Probabilistic User Behavior Models

Eren Manavoglu
Pennsylvania State University
001 Thomas Building
University Park, PA 16802
manavogl@cse.psu.edu

Dmitry Pavlov*
Yahoo Inc.
701 First Avenue
Sunnyvale, California 94089
dpavlov@yahoo-inc.com

C. Lee Giles
Pennsylvania State University
001 Thomas Building
University Park, PA 16802
giles@ist.psu.edu

Abstract

We present a mixture model based approach for learning individualized behavior models for the Web users. We investigate the use of maximum entropy and Markov mixture models for generating probabilistic behavior models. We first build a global behavior model for the entire population and then personalize this global model for the existing users by assigning each user individual component weights for the mixture model. We then use these individual weights to group the users into behavior model clusters. We show that the clusters generated in this manner are interpretable and able to represent dominant behavior patterns. We conduct offline experiments on around two months worth of data from CiteSeer, an online digital library for computer science research papers currently storing more than 470,000 documents. We show that both maximum entropy and Markov based personal user behavior models are strong predictive models. We also show that maximum entropy based mixture model outperforms Markov mixture models in recognizing complex user behavior patterns.

1. Introduction and Related Work

Whether the underlying reason is to detect fraud or malicious visitors, to improve the organization of a Web site to better serve customers or to identify hidden patterns and new trends in consumer behavior for improving profit, massive amounts of Web data are being collected and stored everyday. Understanding user behavior and discovering the valuable information within such huge databases involves several phases: *data cleaning and preprocessing*, where typically noise is removed, log files are broken into sessions and users are identified; *data transformation*, where useful features are selected to represent the data and/or dimension reduction techniques are used to reduce the size of the data; *applying data mining techniques* to identify interesting patterns, statistical or predictive models or correlations among parts of data; *interpretation of the results*, which includes visualization of the discovered knowledge and transforming them into user friendly formats.

The focus of this paper is the data mining and interpretation phases of this process. We investigate the use of maximum entropy mixture models and mixture of Markov models for inferring individualized behavior models of Web users, where a behavior model is a probabilistic model describing which actions the user will perform in the future. Mixture models also provide a means to cluster the data. The interpretation of clusters obtained in our experiments allows us to conclude that maximum entropy and Markov mixture models have both descriptive and predictive power.

A variety of data mining techniques have been used for the purpose of Web data analysis. Association rule extraction, collaborative filtering, clustering, classification, dependency modeling, and sequential pattern analysis are the most common and noticeable of these methods. Association rule extraction has been used to identify sets of items that are accessed together [15]. Collaborative filtering algorithms [21, 18] have been used to first find similar users based on the overlap between their requested items, and then recommend the given user items accessed by the like-minded users. Clustering, in the context of Web data, can either be used to group together similar items or users with similar usage patterns [2]. Probabilistic graphical models are used to discover and represent dependencies among different variables such as, for instance, the effect of gender on the shopping behavior. Dependency [12] and Bayesian networks [11] are examples of such techniques. Sequential pattern analysis algorithms use time-ordered sessions or episodes and attempt to discover patterns such that the current history of items/actions is evidence to the following item/action.

One of the most motivating reasons for Web usage analysis is its potential to provide customized services. Successful applications of personalization based on Web usage

*Work done at NEC Laboratories America.

Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)
0-7695-1978-4/03 $ 17.00 © 2003 IEEE

mining include adaptive Web-sites, where the structure of the Web-site is optimized for each individual's taste [19]; extracting usage patterns for deriving intelligent marketing strategies [5, 1]; personalized recommendations [16] and individualized predictive profile generation [3].

In this paper, we use personalized probabilistic sequential models to represent user behavior. User behavior can be viewed as a probabilistic model $P(A^{next}|H(U))$, where A^{next} is the next action taken by the user U, $H(U)$ is the action history for the user U in the present session, and P can be any probabilistic function. In our previous work on sequence modeling [17] and recommender systems [6] we explored mixture of maximum entropy (maxent) and Markov models in the context of sequential analysis problems. Here, we use mixture models to capture the diversity in individual behaviors. Each component of a mixture model represents a dominant pattern in the data and each sequence (user sessions in our case) is modeled as a weighted combination of these components. By grouping each session into the highest weighted component, we are also able to cluster the user sessions. Personalization is achieved by optimizing the weights for each individual user, as suggested by Cadez et al [3]. We are able to eliminate one of the biggest problems of personalization, the lack of sufficient information about each individual, by starting with a global model and optimizing the weights for each individual with respect to the amount of data we have about him/her.

