured by calculating the probability that a page in a cluster x belongs to category n_x. Entropy tends to favor small clusters. If the cluster has all its pages belonging to one category, the entropy will be 0. The entropy measure increases as the categories become more varied. The overall entropy of the whole clustering solution is measured as the weighted sum of entropy measures of all clusters within the clustering solution. Xiong et al. [HREF17], proved through experimentations that the entropy evaluation does not confirm with the k-means true clusters and its results could be misleading. Therefore, we were not able to rely on the entropy results alone to discover the optimal number of clusters (k). In our distance measures evaluations we run entropy evaluation measures, we calculate the mean of the distances and we plot clusters figures on the clusters obtained using different distance measures.

Markov Model

Markov models are becoming very commonly used in the identification of the next page to be accessed by the Web site user based on the sequence of previously accessed pages [HREF18]. Let P = {p1, p2, ..., pm} be a set of pages in a Web site. Let W be a user session including a sequence of pages visited by the user in a visit. Assuming that the user has visited *l* pages, then prob(pi|W) is the probability that the user visits pages pi next. Page p*l*+1 the user will visit next is estimated by:

$$P_{l+1} = \arg\max_{p \in P}\{P(P_{l+1} = p|W)\} = \arg\max_{p \in P}\{P(P_{l+1} = p|pl, pl-1, \ldots, p1)\} \qquad (8)$$

This probability, prob(pi|W), is estimated by using all sequences of all users in history (or training data), denoted by W. Naturally, the longer *l* and the larger W, the more accurate prob(pi|W). However, it is infeasible to have very long *l* and large W and it leads to unnecessary complexity. Therefore, to overcome this problem, a more feasible probability is estimated by assuming that the sequence of the Web pages visited by users follows a Markov process. The Markov process imposed a limit on the number of previously accessed pages k. In other words, the probability of visiting a page pi does not depend on all the pages in the Web session, but only on a small set of k preceding pages, where k << l. The equation becomes:

$$P_{l+1} = \arg\max_{p \in P}\{P(P_{l+1} = p|pl, pl-1, \ldots, pl-(k-1)\} \qquad (9)$$

k denotes the number of the preceding pages and it identifies the order of the Markov model. The resulting model of this equation is called the all kth order Markov model. Of course, the Markov model starts calculating the highest probability of the last page visited because during a Web session, the user can only link the page he is currently visiting to the next one. The example is similar to Desphpande's Figure 1 [HREF18]. Let S_j^k be a state containing k pages, S_j^k = {pl-(k-1),pl-(k-2),...,pl}. The probability of P(pi| S_j^k) is estimated as follows from a history (training) data set.

$$P(pi|S_j^k) = \frac{Frequency\,(\langle S_j^k, pi\rangle)}{Frequency\,(S_j^k)} \qquad (10)$$

This formula calculates the conditional probability as the ratio of the frequency of the sequence occurring in the training set to the frequency of the page occurring directly after the sequence. The fundamental assumption of predictions based on Markov models is that the next state is dependent on the previous k states. The longer the k is, the more accurate the predictions are. However, longer k causes the following two problems: The coverage of model is limited and leaves many states uncovered; and the complexity of the model becomes unmanageable. Therefore, the following are three modified Markov models for predicting Web page access.

1. All kth Markov model: This model is to tackle the problem of low coverage of a high order Markov model. For each test instance, the highest order Markov model that covers the instance is used to predict the instance. For example, if we build an all 4-Markov model including 1-, 2-, 3-, and 4-, for a test instance, we try to use 4-Markov model to make prediction. If the 4-Markov model does not contain the corresponding states, we then use the 3-Markov model, and so forth [HREF5].
2. Frequency pruned Markov model: Though all kth order Markov models result in low coverage, they exacerbate the problem of complexity since the states of all Markov models are added up. Note that many states have low statistically predictive reliability since their occurrence frequencies are very low. The removal of these low frequency states affects the accuracy of a Markov model. However, the number of states of the pruned Markov model will be significantly reduced.
3. Accuracy pruned Markov model: Frequency pruned Markov model does not capture factors that affect the accuracy of states. A high frequent state may not present accurate prediction. When

we use a means to estimate the predictive accuracy of states, states with low predictive accuracy can be eliminated. One way to estimate the predictive accuracy using conditional probability is called confidence pruning. Another way to estimate the predictive accuracy is to count (estimated) errors involved, called error pruning.

