

Impact of Behavior Clustering on Web Surfer Behavior Prediction^{*}

MALIK TAHIR HASSAN AND ASIM KARIM

*Department of Computer Science
LUMS School of Science and Engineering
Lahore, 54792 Pakistan*

We investigate Web surfer behavior prediction by building generative and discriminative models on the entire history of navigation paths and on behavior clustering of the history. The underlying question that we try to answer is: Does behavior clustering improve behavior prediction? For behavior clustering, we adapt the k -modes clustering algorithm by incorporating a new similarity measure that gives greater weight to matches at the beginning of the navigation path. The initial cluster representatives are selected from the set of most dissimilar paths which also fixes the number of clusters. For generative prediction, we adopt Markov chain Bayesian classification models whereas for discriminative prediction we build SVM models. Experiments are performed on two real-world data sets. Surprisingly, the results show that behavior clustering has no significant impact on Web surfer behavior prediction. We also investigate the impact of time of visit, the number of relevant clusters used in prediction models, and the use of cluster modes on Web surfer behavior prediction. We find that for limited scope data simpler approaches such as prediction using cluster modes can produce highly accurate predictions (less than 1% drop from the best prediction) with greater efficiency.

Keywords: clustering, navigation path prediction, order weighted similarity, sequence prediction, web usage mining, generative discriminative models

1. INTRODUCTION

It is often stated that history repeats itself. Hence, so should Web surfers' navigation paths as humans have the habit of repeating their actions. Similarly, it has been observed that people tend to form groups naturally based on their interests and behaviors. These observations suggest that behavior clustering may improve Web surfer behavior prediction. Web surfer behavior prediction is the process of determining key navigation patterns of users in the future given their navigation path history. Such predictions have immense commercial value as the Web evolves into a primary medium for marketing and sales for many businesses. Web-based businesses seek useful users' patterns to help identify promising events, potential risks, and make strategic decisions.

Web surfer behavior modeling and prediction is a complex machine learning problem that requires careful consideration. We consider the problem of predicting page categories visited by users in first three positions of their navigation paths, where a specific page category refers to Web pages on a particular theme *e.g.* sports, news, and education. This is a simpler problem that can be modeled more readily as compared to the general Web surfer behavior prediction problem [1]. Predicting multiple ordered values, sequence prediction, is of practical interest in many areas besides Web mining, *e.g.*, fore-

Received July 14, 2010; revised November 14, 2010; accepted January 12, 2011.

Communicated by Chih-Jen Lin.

^{*} We gratefully acknowledge the support from Lahore University of Management Sciences (LUMS) and Higher Education Commission (HEC) of Pakistan.

casting the weather of next three days, predicting average stock value in next three weeks, *etc.*

Clustering is the process of dividing objects into groups such that the objects in a group are similar to one another but dissimilar to the objects in other groups. Various techniques of clustering are applied in several areas including image segmentation, object recognition, information retrieval and data mining [2]. Behavior clustering is commonly employed for user segmentation and navigation pattern visualization [3, 4]. Its use in behavior prediction has not been investigated extensively. In particular, the following questions need to be addressed: Does behavior clustering improve behavior predictions? Are clustering-based predictions more efficient? What is the efficiency-effectiveness trade-off of predictions done with and without clustering? We investigate these questions by evaluating the performance of generative and discriminative prediction models when they are built from the entire history of navigation paths and when they are based on clusters of navigation paths. Our generative models are based on Markov chain and Bayesian classification, whereas discriminative models are built using Support Vector Machine (SVM). We perform behavior clustering by adapting the k -modes algorithm [5] with a new similarity measure and a new strategy for selecting seeds. Performance is evaluated on two real-world public data sets. Results show high accuracy comparable to those reported in the literature. Behavior clustering improves Web surfer behavior prediction in a few scenarios only but clustering-based predictions are often more practical than predictions based on un-clustered data because of efficiency. We also investigate several variations of our framework including varying the number of clusters, predicting based on top r relevant clusters only, and predicting using cluster representatives only.

Our main contributions include (i) a simple clustering method for order-weighted nominal sequences, (ii) generative and discriminative models for navigation path prediction, (iii) impact of behavior clustering on Web surfer behavior prediction, and (iv) evaluation on two real-world data sets.

The rest of the paper is organized as follows. We present the related work in section 2. Section 3 defines the Web surfer behavior prediction problem. Our behavior clustering and prediction framework is described in section 4. Experimental evaluation of the models is given in section 5. We conclude in section 6.

2. RELATED WORK

Web surfer behavior modeling and prediction has been studied extensively in the literature. Here, we restrict our discussion to works that use navigation path information only. A majority of such works adopt probabilistic methods for Web surfer behavior prediction [6-10]. Borges and Levene [6] model navigation paths as N -gram probabilistic grammars which assume that the probability of visiting a Web page depends on the previous N pages visited. Deshpande and Karypis [7] propose the use of all k th order Markov models and suggest that many of the non-affecting states in a k th-order Markov model can be pruned to reduce the state space complexity. Awad *et al.* [8] combine Markov models and SVM using Dempster's rule for surfing predictions. They incorporate domain knowledge also but do not study the impact of clustering the history. These probabilistic methods are overly complicated for predicting navigation patterns of prac-

tical value like predicting the first three pages viewed by users. For this reason, we adopt simpler prediction models as done by Hassan *et al.* [10].