We use web-server logs of CiteSeer (a.k.a. ResearchIndex) [1], an online digital library of computer science papers, as our test bed. The site automatically locates computer science papers found on the Web, indexes their full text, allows browsing via the literature citation graph, and isolates the text around citations, among other services [14]. The archive contains over 470,000 documents including the full text of each document, citation links between documents and receives thousands of user accesses per hour. Users of CiteSeer can search both the documents and citations, view and download documents, follow the recommendations, upload documents or correct document information.

We show how the mixture model can be learned directly from the available data. Although maxent learning has high computational cost, the dimension of the action space is inside the limits of feasible computation.

The contributions of this paper can be summarized as:

1. proposed to represent the user behavior as a sequential model;
2. introduced a maxent-based mixture model framework for user behavior modeling;
3. adapted a personalized mechanism which overcomes the insufficient data problem by individual optimization proportional to the amount of data available for each user;
4. evaluated the proposed models and showed that personalization outperforms global models and mixture of maxent models are able to capture complex patterns in user behavior.

The rest of the paper is organized as follows. In Section 2 we give a definition of the problem and describe the general notation. We introduce our model in Section 3. Section 4 describes our visualization method. We give an overview of our data set and preprocessing steps in Section 5. Experimental results and comparisons are given in Section 6. In Section 7 we present our conclusions and ponder future work.

2. General Notation and Problem Definition

We assume that we are given a data set consisting of ordered sequences in some alphabet and that each sequence is labeled with a user id U. For the purposes of this paper we refer to individual items in the alphabet as actions and each sequence represents a user session.

For each action in a user session, the history $H(U)$ is defined as the so-far observed ordered sequence of actions. Our behavior model for individual U is a model, e.g. maxent or Markov, that predicts the next action A^{next} given the history $H(U)$. Therefore the problem is to infer this model, $P(A^{next}|H(U), Data)$, for each individual given the training data.

A serious drawback of personalization algorithms for the Web domain is the insufficient data problem. For many transaction data sets most user ids are seen only in one or two sessions, which makes it impossible to learn reliable predictive profiles for those users. If the Web site does not require registration and the user ids are set with temporary cookies, the situation gets even worse. Log files will have lots of users with only a few sessions, most of which won't be seen in the future transactions at all and most of the users seen in run-time will be new users, unknown to the system.

This is the primary reason why a straightforward approach to personalization, that consists of learning the model for each user only from that user's past transactions, fails for the personalization task with the Web data. Specifically, even after being learned on a wealth of training data for a user, the system could suffer from over-fitting and "cold-start" problem for new visitors the Web site.

The approach that we advocate is to use a global mixture model to capture specific patterns of general behavior of the users, and once the global model is learned, we optimize the weight of each component for each known user individually, hence combining the global patterns with individual irregularities.

[1] http://www.researchindex.com

3. Mixture of Maxent and Markov Models

In this section we describe the global and individualized maxent and Markov mixture models.

3.1. Global Mixture Models

The use of mixture models to represent the behavior of an individual can be viewed as assuming that the ordered sequence of actions of a visitor U at the Web site, is assigned to cluster k with a probability α_k, $(k = 1, \ldots, N_c)$, and each cluster assigns a probability to the sequence via a distribution specific to that cluster. The formal definition of a N_c-component mixture model is as follows:

$$P(A^{next}|H(U), Data) \quad = \quad \sum_{k=1}^{N_c} \alpha_k P(A^{next}|H(U), Data, k)$$

where $\sum_{k=1}^{N_c} \alpha_k = 1$. α_k is the prior probability of cluster k, and $P(A^{next}|H(U), Data, k)$ is the distribution for the k-th component. For the global model α_k's take the same values across all the users. Based on the results of our previous research [17, 6] we decided to use first order Markov model and maxent to model cluster-specific distributions. Both models are explained in the following sections.