In this paper, we employ the frequency pruned Markov model. When choosing the Markov model order, our aim is to determine a Markov model order that leads to high accuracy with low state space complexity. Figure 1 reveals the increase of precision as the all kth order Markov model increases. On the other hand, table 1 shows the increase of the state space complexity as the order of all kth Markov model increases. Based on this information, we use the all 2nd order Markov model because it has better accuracy than that of the all 1st order Markov model without the drawback of the state space complexity of the all 3rd and all 4th order Markov model.

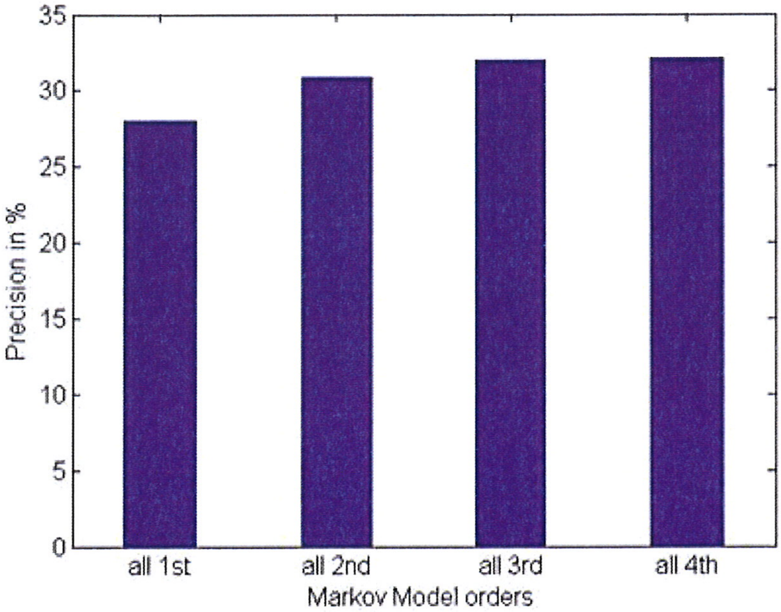


Figure 1: Precision of all 1-, 2-, 3-and 4-Markov model orders.

Table 1: Number of states of Markov model orders.

Model	All-kth States
1st order	745
2nd order	9162
3rd order	14977
4th order	17034

Combining Clustering and Markov Model

The web data is heterogeneous in nature. Each session is a collection of visited Web pages by the user. Every user has a different level of browsing expertise and sessions are formed mainly haphazardly because users usually follow different paths when trying to access the same page. Clustering combines similar Web page paths or user sessions together and subsets of data are therefore more homogeneous resulting in simpler Markov model computations. By applying clustering to abstracted user sessions, it is more likely to find groups of sessions with similar pages that help increase the Markov model accuracy. For example, consider the four Web sessions in table 2, and the 2 clusters derived using the k-means clustering algorithm:

Table2: Example of user sessions.

W1	A, B, F, G, I
W2	A, C, D, G, I
W3	B, C, D, E, H
W4	B, C, D, E, F

Cluster 1:

W3	B, C, D, E, H
W4	B, C, D, E, F

Cluster 2:

W1	A, B, F, G, I
W2	A, C, D, G, I

Assuming that there is a new Web session: A, B, C, D what is the probability that the new page to be accessed by the user is page E? According to k-means clustering algorithm, and according to the distance measure between the new data points and the data points in the existing clusters, the new session belongs to cluster 1. The Markov model analysis performed on the subset cluster 1 yields a 1.0 probability for accessing page E next. However, performing Markov model analysis on the whole data set yields a 0.67 probability.