Web data clustering has been employed for user segmentation, behavior visualization, and to a lesser extent, prediction. Lu *et al.* [11] generate Significant Usage Patterns by clustering clickstream data using pair-wise alignment and then building a first-order Markov model for each of the clusters. A similar problem and solution is proposed by Halvey *et al.* [12]. They use clustering based on *time* and show improvement in navigation prediction through an empirical analysis on a sample of usage logs for Wireless Application Protocol browsing. Our approach focuses on key navigation patterns or behaviors and is based on behavior clustering (not time clustering). Pallis *et al.* [13] have also done clustering of users' navigation sessions but their focus is on identifying patterns and similarities, and not navigation prediction. Sequence based clustering of Web sessions has been done by Park *et al.* [14]. They investigate the impact of different sequence representations, similarity measures, number of clusters, number of web pages *etc.* on clustering. We on the other hand study the impact of clustering on navigation prediction.

Niu *et al.* [15] extend the Markov model approach by introducing a similarity measure for navigation paths. They group the Web users based on their access logs and predictions are then done based on the historical records in a single group. We use a new similarity measure and a simpler prediction model. Cadez *et al.* [4] have developed a tool for visualizing navigation patterns using model based clustering. Their emphasis is not on prediction of visits but on the visualization of behavior clusters. Another work focusing on behavior cluster visualization is reported by Kim [3]. His similarity measure, which gives greater weight to matches at beginning of navigation paths, is similar to ours. Liu and Keselj [16] cluster the user sessions to get representative navigation patterns and use them in combination with the contents of the web pages to classify and predict future requests. In our approach the contents of the web pages are considered to be unknown.

Li and Lee [17] propose a single pass algorithm to find top- k path traversal patterns for clickstream data. The algorithm, DSM-TKP (Data Stream Mining for Top- K Path traversal patterns), uses a summary data structure TKP-forest (a forest of Top- K Path traversal patterns). Pitman *et al.* [18] use clustering of users' web sessions to extract business intelligence from weblogs of a tourism portal. After applying Principle Component Analysis (PCA), they use x -means [19] to find that different user groups have significantly different information needs. Their focus is analysis instead of prediction.

3. WEB SURFER BEHAVIOR PREDICTION PROBLEM

Learning to predict the navigation paths of users given their navigation history is commonly referred to as Web surfer behavior prediction. A navigation path is a sequence of Web page categories (or Web pages) that a user visits on a particular visit session. For a particular page category, a user may browse through a number of pages belonging to that category before terminating the session or moving on to another category.

Let $\mathbf{D} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$ be the historical data of navigation paths for all users, where $\mathbf{p}_i$ identifies the i th path and N is the total number of paths in the data. A navigation path $\mathbf{p}$ is defined by the vector $\mathbf{p} = [u, t, c_1, c_2, \dots, c_n]$ where u is the user ID, t is the start time of the session, and c_i is the page category visited at position i of the sequence. A naviga-

tion path can have a finite number $n \geq 1$ of page categories c_i such that $c_i \neq c_{i+1}$, for all $i \leq n - 1$ (*i.e.* a category transition must occur at each position i for $i < n$). In other words, a navigation path $\mathbf{p}$ is an (x, y) pair in which $x = [u, t]$ defines the particulars of the path and $y = [c_1, c_2, \dots, c_n]$ defines the sequence of page categories visited in the path.

Instead of predicting the entire sequence of c_i ($i = 1, \dots, n$), we focus on a simpler pattern of practical value: predicting the first three page categories in a session. However, the proposed models can be extended easily to the prediction of any number of categories. The choice of the first three categories is an experimental convenience, especially when the focus of the paper is on the impact of behavior clustering on surfing prediction rather than prediction alone. Same or similar problem of predicting first three or next three items only is also discussed in literature [1, 10, and 20]. Thus, given the above problem setting, the task is to learn to predict for known users the first three page categories visited, *i.e.*, c_1, c_2, c_3 . Notice that the desired outcome is in general not one label but a sequence of labels.

4. CLUSTERING-PREDICTION FRAMEWORK FOR WEB SURFER BEHAVIOR PREDICTION

We formulate and evaluate models based on: behavior clustering, Bayesian or SVM prediction, and a combination of both. The latter combined approach, which captures the key ideas from the previous two, consists of two steps. In the first step, unsupervised behavior clustering is done. The training data is grouped into clusters relying on the page category sequence information only (the c_i , $i = 1, \dots, n$, in $\mathbf{p}$). In the second step, supervised learning is done from the clusters. We adapt the k -modes clustering algorithm for behavior clustering and use Markov chain based generative models and SVM based discriminative models for prediction. The steps are described in the subsequent sections.

4.1 Behavior Clustering

In the first step, the historical data of navigation paths, $\mathbf{D}$, is partitioned into k clusters based on the similarity between sequences of categories. Clustering is done by an adapted version of the k -modes algorithm [5]. k -modes works similar to the iterative partitioning algorithm k -means, but in k -modes the centroid of a partition is the mode instead of the arithmetic mean, and a suitable similarity measure is used to handle the categorical data. Our adaptations include strategies for selecting the parameter k and the initial cluster representatives, and use of a new similarity measure called *MatchScore*.