3.1.1 Markov Model

In the first order Markov model, the current action depends on the history $H(U)$ only through the last observed action, A^{prev}. The definition of a Markov model for the distribution of the k-th cluster is therefore

$$P(A^{next}|H(U), Data, k) \quad \propto \quad \theta_{0,k} \prod_{h=1}^{|H(U)|} \theta_{h\rightarrow(h+1),k}$$

where $\theta_{0,k}$ is the probability of observing $H(U)_0$ as the first action in the history, and $\theta_{(h\rightarrow h+1),k}$ is the probability of observing a transition from action number h to action number $h+1$ in the history. For $h = |H(U)|$, action with index $h+1$ is A^{next}. The number of parameters is quadratic in the number of actions. Note that the regular Markov model only depends on the so-called *bigrams* or first order Markov terms, i.e. the frequencies of pairs of consecutive actions.

3.1.2 Maximum Entropy Model

It's also possible to model the component distribution $P(A^{next}|H(U), Data, k)$ as a maximum entropy model. Maximum entropy provides a framework to combine information from different knowledge sources. Each knowledge source imposes a set of constraints on the combined model. The intersection of all the constraints contains a set of probability functions, satisfying all the conditions. Maximum entropy principle chooses among these functions the one with the highest information entropy, i.e. the most flat function. We are motivated to use maximum entropy approach in order to combine first order Markov model features with other properties of the data. More specifically, we believe that the most recent action, A^{prev}, has the most influence on the current action taken by the user. However, we also believe that actions other than A^{prev} seen in the history $H(U)$ are also effective. Higher order Markov models may seem to be solving this problem, but it is not feasible to build them for high-dimensional data due to the curse of dimensionality. Furthermore, higher order Markov models use a strict order of the action sequence. Maxent, on the other hand, can be set up with much milder restrictions.

We selected two flavors of low-order statistics or features, as they are typically referred to in the maximum entropy literature, for estimation [13]. Bigrams, or first order Markov terms, were one type. In order to introduce long term dependence of A^{next} on the actions that occurred in the history of the user session, we include triggers, position-specific or non-position-specific, in addition to bigrams. A non position-specific trigger is defined as a pair of actions (a, b) in a given cluster such that $P(A^{next} = b|a \in H(U))$ is substantially different from $P(A^{next} = b)$. If we restrict the action pairs to be exactly $|H(U)|$ actions apart from each other, the resulting trigger would be position-specific. We use both types of triggers in our experiments. To measure the quality of triggers and in order to rank them we computed mutual information between events $E_1 = \{A^{next} = b\}$ and $E_2 = \{a \in H(U)\}$. We then discarded low scoring triggers but retained all bigrams. Note that the quantity and quality of selected triggers depend on the length of $H(U)$. Since the majority of the user sessions is shorter than 5 actions, we chose 5 to be the maximum length of the history.

The set of features, bigrams and triggers in our case, together with maximum entropy as an objective function, can be shown to lead to the following form of the conditional maximum entropy model

$$P(A^{next}|H(U), Data) = \frac{1}{Z_\lambda(H(U))} exp[\sum_{s=1}^{S} \lambda_s F_s(A^{next}, H(U))]$$

where $Z(H(U))$ is a normalization constant ensuring that the distribution sums to 1 and F_s are the features. The set of parameters $\{\lambda\}$ needs to be found from the following set of equations that restrict the distribution $P(A^{next}|H(U), Data)$ to have the same expected value for each feature as seen in the training data:

$$\sum_H \sum_A P(A|H, Data) F_s(A, H) = \sum_{H(U)} F_s(A(H(U)), H(U)), \quad s = 1, \ldots, S$$

where the left hand side represents the expectation (up to a normalization factor) of the feature $F_s(A, H)$ with respect

to the distribution $P(A|H, Data)$ and the right hand side is the expected value (up to the same normalization factor) of this feature in the training data.

There exist efficient algorithms for finding the parameters $\{\lambda\}$ (e.g. generalized [7], improved [20] and sequential conditional [10] iterative scaling algorithms) that are known to converge if the constraints imposed on P are consistent. The pseudocode of the algorithm and a detailed discussion on the ways of speeding it up can be found, for example in [13, 9, 10].

Under fairly general assumptions, maximum entropy model can also be shown to be a maximum likelihood model [20]. Employing a Gaussian prior with a zero mean on parameters λ yields a maximum aposteriori solution that has been shown to be more accurate than the related maximum likelihood solution and other smoothing techniques for maximum entropy models [4]. We use Gaussian smoothing in our experiments for a maxent model.