Experimental Evaluation

Data Collection and Preprocessing

For our experiments, the first step was to gather log files from active Web servers. Usually, Web log files are the main source of data for any e-commerce or Web related session analysis [HREF19]. The log file we used as a data source for our experiments is a day's worth of all HTTP requests to the EPA WWW server located at Research Triangle Park, NC. The logs are an ASCII file with one line per request, with the following information: The host making the request, date and time of request, requested page, HTTP reply code and bytes in the reply. The logs were collected for Wednesday, August 30 1995. There were 47,748 total requests, 46,014 GET requests, 1,622 POST requests, 107 HEAD requests and 6 invalid requests. Before using the EPA log file data, it was necessary to perform data preprocessing [HREF20, HREF21]. We removed erroneous and invalid pages. Those include HTTP error codes 400s, 500s, and HTTP 1.0 errors, as well as, 302 and 304 HTTP errors that involve requests with no server replies. We also eliminated multi-media files such as gif, jpg and script files such as js and cgi. The next step was to identify user sessions. A session is a sequence of URLs requested by the same user within a reasonable time. The user is uniquely defined by an IP address recorded in each http request within the time-frame of a single session. The end of a session is determined by a 30 minute threshold between two consecutive Web page requests. If the number of requests is more than the predefined threshold value, we conclude that the user is not a regular user; it is either a robot activity, a Web spider or a programmed Web crawler. Short sessions were also removed and only sessions with at least 5 pages were considered. the EPA preprocessing and filtering resulted in 799 Web sessions. The sessions of the data set are of different length. Web pages forming the sessions are in sequence of their user access. All sessions are represented by vectors with the number of occurrence of pages as weights. This will draw sessions with similar pages closer together when performing clustering techniques.

Distance Measures Evaluation

Table 3 lists entropy measures for only some of the clusters due to space limitation. The table demonstrates that, in general, Cosine and Pearson Correlation constitute better clusters than the other distance measures.

Table3: Entropy measures for different clusters.

Clusters	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**20**	**30**	**40**	**50**
Euclidean	0.42	0.38	0.32	0.58	0.31	0.28	0.25	0.30	0.26	0.21	0.19	0.23	0.22
City	0.52	0.48	0.50	0.49	0.46	0.42	0.39	0.31	0.29	0.27	0.25	0.24	0.23
Hamming	0.56	0.49	0.53	0.50	0.47	0.39	0.41	0.38	0.36	0.29	0.25	0.31	0.34
Cosine	0.36	0.32	0.37	0.43	0.25	0.21	0.22	0.21	0.17	0.16	0.19	0.22	0.23
Correlation	0.30	0.28	0.30	0.37	0.20	0.21	0.23	0.19	0.20	0.19	0.18	0.19	0.21

Our basic motivation behind clustering is to group functionally related sessions together based on Web services requested in order to improve the Markov model accuracy. The Markov model accuracy increases with the increase of the number of clusters due to the fact that more functionally related sessions are grouped together. However, Markov model computation complexity nature requires a limited number of clusters. Having this in mind and examining the entropy measures in Table 3 we conclude that using Cosine distance measure with the number of clusters (k)=7 will lead to good clustering results while keeping the number of clusters to a minimum. The 7 clusters were obtained in 17 iterations with the least sum of distances of 99.1192. Figure 2, Figure 3, Figure 4, Figure 5 and Figure 6 represent clusters using Euclidean, Hamming, City Block, Pearson Correlation and

Cosine distance measures respectively. They plot the silhouette value represented by the cluster indices displaying a measure of how close each point in one cluster is to points in the neighboring clusters. The silhouette measure ranges from +1, indicating points that are very distant from neighboring clusters, to 0, indicating points that do not belong to a cluster. The figures reveal that the order of distance measures from worst to best are Hamming, City Block, Euclidean, Pearson Correlation and Cosine respectively. For instance, the maximum silhouette value in Figure 3 for Hamming distance is around 0.5, whereas, the silhouette value of Figure 6 for Cosine distance ranges between 0.5 and 0.9. The larger silhouette value of the Cosine distance implies that the clusters are separated from neighboring clusters.