4.1.1 Selecting k and the k initial paths

In most partitioning-based clustering algorithms, including the k -modes algorithm, two important decisions have to be made before their application: the selection of the desired number of clusters k and the selection of the initial cluster representatives or seeds. Several strategies have been proposed for this. Pena *et al.* [21] provide a comparison of four initialization methods for k -means: random, Forgy, MacQueen, and Kaufman. Bradley *et al.* [22] estimate the distribution of modes to provide better initial representa-

tives for k -means clustering. Pelleg *et al.* [19] estimate the number of clusters by using a statistically based criterion that maximizes the model's posterior probability.

Our strategy is simpler and relevant to the problem. We set the maximum possible value of k (denoted by m) to be the maximum number of navigation paths with zero pair wise similarity. These paths represent the most dissimilar paths in the navigation path history and hence are ideal for the initial seeds for clustering. This strategy defines both the number of desired clusters and the initial seeds for clustering. In our implementation, we experiment with $k = m$, $k \approx 0.5m$, $k \approx 0.25m$, and $k = 1$ (no clustering) by randomly selecting the k most dissimilar navigation paths as initial cluster representatives.

4.1.2 MatchScore similarity measure

We propose a new similarity measure called *MatchScore* designed specifically for clustering of navigation paths. This measure is motivated from the evaluation criterion used by the 2007 ECML/PKDD Discovery Challenge [1] and is similar to the order-weighted measure used by [3]. Given two navigation paths $\mathbf{p}_1$ and $\mathbf{p}_2$, the *MatchScore*($\mathbf{p}_1$, $\mathbf{p}_2$) is the sum of scores for matching categories in the two paths, where the scores are based on the positions at which the categories match. Mathematically,

$$MatchScore(\mathbf{p}_1, \mathbf{p}_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} IsAMatch(c_i, c_j) \times \min(s_i, s_j) \quad (1)$$

where n_1 and n_2 are the lengths of the navigation paths $\mathbf{p}_1$ and $\mathbf{p}_2$, respectively, c_i denotes the category visited at position i , and s_i is the score for a match at position i . The function *IsAMatch*(c_i, c_j) returns 1 when $c_i = c_j$ and this category has not been matched earlier; otherwise, the function returns zero. This means that once a category pair has been matched (at same or different position) it is not included in the score again.

The score s_i gives greater weight to a match in the beginning of the path. In our work, we consider matches in the first three positions, and define s_i as 5, 4, or 3 for i equal to 1, 2, and 3, respectively. With this definition, *MatchScore* can range from 0 (no similarity) to 12 (maximum similarity). An example of *MatchScore* calculation is given in Table 1.

Table 1. MatchScore calculation example, MatchScore($\mathbf{p}_1$, $\mathbf{p}_2$).

Categories in $\mathbf{p}_1$:	9	17	15	2	$= 1 \times \min(5, 5) + 0 \times \min(5, 4) + 0 \times \min(5, 3) +$ $0 \text{ (already matched)} + 0 \times \min(4, 4) + 0 \times \min(4, 3) +$ $0 \text{ (already matched)} + 1 \times \min(3, 4) + 0 \times \min(3, 3)$ $= 5 + 3 = 8$
Categories in $\mathbf{p}_2$:	9	15	2		

Our similarity measure captures the fact that matches in the beginning of visit sessions are more important than those occurring later in the session. Furthermore, it provides intuitive solutions to category matches that do not occur at the same position and for repeating categories in the navigation paths. Note that other existing measures like Kendall's rank correlation and Spearman's rank correlation [23] are not suitable here as the problem is different from simple ranking. These ranking coefficients are unable to capture the order-weighted semantics of navigation paths. Kendall tau distance checks

the pairwise disagreements between two lists but does not give scores to partial matches as precisely as our *MatchScore*. Simple matching does not help too as the matches in early positions are considered more important here, and should be given higher weight.

4.1.3 *k*-modes clustering algorithm

We use a modified version of *k*-modes clustering algorithm to group together the navigation paths exhibiting similar surfing behavior. We cluster with different values of *k* ($1 \leq k \leq m$), where *m* is the maximum number of paths in the training data with zero pairwise *MatchScore*. These *m* navigation paths are taken as initial cluster representatives or modes. The algorithm is executed starting from these modes adopting the *MatchScore* similarity measure. The algorithm is:

Input: The number of clusters *k* and a database containing *N* navigation paths.

Output: A set of *k* clusters that maximizes the overall *MatchScore* or *ClusteringScore*.

Method: (1) choose *k* unique navigation paths as the initial cluster seeds;
 (2) repeat
 (3) (re)assign each navigation path to the cluster to which it is the most similar, based on the mode of the navigation paths in the cluster;
 (4) update the cluster modes, *i.e.*, calculate the mode of the navigation paths for each cluster;
 (5) until overall *MatchScore* maximizes

The clustering procedure is stopped when the overall *MatchScore*, or *ClusteringScore*, remains unchanged from one iteration to the next. The overall *MatchScore* is:

$$ClusteringScore = \sum_{i=1}^k \sum_{p \in L_i} MatchScore(p, d_i) \quad (2)$$

where *p* is a navigation path in cluster L_i and d_i is the mode of cluster L_i (*i.e.* the most frequent sequence in cluster L_i).