3.2. Personalized Mixture Model

We personalize the mixture model by using individual cluster probabilities, $\alpha_{U,k}$'s, for each user. The resulting model is therefore specific to each user U:

$$P_U(A^{next}|H(U), Data) = \sum_{k=1}^{N_c} \alpha_{U,k} P(A^{next}|H(U), Data, k)$$

where $\sum_{k=1}^{N_c} \alpha_{U,k} = 1$. The component distribution, $P(A^{next}|H(U), Data, k)$, is the same as in global mixture model: either maximum entropy or Markov model, which is fixed across all users. The N_c component distributions can also be viewed as N_c dimensions of the whole population's behavior space. $\alpha_{U,k}$'s specify where the user U stands in this population. This formulation allows the use of the whole population's experience for each individual's own use, thus avoiding the over-fitting problem. Unknown user problem is resolved naturally as well, by using the global α_k's for new users.

3.3. Parameter Estimation

We assume that the action sequences are drawn independently from a fixed distribution. Thus, the likelihood of the data can be formulated as the product of the individual likelihoods:

$$P(Data|\Theta) = \prod_{k=1}^{N_u} P(Data_U|\Theta)$$

where Θ stands for the full set of parameters of the model and N_u is the number of users.

By the chain rule:

$$P(Data_U|\Theta) = \prod_{s=1}^{N_s} \prod_{j=1}^{N_s a} P(A^j|H(U), \Theta)$$

where N_s is the number of sesions user U has and $N_s a$ is the number of actions taken by user U in session s .

Unknown parameters for the global model include α_k's and λ_k's of the maxent or θ_k's of the Markov model. Parameters can be learned by using the Expectation-Maximization (EM) algorithm as described in [3, 8].

For learning the personalized model, two different approaches can be taken. Our goal is to learn individual $\alpha_{U,k}$'s, therefore we can fix the component distribution model's parameters (i.e. λ_k's of the maxent model or θ_k's of the Markov model) to the values of the global model, and perform the optimization on the $\alpha_{U,k}$'s only. Or we can vary the component distribution's parameters as well. For both cases the optimization is carried out for each user individually, i.e. personal models are trained on each user's data set separately.

If the first approach is taken and the component distribution model parameters are fixed, EM algorithm is run on each individual's own data set to find $\alpha_{U,k}$'s, which are initialized with the global α_k values. If the second approach is chosen instead, EM algorithm is used to learn both $\alpha_{U,k}$'s and component distribution model parameters, which are again initialized with the values learned for the global model. Steps of the parameter estimation process can be summarized as follows:

- Run EM on the whole data set to learn global α_k's and component distribution model parameters;
- Group the sessions by individuals;
- Do either
  - Fix component distribution parameters to the global values and initialize $\alpha_{U,k}$'s with global α_k values;
  - Run EM on the individual data sets to learn $\alpha_{U,k}$'s.

  Or
  - Initialize $\alpha_{U,k}$'s and component distribution parameters with global values;
  - Run EM on the individual data sets to learn all the parameters.

According to this framework, for the new users in the test set, user specific α values will be the initialization values, which are the global α_k's, since there will be no user data to change it.

Notice that even if the component distribution parameters are optimized for the personal model, these values won't be user specific values. Cadez et al. [3] mention that the final values of the parameters of the multinomial model are close to the initial estimates, however, we found that for maxent and Markov models this is not true. Optimizing the λ_k's of the maxent model or θ_k's of the Markov model for the second time causes the model to over-fit the known users' behavior. We recomend using the initial global model

for the unknown users if this approach is taken for parameter estimation. Since the difference between the recommended method and fixed component distribution parameter method is negligible and optimizing the λ_k's for maxent is too time consuming, we chose fixing these parameters to conduct our experiments.

4. Visualization and Interpretation

As mentioned earlier, each component of a mixture model can be viewed as a cluster, representing a certain pattern present in the data. The resulting model represents each session as a weighted combination of these clusters. Given the observed session S_U of user U, the probability distribution over the cluster variable k can be computed by the Bayesian rule:

$$P(k|S_U, Data) = \frac{\alpha_k P_U(S_U|Data, k)}{\sum_{k'=1}^{N_c} \alpha_{k'} P_U(S_U|Data, k')}$$

where

$$P_U(S_U|Data, k) = \prod_{j=1}^{|S_U|} P_U(A^j|H(U), Data, k)$$

Once $P(k|S_U, Data)$'s, which are aldo referred to as membership probabilities, are computed we assign session S_U to the cluster with highest probability. Instead of this $hard$ assignment strategy we could also do $soft$ clustering and assign the session to a set of clusters.