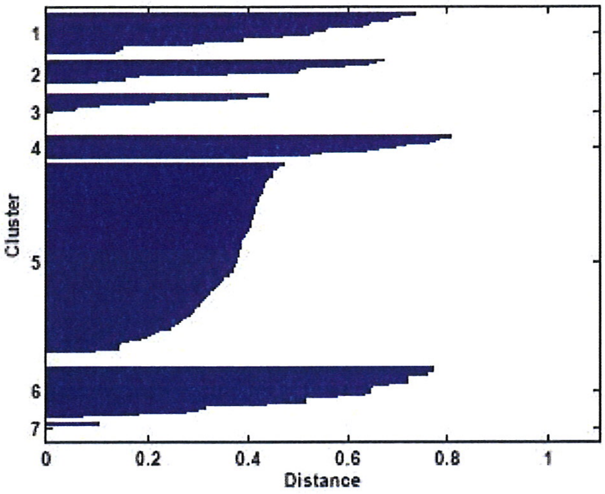


Figure 2: Euclidean distance measure with 7 clusters.

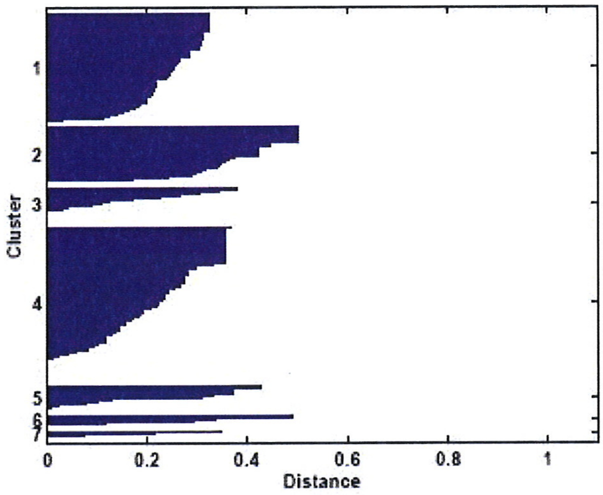


Figure 3: Hamming distance measure with 7 clusters.

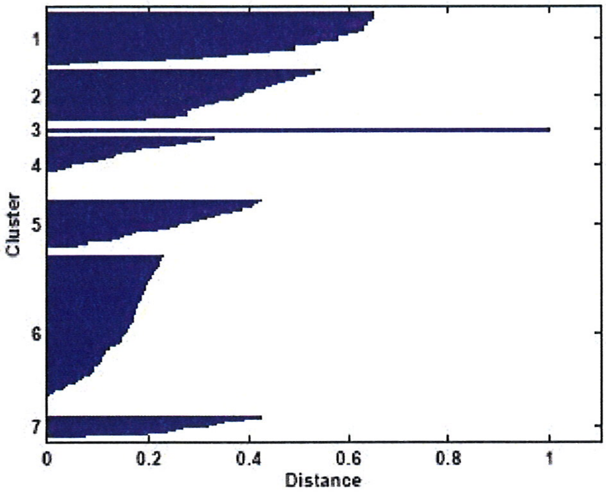
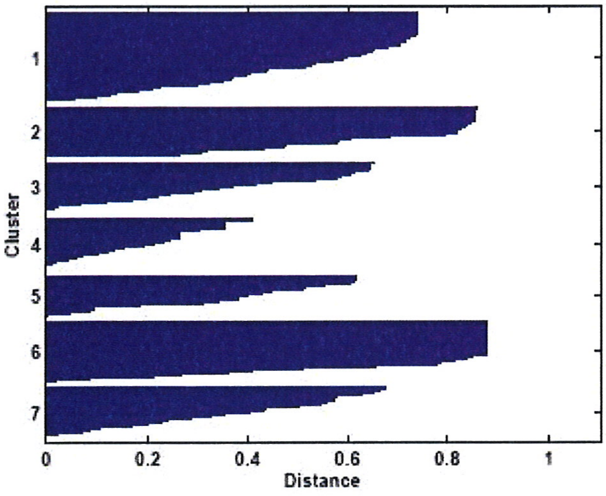


Figure 4: City Block distance measure with 7 clusters.