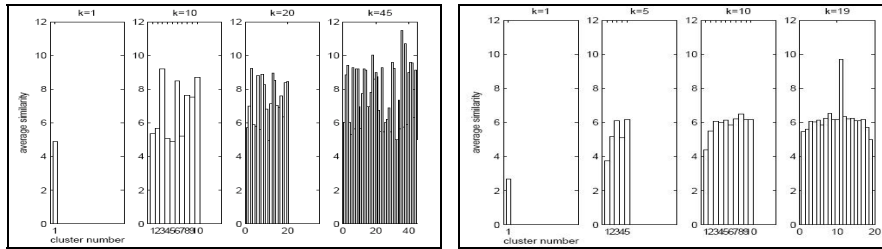


Fig. 1. Average intra-cluster similarity for different values of *k* (ECML data left, KDD data right).

Fig. 1 verifies the correctness of the clustering algorithm. It shows the average intra-cluster similarity of each cluster for different values of *k* (the number of clusters).

The average intra-cluster similarity of a cluster is the average similarity of data points in the cluster from the cluster's representative (mode). By definition of clustering,

increasing the value of k should result in more homogeneous clusters. This trend is highlighted in this figure for two different Web navigation data sets. This verifies that our algorithm is successful in grouping similar navigation paths in the same cluster. The data sets utilized in this figure are described in section 5.1.

4.2 Behavior Prediction

Two popular types of classification and prediction models have been used in practice: generative and discriminative. Generative models try to learn the posterior probability distribution by first estimating the probability distribution of the data. On the other hand, discriminative models try to learn a discriminant function directly from data and use this information to classify new data points. Each type of models has its pros and cons.

We select the most representative models/methods from each type so as to ensure a fair evaluation of the impact of behavior clustering on prediction. Section 4.2.1 presents Bayesian and Markov models that are generative in nature, while section 4.2.2 presents a SVM based model which is discriminative in nature.

4.2.1 Bayesian prediction

We assume that the categories at positions 1, 2 and 3 form a Markov chain. The first category starts this chain and is determined by a Bayes classifier trained on the navigation path data. The second and third categories are decided by considering the Bayes classification at that position and the transition probability from the previous position. The reason for selecting the first-order Markov model over higher order models is two-fold: (1) the problem involves the prediction of only the first three page categories for which a first-order Markov model is sufficiently accurate; (2) the first-order Markov model is computationally efficient as compared to higher-order Markov models.

According to the Bayes rule, the posterior probability of page category C_j visited in position j ($j = 1, 2$, or 3) of a visit session $\mathbf{x}$ in cluster(s) L is given by

$$P^B(C_j | \mathbf{x}, L) = \frac{P(\mathbf{x} | C_j, L)P(C_j | L)}{P(\mathbf{x} | L)}. \quad (3)$$

The conditioning on L emphasizes that the probabilities are estimated from the data in L . This can be the data in the most relevant cluster, in top r most relevant clusters, or in the entire data (all clusters). With this notation, the same formulation can be used for prediction models built with and without clustering. The relevance of a cluster to an input $\mathbf{x}$ is determined by the frequency with which the input occurs in the cluster.

The most probable page category visited at the start of the sequence ($C_1 = c_1$) is given by:

$$c_1 = \arg \max_c P(\mathbf{x} | C_1 = c, L)P(C_1 = c | L). \quad (4)$$

This fixes the start state of the Markov chain. The subsequent states can be found by combining the predictions of the Bayes classifier (Eq. (3)) and the Markov model. According to the Markov property, for a given visit session $\mathbf{x}$ the posterior probability of

page category C_j visited in position j ($j = 2, 3$) depends only on C_{j-1} and can be expressed as

$$P^M(C_j | C_{j-1}, \mathbf{x}, L) = \frac{P(C_j \wedge C_{j-1} \wedge \mathbf{x} \wedge L)}{P(C_{j-1} \wedge \mathbf{x} \wedge L)} \quad (5)$$

$$= \frac{P(\mathbf{x} | C_j, C_{j-1}, L)P(C_j | C_{j-1}, L)}{P(\mathbf{x} | C_{j-1}, L)}. \quad (6)$$

Using Eqs. (3) and (6), the page category ($C_j = c_j$) visited at position j ($j = 2, 3$) is given by

$$c_j = \arg \max_c P^B(C_j = c | \mathbf{x}, L)P^M(C_j = c | C_{j-1}, \mathbf{x}, L). \quad (7)$$

This equation is based on the assumption that the predictions of the Bayes and Markov models are independent. Note that the denominators in Eqs. (3) and (6) can safely be ignored in computation of Eq. (7).

These Bayesian models are based on knowledge of the visit session, $\mathbf{x} = [u, t]$, where u is the user ID and t is the timestamp of the session. When both u and t are considered in the models the naive Bayes assumption of conditional independence can be invoked to simplify the joint probability of u and t into the product of probabilities of u and t .

Bayesian Prediction algorithm is:

Training	Testing
Input: Clusters of training data, (x, y) pairs, where $x = [u, t]$, $y = [c_1, c_2, \dots, c_n]$	Input: Estimated probabilities; Test data <i>i.e.</i> x part only
Output: Estimated probabilities	Output: Predicted categories, $y = [c_1, c_2, c_3]$
Method: (1) Estimate probabilities in Eqs. (3) and (6)	Method: (1) Repeat (2) Take next test example, x ; (3) Select L , the most relevant cluster of x ; (4) Calculate c_1 , as in Eq. (4); (5) Calculate c_2 and c_3 , using Eq. (7); (6) Output the predicted categories c_1, c_2, c_3 ; (7) Until no test example left

4.2.2 SVM prediction

SVMs are popular discriminative classifiers that learn a maximum margin hyper plane for separating two classes in a feature space defined by a kernel function. We present a simple yet effective SVM-based approach for Web surfer behavior prediction that takes into account the dependence of page categories and navigation path clustering.