Interpreting the key behaviors exhibited by the users in each cluster is important for a number of tasks, such as managing the site, targeted advertising, identifying malicious visitors. It also helps understanding the navigation patterns of different user groups and therefore helps in organizing the site to better suit the users. Visualizing the users' behavior also makes it possible to identify and provide customized services, like customized help and recommendations.

5. Data Description and Preprocessing

Our data set consists of CiteSeer log files covering a period of approximately two months. The log files are a series of transaction in the form <time, action, user id, action related information>. The complete list of user actions that were available in CiteSeer during the period of our experiments can be found in Table 1. Some of these actions are not being used in CiteSeer anymore.

When a user accesses CiteSeer, a temporary cookie is set on the client side, if a cookie enabled browser is being used. CiteSeer uses this cookie to identify returning users. If no cookie is found, a new user id is given to the user. Each access is recorded on the server side with a unique user id and time stamp.

Table 1. CiteSeer user actions and their descriptions.

Active Bibliography	Active bibliography of a document
Bibtex	Bibtex entry page of the active document
Same Site Documents	The page of documents residing on the same site
Related Documents	Related documents page
Users Who Viewed	Documents viewed by the viewers of active document
Text Related	Page with the list of text based similar documents
Author Homepage	Homepage of the active document's author
Source URL	Original URL of the document
Add Documents	Document upload request
Submit Documents	Document upload submission
Correct Document Title	Request to correct a document's title
Submit Document Title Correction	Title correction submission
Correct Document Abstract	Request to correct a document's abstract
Submit Document Abstract Correction	Abstract correction submission
Check Citations	Citations referring to the active document
Cached Page	Cached page image of the active document
Download	Download a document
Update Cache	Update the cached copy of the active document
Add Comment	Submit comments about the active document
Rate	Rate the active document
Citation Query	Submit a citation query
Document Query	Submit a document query
Document Details	Document's details page
Context	Document's citation context information page
Context Summary	Document's citation context summary page
Homepage	CiteSeer homepage
Help	CiteSeer help page

First step of preprocessing the data is aggregating the transactions by user id and breaking them into sessions. We use time oriented heuristics to recognize new sessions. For a fixed user id, we define a session as a sequence of actions with no two consecutive actions more than 300 seconds apart. If a user is inactive for more than 300 seconds his/her next action is considered as the start of a new session.

Next, we identify robots and discard sessions belonging to them. We examine the histogram of number of accesses in one session to recognize robots. Users who access the archive more than some threshold in one session, are labeled as robots. After removing the robot sessions we collapse the same consecutive actions into a single instance of that action, and discard sessions which contain only one action.

We chronologically partitioned the data into 1,720,512 training sessions and 430,128 test sessions. The total number of actions in the training data is 12,200,965 and in test data this number is 3,853,108. The average number of sessions per user is 7 in the training data and 9 in the test data. The preprocessed data is represented as a collection of ordered sequences of user actions, where each sequence is labeled with a user id. Test data includes 54,429 users out of which only 8139 of the users were seen in the training data also. Since the model proposed in this paper uses the global model for the unknown users, the effects of personalization

won't be seen clearly in the results for all the users. We therefore report the results on the revisiting users, and give the results for the whole data if there are any major differences between the two cases.

6. Experimental Results and Comparisons

We evaluated the user behavior models based on the accuracy of their predictions and visualized the user behavior clusters to demonstrate the descriptive ability of the models. Prediction accuracy is evaluated by scanning the user sessions and for each action in the session predicting the identity of the following action.

In our experiments we compare mixtures of position specific (PS) maximum entropy models, mixtures of non position specific (non-PS) maximum entropy models and mixtures of Markov models. For maximum entropy models, the length of the history was set to 5. Our main criteria for prediction evaluation is the $hit\ ratio$, which is the ratio of the correct predictions to the total number of predictions made. The predictions made by the mixture models are actually lists of ranked actions, where the ranking is done by ordering the actions by their probability values. If the system were to predict only one action, the first action on the ranked list would be chosen. However, the quality of the remaining predictions is also an indication of the success of the model. Therefore we take the first N predictions on the list and evaluate the performance of the models based on the success of these N predictions, for $N = 1, ..., 5, 10$. In this case, a hit occurs if the true action is predicted in any of these N guesses.