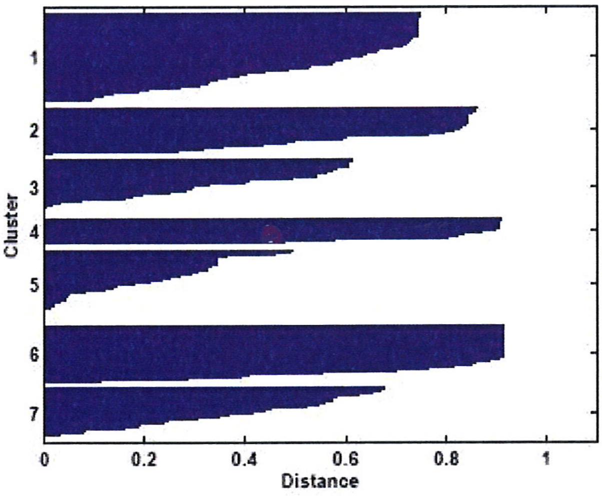


Figure 5: Correlation distance measure with 7 clusters.

Figure 6: Cosine distance measure with 7 clusters.

Figure 7 reveals the mean value of distances for different clusters. It is calculated by finding the average of distance values between points within clusters and their neighboring clusters. The higher the mean value, the better clusters we get. It is worth noting that the information Figure 7 provides does not prove much on its own because it does not take into consideration points distribution within clusters.