As defined in section 3, each navigation path is an $(\mathbf{x}, \mathbf{y})$ pair where $\mathbf{x} = [u, t]$ contains the particulars of the navigation path and $\mathbf{y} = [c_1, c_2, \dots, c_n]$ is the sequence of page categories visited in the navigation path. To predict the first three page categories browsed in a navigation path, the following SVM-based algorithm is adopted:

Training

Input: Input data (x, y) in Weka format
Output: Prediction models: SVM₁, SVM₂ and SVM₃
Method:
 (1) Train SVM₁ on x to predict c_1
 (2) Train SVM₂ on x & c_1 to predict c_2
 (3) Train SVM₃ on x , c_1 & c_2 to predict c_3
 (4) Output SVM₁, SVM₂ and SVM₃

Testing

Input: Prediction models: SVM₁, SVM₂ and SVM₃,
 Test data in Weka format, (x) part only
Output: Predicted data, $y = [c_1, c_2, c_3]$
Method:
 (1) Predict c_1 using SVM₁ and x
 (2) Predict c_2 using SVM₂, x and c_1 from step 1
 (3) Predict c_3 using SVM₃, x , c_1 from step 1
 and c_2 from step 2
 (4) Output the predicted categories c_1, c_2, c_3

If data have been clustered, as described in section 4.1.3, the above SVMs can be modified to include clustering information for cluster-based behavior prediction. This is done by adding a new input feature that identifies the most relevant cluster for the user.

5. EXPERIMENTAL EVALUATION

The evaluations include impact of number of clusters (k) formed by behavior clustering, number of most relevant clusters (r) used by prediction models, using time stamp in prediction models, and using cluster modes for prediction. We present results for two real data sets.

5.1 Data and Their Characteristics

5.1.1 ECML data

The 2007 ECML/PKDD Discovery Challenge [1] data set (ECML data) comprises of 4 weeks of Web surfing. The first 3 weeks of data are used for training while the last week's data are reserved for testing. An example record of the data is given below.

Path ID	User ID	Timestamp	Path
27	1	1169814548	(7, 3), (17, 3) ...

Path ID and user ID are the unique identifiers; Timestamp is the UNIX time at which the session started. Path is a sequence of (page category ID, number of pages) pairs identifying the category and the number of pages in that category visited. We find that there are 20 unique categories. The three weeks training data comprise of 379,485 sessions whereas the one week test data contain 166,299 sessions. There are 4,882 distinct users having a non-uniform distribution. The minimum and maximum number of visits by a user in the training data is 7 and 497, respectively, with an average of 77.7 visits per user. The minimum and maximum number of visits by a user in the test data is 1 and 215, respectively. Page categories are also non-uniformly distributed. ECML data set tracks users on several sites across a substantial geographical region and has a wider scope as compared to the second data set we use *i.e.* KDD data set.

For ECML data, the maximum number of clusters is 19, *i.e.* $m = 19$ (the maximum number of dissimilar paths). After clustering, we find that users spread over multiple clusters and this spread resembles a skewed normal distribution with long tail (Fig. 2,

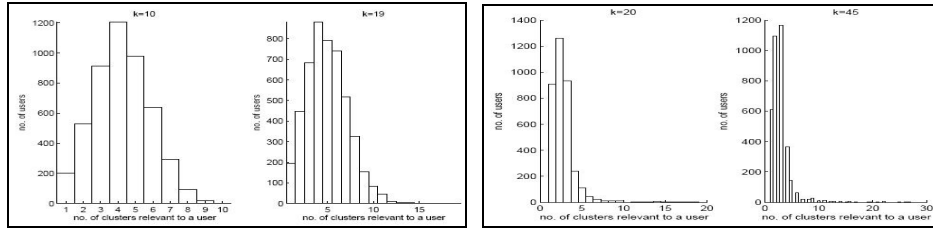


Fig. 2. Distribution of users across behavior clusters (ECML data on left, KDD data on right).

left). A vast majority of users are present in a few number of behavior clusters; however, few users are present in only one behavior cluster. Thus many users are exhibiting multiple behaviors.

5.1.2 KDD cup data

The KDD Cup 2000 [24] data set (KDD data) contains Web click-stream and purchase transactions. The data span over a period of three months. The first two months are for training purpose and the last month for testing. This data contain more than 200 attributes including user's demographic information, session information, *etc.* As we are interested in navigation path information only, we preprocess the data to the same format as the ECML data described above. For this purpose; Request Date, Request Date Time, Request Sequence, and Session Cookie ID attributes are used. Every request template (the dynamic page requested by the user) in the original data is considered a separate category (*e.g.* home, lifestyles, leg news *etc.*). These categories are assigned unique numeric IDs. A category includes links to other categories and static Web pages which is used to form the sequence of categories and views per category. The processed data contain 85 unique categories, 3,577 distinct users, 23,911 visit sessions for training, and 4,690 visit sessions for testing. Users and categories have a non-uniform distribution.