We also report the likelihoods of the models on the test data, since it's our optimization criteria and is another indication of how well the model represents the data.

Table 2. Hit ratio results on known users for 3 component mixture model.

	Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	Bin 10
Global Markov	0.5849	0.7826	0.8502	0.8982	0.9229	0.9816
Personal Markov	0.5872	0.7858	**0.8578**	**0.9061**	**0.9291**	**0.9825**
Global PS Maxent	0.6127	0.7863	0.8388	0.8820	0.9081	0.9794
Personal PS Maxent	0.6153	0.7874	0.8430	0.8862	0.9153	0.9810
Global NonPS Maxent	0.6122	0.7813	0.8337	0.8715	0.9059	0.9787
Personal NonPS Maxent	**0.6154**	**0.7879**	0.8402	0.8765	0.9100	0.9806

In Table 2, we present hit ratio results for $N = 1, ..., 5, 10$ on the known users for 3-component mixture model, and in Table 3 hit ratios for 10-component mixture models are presented. Regardless of the number of components and the length of the prediction list, personalized models outperformed the corresponding global models. PS and non-PS specific maxent models' hit ratios are very close to each other, but non-PS model performed better than the PS in the 10-component mixture model and for $N \leq 2$ in 3-component mixture model. An interesting point about the non-PS maxent model is the altitude of the effect of personalization on it. Although the PS model is better in all test cases for the global models, personalization improves the non-PS model more, such that it's able to beat the PS model.

Table 3. Hit ratio results on known users for 10 component mixture model.

	Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	Bin 10
Global Markov	0.6073	0.7967	0.8639	0.9083	0.9361	0.9842
Personal Markov	**0.6245**	**0.8054**	**0.8824**	**0.9232**	**0.9472**	**0.9867**
Global PS Maxent	0.6139	0.7835	0.8450	0.8856	0.9115	0.9797
Personal PS Maxent	0.6209	0.7953	0.8555	0.8963	0.9247	0.9823
Global NonPS Maxent	0.6113	0.7829	0.8393	0.8785	0.9097	0.9799
Personal NonPS Maxent	0.6226	0.7970	0.8556	0.8941	0.9256	0.9834

As follows from the tables, personalized Markov mixture model has the highest hit ratio for the known users. However non-PS maxent was able to perform better for $N \leq 2$ in the 3-component model. This result may seem surprising considering the fact that first order Markov models are making use of only bigrams, whereas maxent models are using triggers in addition to bigrams, but it's not. The goal of maximum entropy is to choose the most general model within the set of functions satisfying the constraints. Markov models, on the other hand, do not have this property, and thus may fit the training data better. The advantage of maxent models can be seen more clearly when looked at the results for all users. Table 4 and Table 5 present the hit ratios of the personal models for 3-component and 10-component mixture models, respectively. Non-PS maxent outperforms Markov model for all prediction list lengths, but 4 in 3-component mixture model, and it performs worse only for $N = 3, 4, 5$ in the 10-component mixture model.

In Table 6, we report the likelihood of the personalized models for the test data. Best likelihood is achieved by Markov mixture model and non-PS maxent mixture follows it. PS maxent mixture performs even worse when the number of components is increased.

As discussed in Section 4 we are also interested in the interpretation of the user behavior clusters. Each user session is grouped into the cluster for which it has the highest $\alpha_{U,k}$

Table 4. Hit ratio results of personalized mixture models on all users for 3 components.

	Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	Bin 10
Personal Markov	0.5699	0.7372	0.8014	**0.8714**	0.8941	0.9525
Personal PS Maxent	0.5872	0.7542	0.8095	0.8506	0.8773	0.9357
Personal Non-PS Maxent	**0.5948**	**0.7615**	**0.8126**	0.8636	**0.8943**	**0.9557**

Table 5. Hit ratio results of personalized mixture models on all users for 10 components.

	Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	Bin 10
Personal Markov	0.5830	0.7680	**0.8520**	**0.8891**	**0.9132**	0.9567
Personal PS Maxent	0.5865	0.7492	0.8208	0.8595	0.8857	0.9416
Personal Non-PS Maxent	**0.6081**	**0.7713**	0.8336	0.8679	0.8982	**0.9614**

value. For cluster visualization we chose 100 sample user sessions randomly from each cluster. Each unique action is represented by a unique color (action-color mapping is also shown in the figure). Hence, each user session is represented as a row of colored squares, where each squares corresponds to an action.