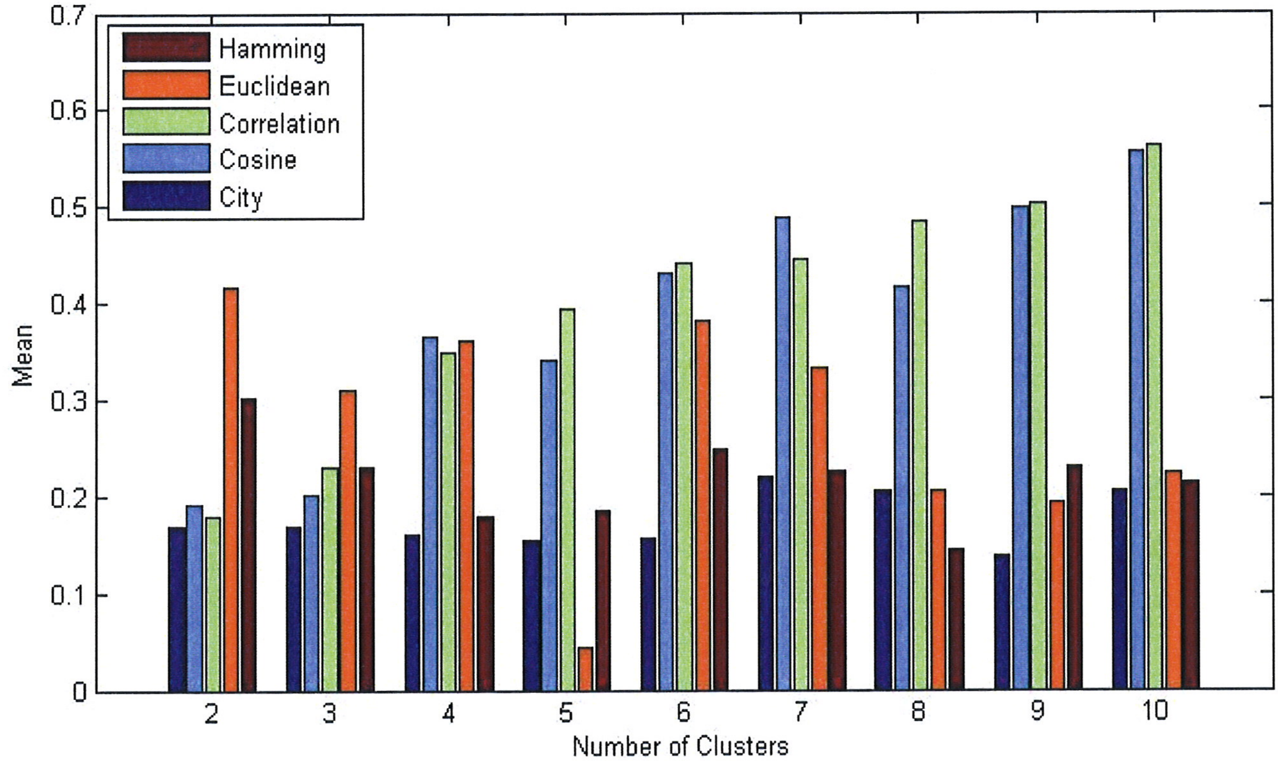


Figure 7: The mean value for 2 ... 10 clusters using different distance measures

The results of the distance plots in Figures 2-6, the distance mean values in Figure 7 as well as the entropy calculations all reveal that Cosine and Pearson Correlation form better clusters than Euclidean, City Block and Hamming distance measures. Based on this information, we choose Cosine measures with k=7 for the prediction accuracy evaluation.

Experiment Results

Merging Web pages by web services according to functionality reduces the number of unique pages from 2924 to 155 categories. The sessions were divided into 7 clusters using the k-means algorithm and according to the Cosine distance measure. For each cluster, the categories were expanded back to their original form in the data set. This process is performed using a simple program that seeks and displays the data related to each category. If we consider the categorization example in section 3, cie category will be expanded back to cie/metadata.txt.html cie/index.html cie/summer95 and cie/summer95/articles. If a user accesses cie/index.html, there is a chance he/she will access cie/summer95 then cie/summer95/articles next. Markov model implementation was carried out for the whole data set. The data set was divided into training set and test set and 2-Markov model accuracy was calculated accordingly. Then, using the test set, each transaction was considered as a new point and distance measures were calculated in order to define the cluster that the point belongs to. Next, 2-Markov model prediction accuracy was computed considering the transaction as a test set and only the cluster that the transaction belongs to as a training set. Prediction accuracy results were achieved using the maximum likelihood based on conditional probabilities as stated in equation 3 above. All predictions in the test data that did not exist in the training data sets were assumed incorrect and were given a zero value. All implementations were carried out using MATLAB. The Markov model accuracy was calculated using a 10-fold cross validation. The data was split into ten equal sets. First, we considered the first nine sets as training data and the last set for test data. Then, the second last set was used for testing and the rest for training. We continued moving the test set upward until the first set was used for testing and the rest for training. The reported accuracy is the average of ten tests. Figure 9 compares the Markov model accuracy of the whole data set to Markov model accuracy using clusters based on Euclidean, Correlation and Cosine distance measures with k=7.