The maximum number of clusters for this data is 45 *i.e.* $m = 45$ (the maximum number of dissimilar navigation paths). After behavior clustering, the distribution of users across clusters is a bit different from the ECML data. Fig. 2 (right) shows that the majority of users is presented in 2 or 3 clusters, *i.e.*, the spread of users across clusters is somewhat narrower than the ECML data. This is because of the limited scope of this data with all pages belonging to one site and with most users starting surfing from one page.

5.2 Evaluation Criteria

For each predicted navigation path, the three predicted page categories are compared with the page categories in actual navigation path and a score is generated. This score is the sum of weights assigned to the three predicted categories. The (average) percent score for predicting N navigation paths is given by

$$\text{Percentscore} = \frac{1}{N} \sum_{i=1}^N \frac{\text{Score}(i)}{\text{MaxScore}} \times 100$$

where $\text{Score}(i)$ is the score for navigation path i and MaxScore is the maximum possible score for a prediction.

The calculation is very similar to our order weighted similarity measure discussed in section 4.1.2. The difference here is that the number of categories in actual navigation path is not fixed to be three. For instance, If the first, second, and third categories are predicted correctly, then assign weights 5, 4, and 3, respectively, to these positions. If a prediction is incorrect for the category at first position, then it is assigned a weight of 4 if that category occurs in the second position, 3 if it occurs in the third position, 2 if it occurs in the fourth position, 1 if it occurs in position five and beyond, and 0 if it does not occur. The weight assigned cannot be greater than the maximum possible for that position (e.g. the weight assigned to position 3 cannot be greater than 3). This evaluation criteria was also used by [1] for the evaluation of Web surfer behavior prediction.

For SVM models, in addition to computing the percent score as described above, the standard classification accuracy is reported for each page category in a navigation path.

5.3 Results and Discussion

5.3.1 Bayesian prediction results

Table 2 shows percent scores for predicting the first three page categories visited for ECML and KDD data, respectively. Prediction results are given for Bayesian models based on no clustering ($k = 1$) and three different clusterings, including when $k = m$. Moreover, prediction results are presented for models based on the top r most relevant clusters (including $r = 1$ and $r = k$) for each user.

Table 2. Bayesian prediction percent scores, k = Total clusters, r = Relevant clusters.

ECML Data					KDD Data				
$r \backslash k$	19	10	5	1	$r \backslash k$	45	20	10	1
1	82.30	82.28	82.14	82.94	1	60.42	61.33	61.51	62.08
3	82.93	82.96	82.99	—	3	61.27	61.65	61.49	—
4	82.97	82.97	82.94	—	4	61.78	61.82	61.93	—
5	82.96	82.96	82.94	—	10	62.07	62.08	62.08	—
k	82.94	82.94	82.94	—	k	62.08	62.08	62.08	—

For ECML data (see Table 2, left), the best prediction is obtained when $k = 5$ and $r = 3$. However, this result is only marginally better than that produced without clustering ($k = 1$). In general, increasing the number of relevant clusters used from $r = 1$ in cluster-based prediction improves performance with results for $r \geq 4$ being equal to or better than that obtained without clustering. This is consistent with the observation that the majority of users spread over four behavior clusters (see Fig. 2, left). Thus, using four most relevant clusters in prediction yields the best results as higher numbers of relevant clusters often contribute to noise.

For KDD data (see Table 2, right), the best predictions are obtained when no clustering is done ($k = 1$) or when a large fraction of relevant clusters ($r \geq 10$) are used for prediction. Thus, a large fraction of data is required to achieve the best performance, and cluster-based prediction is less effective (as compared to that for ECML data). Moreover, as for ECML data, the best result is not significantly better than that obtained when a

fewer number of relevant clusters are used in cluster-based prediction. We experimented with maximum likelihood estimation of the models with and without add-one smoothing without any significant difference in performance.

5.3.2 SVM prediction results

Table 3 shows the SVM prediction results for ECML and KDD data. It gives the percent accuracy for predicting the page category at each position (*i.e.* c_1 , c_2 , and c_3) and the percent score for predicting the entire sequence. For cluster-based predictions, the number of clusters used is 19 and 45 for ECML and KDD data, respectively, and the most relevant cluster is included as an input feature. These results are obtained from implementing the first, second, and third SVM model (with and without clustering) using LibSVM [25] with RBF kernel function. As compared to the best results for Bayesian prediction, SVM prediction percent scores are slightly lower for ECML data and slightly higher for KDD data. For both data sets, however, SVM prediction is not impacted by incorporating the most relevant cluster input feature in the model.

Table 3. SVM prediction percent accuracies and percent scores.

Prediction	SVM	Accuracy(%)			
		Without Clustering		With Clustering	
		ECML	KDD	ECML	KDD
C_1	First	74.47	72.49	74.47	72.49
C_2	Second	46.70	20.84	46.77	20.84
C_3	Third	56.73	20.59	56.76	20.59
Average	—	59.30	37.97	59.33	37.97
Score (%)	—	82.87	62.47	82.88	62.47

5.3.3 Comparison and discussion

Prediction results for ECML data have also been reported in [9, 20]. In this work, we train our models using the first three page categories only. When we train the models on complete navigation paths, our results (with no clustering) are slightly better than those reported in [9]. Our results are slightly lower than those reported in [20], however, our models are simpler and more efficient than the risk minimization and regression based approach developed by [20]. Moreover, none of the two approaches use or investigate behavior clustering for behavior prediction. As seen from the previous section, behavior clustering is not improving behavior prediction significantly or consistently. To verify the statistical significance of behavior clustering on prediction, we perform the paired t -test at 0.05 level of significance. The normality of the data is verified using 1-Sample K-S test. The null hypothesis of the t -test is: prediction means of with and without clustering samples are equal. The results of the paired t -test reveal that at 0.05 level of significance we cannot reject the null hypothesis. Hence, clustering has no significant impact on behavior prediction, as determined for our selected real world data sets. Thus, behavior clustering, which is computationally expensive in itself, should not be performed if the only goal is to obtain better prediction models. However, it remains to be investigated whether this observation still holds when several months or years of historical data are available.