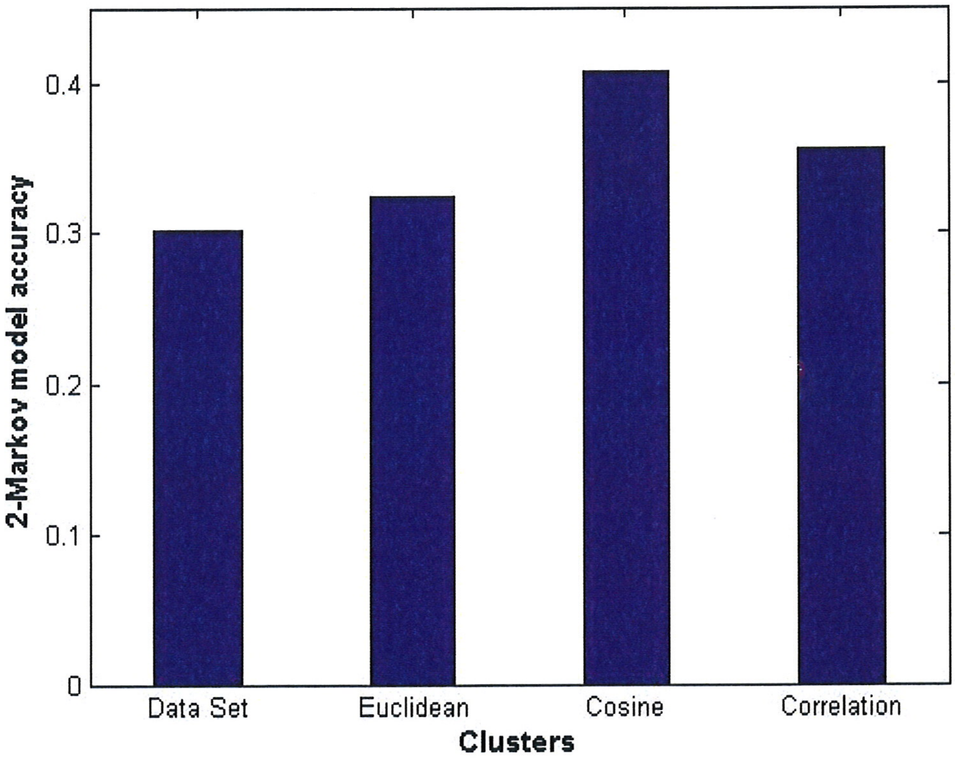


Figure 8: Markov model accuracy of whole data set and Markov model accuracy using clusters based on Euclidean, Correlation and Cosine distance measures with k=7.

All clustering runs were performed on a desktop PC with a Pentium IV Intel processor running at 2 GHz with 2 GB of RAM and 100 GB of hard disk memory. In our largest runs with K = 50, we exhausted around 6.1 MB of memory in 34 seconds. The runtime of the k-means algorithm, regardless of the distance measure used, is equivalent to *O(nkl)* [HREF15], where *n* is the number of items, *k* is the number of clusters and *l* is the number of iterations taken by the algorithm to converge. For our experiments, where *n* and *k* are fixed, the algorithm has a linear time complexity in terms of the size of the data set. The k-means algorithm has a *O(k + n)* space complexity. This is because it requires space to store the data matrix. It is feasible to store the data matrix in a secondary memory and then the space complexity will become *O(k)*. k-means algorithm is more time and space efficient than hierarchical clustering algorithms with $O(n^2 \log n)$ time complexity and $O(n^2)$ space complexity. As for all 2nd order Markov model, the running time of the whole data set was similar to that of the clusters added together because the running time is in terms of the size of the data. i.e. T(n)=T(k1)+T(k2)+T(k3)+...T(ki) where time is denoted by T, the number of items in the data set is denoted by n, and the clusters are denoted by ki.

Conclusion

This paper improves the overall prediction accuracy by grouping the data set sessions into clusters. The Web pages in the user sessions are first allocated into categories according to Web services that are functionally meaningful. Then, k-means clustering algorithm is implemented using the most appropriate number of clusters and distance measure. Prediction techniques are applied using each cluster as well as using the whole data set. The experimental results reveal that implementing the k-means clustering algorithm on the data set improves the accuracy of the next page access prediction. The prediction accuracy achieved is an improvement to previous research papers that addressed mainly recall and coverage.

References

A. Banerjee and J. Ghosh. Clickstream clustering using weighted longest common subsequences. SIAM Conference on Data Mining, Chicago, pages 33-40, 2001.

I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Model-based clustering and visualization of navigation patterns on a web site. Data Mining and Knowledge Discovery, 7, 2003.

M. Deshpande and G. Karypis. Selective models for predicting web page accesses. Transactions on Internet Technology, 4(2):163–184, 2004.

C. F. Eick, N. Zeidat, and Z. Zhao. Supervised clustering -algorithms and benefits. IEEE ICTAI'04, pages 774–776, 2004.

S. Gunduz and M. T. OZsu. A web page prediction model based on clickstream tree representation of user behavior. SIGKDD'03, USA, pages 535- 540, 2003.

M. Halkidi, B. Nguyen, I. Varlamis, and M. Vazirgiannis. Thesus: Organizing web document collections based on link semantics. The VLDB Journal, 2003(12):320–332, 2003.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. ACM Computing Surveys, 31(3):264–323, 1999.

D. Kim, N. Adam, V. Alturi, M. Bieber, and Y. Yesha. A clickstreambased collaborative filtering personalization model: Towards a better performance. WIDM '04, pages 88–95, 2004.

J. Pitkow and P. Pirolli. Mining longest repeating subsequences to predict www surfing. USENIX Annual Technical Conference, pages 139–150, 1999.

R. Sarukkai. Link prediction and path analysis using markov chains. 9th International WWW Conference, Amsterdam, pages 377–386, 2000.

M. Spiliopoulou, L. C. Faulstich, and K. Winkler. A data miner analysing the navigational behaviour of web users. Workshop on Machine Learning in User Modelling of the ACAI'99, Greece, 1999.

J. Srivastava, R. Cooley, M. Deshpande, and P. Tan. Web usage mining: Discovery and applications of usage patterns from web data. SIGDD Explorations, 1(2):12–23, 2000.

A. Strehl, J. Ghosh, and R. J. Mooney. Impact of similarity measures on web-page clustering. AI for Web Search, pages 58–64, 2000.

Q. Wang, D. J. Makaroff, and H. K. Edwards. Characterizing customer groups for an e-commerce website. EC'04, USA, pages 218–227, 2004.

H. Xiong, J. Wu, and J. Chen. K-means clustering versus validation measures: A data distribution perspective. KDD'06, USA, pages 779–784, 2006.

Q. Zhao, S. S. Bhomick, and L. Gruenwald. Wam miner: In the search of web access motifs from historical web log data. CIKM'05, Germany, pages 421–428, 2005.

S. Zhong and J. Ghosh. A unified framework for model-based clustering. Machine Learning Research, 4:1001–1037, 2003.

J. Zhu, J. Hong, and J. G. Hughes. Using markov models for web site link prediction. HT'02, USA, pages 169–170, 2002.

Hypertext References

HREF1
    http://www.usq.edu.au/
HREF2
    http://www.unisa.edu.au/
HREF3
    http://www.sci.usq.edu.au/staff/wang/
HREF4
    http://doi.acm.org/10.1145/1031470
HREF5
    http://www.usenix.org/publications/library/proceedings/usits99/pitkow.html
HREF6
    http://www.lans.ece.utexas.edu/~abanerjee/papers/01/lcs.pdf
HREF7
    http://citeseer.ist.psu.edu/zhu02using.html
HREF8
    http://www.datalab.uci.edu/papers/webcanvas.pdf

HREF9
http://portal.acm.org/citation.cfm?doid=988805

HREF10
http://www.acm.org/sigs/sigkdd/explorations/issue1-2/srivastava.pdf

HREF11
http://jmlr.csail.mit.edu/papers/volume4/zhong03a/zhong03a.pdf

HREF12
http://www.lans.ece.utexas.edu/~strehl/strehl-abstracts.html

HREF13
http://portal.acm.org/citation.cfm?coll=GUIDE&dl=GUIDE&id=956815

HREF14
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1374270

HREF15
http://www.cs.rutgers.edu/~mlittman/courses/lightai03/jain99data.pdf

HREF16
http://wotan.liu.edu/docis/dbl/vldbjo/2003_12_4_320_TOWDCB.html

HREF17
http://portal.acm.org/citation.cfm?id=1150402.1150503

HREF18
http://ieeexplore.ieee.org/iel5/8719/27592/01232044.pdf

HREF19
http://ka.rsten-winkler.de/publications/WUM_ACAI1999_CameraReady.pdf

HREF20
http://www9.org/w9cdrom/68/68.html

HREF21
http://portal.acm.org/ft_gateway.cfm?id=1099679&type=pdf

Copyright