Recently, it has been reported that the practice of generating user navigation sessions using timeouts can break a single logical session into two [26]. This issue negatively affects the performance of any statistical model built from such data. More specifically, it muddles the user behavior at the start of the visit session. In additional experiments, we found out that this issue exists in our experimental data sets. For example, in the ECML data, sessions are terminated when a user remains inactive for 30 minutes. We found out that more than 15% of the sessions in this data are continuations of a previous session of the same user that started between 30 and 60 minutes ago. This discovery provides another reason for the insignificant impact of behavior clustering on the prediction of the first three page categories visited by users.

The above observation suggests that we should focus on position independent behavior prediction or highly probable sequence discovery. These are topics for future investigations and are beyond the scope of the current paper.

5.3.4 Incorporating timestamp in prediction models

We investigate the impact of incorporating time stamp information *i.e.* the start time of the web navigation session (in addition to user ID) in Bayesian and SVM prediction when behavior clustering is not done ($k = 1$). We discretize the timestamp field into four values: weekday-day, weekday-night, weekend-day, and weekend-night. The time period between 8 AM and 6 PM is considered as daytime.

The prediction results for both Bayesian and SVM models are not significant, using paired t test at 0.05 level of significance. From a practical point of view, incorporating timestamp information provides no benefit for both of these data sets. Moreover, the efficiency of both models degrades with the incorporation of time stamp information.

5.3.5 Prediction by most relevant cluster mode

The first three page categories visited by a user can be predicted by the mode of the most relevant behavior cluster of the user, Table 4. Once behavior clustering is done, we use the mode of the user's most relevant cluster for prediction. For the ECML data, clustering segments the users according to their surfing behaviors as reflected by the increase in prediction scores with the increase in k . For the KDD data, the best result is obtained when no clustering is done ($k = 1$). This data set is smaller and less varied, and thus the mode of the entire data is a good representative of all users. These results demonstrate the trade-off between prediction performance and efficiency whereby significant improvement in efficiency can be achieved by a slight reduction in prediction performance.

Table 4. Prediction percent scores obtained when using the most relevant cluster mode.

ECML	$k = 19$	$k = 10$	$k = 5$	$k = 1$	KDD	$k = 45$	$k = 20$	$k = 10$	$k = 1$
Score (%)	79.40	75.92	61.35	42.94	Score (%)	58.18	59.71	59.71	61.13

6. CONCLUSION

In this paper, we present and evaluate several models for Web surfer behavior prediction studying the impact of behavior clustering on prediction and the effectiveness efficiency trade-off of various settings. We consider the prediction of the first three page

categories visited in a Web navigation path. The history of navigation paths is partitioned using a k -modes clustering algorithm appropriately modified for the problem of sequence clustering. In particular, we use a new similarity measure called *MatchScore* for determining the similarity between two navigation paths and a new approach for selecting initial cluster representatives. Predictions are made by Bayesian and SVM models built on clustered and un-clustered data, and by most relevant cluster mode. We evaluate our models on two real world Web surfing data sets.

The following conclusions can be drawn from this evaluation: (1) Behavior clustering can improve surfing prediction using both Bayesian and SVM models. This improvement depends upon the distribution of users across behavior clusters. For our evaluation data sets, which are fair representatives, the improvement in prediction performance is minimal. This can be attributable to the relatively smaller sizes of the data sets; (2) The computational complexity of prediction increases with the number of clusters; (3) The simple prediction approach of recommending the most relevant cluster mode can produce accurate predictions at significantly improved efficiency; (4) Incorporating time stamp information with user ID in the models does not produce significant improvement in prediction; (5) If large quantities of historical data are not available then simpler prediction models or simple predictions based on behavior clustering are sufficiently accurate for practical purposes. In the future, we plan to extend our investigation to more and larger data sets, and focus on improved hybrid clustering-prediction algorithms for the Web surfer behavior prediction problem.

REFERENCES

1. E. H. S. Nguyen, "Ecml/pkdd: Discovery challenge," in *Proceedings of ECML/PKDD: Discovery Challenge*, <http://www.ecmlpkdd2007.org/challenge>, 2007.
2. A. Jain, M. Murty, and P. Flynn, "Data clustering: A review," *ACM Computing Surveys*, Vol. 31, 1999, pp. 264-323.
3. Y. Kim, "Weighted order-dependent clustering and visualization of web navigation patterns," *Decision Support Systems*, Vol. 43, 2007, pp. 1630-1645.
4. 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*, Vol. 7, 2003, pp. 399-424.
5. Z. Huang, "Extensions to the k -means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery*, Vol. 2, 1998, pp. 283-304.
6. J. Borges and M. Levene, "Data mining of user navigation patterns," in *Proceedings of International Workshop on Web Usage Analysis and User Profiling*, 2000, pp. 92-111.
7. M. Deshpande and G. Karypis, "Selective markov models for predicting web page accesses," *ACM Transactions on Internet Technology*, Vol. 4, 2004, pp. 163-184.
8. M. Awad, L. Khan, and B. Thuraisingham, "Predicting www surfing using multiple evidence combination," *The VLDB Journal*, Vol. 17, 2008, pp. 401-417.
9. M. T. Hassan, K. N. Junejo, and A. Karim, "Bayesian inference for web surfer behavior prediction," in *Proceedings of ECML/PKDD: Discovery Challenge*, 2007, pp. 21-29.
10. M. T. Hassan, K. N. Junejo, and A. Karim, "Learning and predicting key web navi-

- gation patterns using Bayesian models,” in *Proceedings of International Conference on Computational Science and Its Applications*, 2009, pp. 877-887.
11. L. Lu, M. Dunham, and Y. Meng, “Discovery of significant usage patterns from clusters of clickstream data,” in *Proceedings of Webkdd Workshop Notes*, 2005, pp. 68-79.
 12. M. Halvey, M. T. Keane, and B. Smyth, “Birds of a feather surf together: Using clustering methods to improve navigation prediction from internet log files,” *Lecture Notes in Computer Science*, Vol. 3587, 2005, pp. 174-183.
 13. G. Pallis, L. Angelis, and A. Vakali, “Validation and interpretation of web users’ sessions clusters,” *Information Processing and Management*, Vol. 43, 2007, pp. 1348-1367.
 14. S. Park, N. Suresh, and B. Jeong, “Sequence-based clustering for web usage mining: A new experimental framework and ANN-enhanced K -means algorithm,” *Data and Knowledge Engineering*, Vol. 65, 2008, pp. 512-543.
 15. B. Niu and S. Shiu, “Using similarity measure to enhance the robustness of web access prediction model,” *Lecture Notes in Computer Science*, Vol. 3683, 2005, pp. 107-111.
 16. H. Liu and V. Keselj, “Combined mining of web server logs and web contents for classifying user navigation patterns and predicting users’ future requests,” *Data and Knowledge Engineering*, Vol. 61, 2007, pp. 304-330.
 17. H. Li and S. Lee, “Mining top- k path traversal patterns over streaming web click sequences,” *Journal of Information Science and Engineering*, Vol. 25, 2009, pp. 1121-1133.
 18. A. Pitman, M. Zanker, M. Fuchs, and M. Lexhagen, “Web usage mining in tourism – A query term analysis and clustering approach,” in *Proceedings of International Conference on Information and Communication Technologies in Tourism*, 2010, pp. 393-403.
 19. D. Pelleg and A. W. Moore, “X-means: Extending k -means with efficient estimation of the number of clusters,” in *Proceedings of the 17th International Conference on Machine Learning*, 2000, pp. 727-734.
 20. K. Dembczynski, W. Kotlowski, and M. Sydow, “Effective prediction of web user behaviour with user-level models,” *Fundamenta Informaticae*, Vol. 89, 2008, pp. 189-206.
 21. J. M. Pena, J. A. Lozano, and P. Larranaga, “An empirical comparison of four initialization methods for the k -means algorithm,” *Pattern Recognition Letters*, Vol. 20, 1999, pp. 1027-1040.
 22. P. S. Bradley and U. M. Fayyad, “Refining initial points for k -means clustering,” in *Proceedings of the 15th International Conference on Machine Learning*, 1998, pp. 91-99.
 23. M. Kendall and J. D. Gibbons, *Rank Correlation Methods*, A Charles Griffin Title, 5th ed., Oxford University Press, New York, 1990.
 24. R. Kohavi, C. Brodley, B. Frasca, L. Mason, and Z. Zheng, “KDD-Cup 2000 organizers’ report: Peeling the onion,” *SIGKDD Explorations*, Vol. 2, 2000, pp. 86-98.
 25. Y. El-Manzalawy and V. Honavar, “WLSVM: Integrating LibSVM into Weka environment,” *Software*, 2005, <http://www.cs.iastate.edu/yasser/wlsvm>.
 26. M. Meiss, J. Duncan, B. Goncalves, J. Ramasco, and F. Menczer, “What’s in a session:

tracking individual behavior on the web,” in *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, 2009, pp. 173-182.



Malik Tahir Hassan received his B.Sc. degree in Math. and Stats. from Government College Lahore, Pakistan, in 1999, and his M.Sc. Computer Science degree, in 2002, from Quaid-i-Azam University, Islamabad, Pakistan. He completed his M.S. Computer Science in 2006 from Lahore University of Management Sciences (LUMS) and is currently a Ph.D. candidate at LUMS. His research interests include web usage mining, data mining and machine learning.



Asim Karim received the B.Sc. (Honors) degree in Civil Engineering from UET, Lahore, Pakistan, in 1994, the M.S. degree in Structural Engineering from The Ohio State University, USA, in 1996, and the Ph.D. degree in Computer and Information Science and Systems Engineering in 2002. He is currently working as Associate Professor in the Department of Computer Science, LUMS, Pakistan. His research interests include spam filtering, text categorization, generative discriminative methods, and social computing applications.