Figure 1. User Clusters Generated by the 10-Component Markov Mixture Model.

This visualization technique has enabled us to actually identify different behavior models among CiteSeer users. Users identified as belonging to Cluster 8 by the Markov model (Figure Figure 1), for example, go to CiteSeer homepage, submit citation queries, view document details and

Table 6. Test data likelihoods for the personalized models.

	PS Maxent	Non-PS Maxent	Markov Model
3 Component	-2.10191	-2.03604	-2.04539
10 Component	-2.19488	-2.00861	-1.93454

Figure 2. User Clusters Generated by the 10-Component Non-PS Maximum Entropy Mixture Model.

context or download the document. Cluster 9 users, on the other hand, view details of a document and download, with hardly taking any other actions. The interesting point about Cluster 9 is that these users go to the details of a document directly, without submitting a query. This is probably an indication of browsing CiteSeer via another search engine. Following Figure 1, it's also clearly seen that Cluster 6 represents the users who after viewing the context or details of a document try to correct the title and then download it. Maximum entropy model (Figure 2) is able to capture the mentioned behavior models, as well as more complex ones. Cluster 4 of maxent model represents users who probably browse CiteSeer through another engine. At first sight Cluster 6 may seem to be presenting the same pattern, however there's a huge difference between the two. Users of Cluster 6 do submit a document query before the document details - download cycle, suggesting that after viewing document details or downloading they go back to the query results page to browse the rest of the results. Although some session in Cluster 1 of Markov model show a similar pattern, it's not as clear. Maximum entropy model was also able to identify a cluster of users, Cluster 3, who check the recommendations after viewing the document information. These users also happen to correct document abstracts or titles.

Overall, we conclude that personalized mixture of maximum entropy and Markov models provide a decent pre-

dictive model for representing user behaviors, and a useful mechanism for identifying and interpreting user behavior patterns for the Web data.

7. Conclusions and Future Work

We described a mixture model based approach to generating and visualizing individual behavior models for CiteSeer users. We represented the Web data as a collection of ordered action sequences for each user. We introduced a maximum entropy based approach for modeling the user behavior, motivated by its ability to model long term dependencies in data sequences. In addition to maxent model, we also investigated the use of first order Markov mixture models. We demonstrated that both methods are able to generate strong predictive models with different strengths and weaknesses. Markov model performed better for predicting the behavior of the known users, whereas maximum entropy model was better at modeling the global behavior model, and therefore the unknown users also. We used a simple method to achieve personalization, yet managed to avoid the insufficient data problem of traditional personalization techniques. By using mixture model based clustering we were able to identify and visualize specific behavior patterns of CiteSeer users, where it was demonstrated that maximum entropy model's computational cost pays off at recognizing complex dominant patterns of user behavior.

We plan to expand our work on identifying specific user behavior patterns and provide customized services, for instance customized recommendations, for each of the behavior model groups. We are also interested in naming these groups of users. We intend to perform real-time experiments on CiteSeer with our maximum entropy based predictive model. We are also planning to apply our personalization algorithm to mixtures of hidden Markov models and compare it with the maxent model proposed in this paper.

8. Acknowledgements

This work has been partially supported by a grant from Lockheed Martin. We would like to thank Steve Lawrence for making the CiteSeer log data available to us.

References

[1] A. G. Buchner and M. D. Mulvenna. Discovering internet marketing intelligence through online analytical web usage mining. *SIGMOD Record*, 27(4):54–61, 1998.

[2] I. V. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Visualization of navigation patterns on a web site using model-based clustering. In *Knowledge Discovery and Data Mining*, pages 280–284, 2000.

[3] I. V. Cadez, P. Smyth, E. Ip, and H. Mannila. Predictive profiles for transaction data using finite mixture models. Technical Report UCI-ICS 01-67, UC Irvine, 2001.

[4] S. Chen and R. Rosenfeld. A gaussian prior for smoothing maximum entropy models. Technical Report CMUCS -99-108, Carnegie Mellon University, 1999.

[5] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.

[6] D. P. D. Pavlov, E. Manavoglu and C. L. Giles. Collaborative filtering with maximum entropy. Technical Report 2003-L001, NEC Labs, 2003.

[7] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43:1470–1480, 1972.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B-39:1–38, 1977.

[9] J. Goodman. Classes for fast maximum entropy training. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[10] J. Goodman. Sequential conditional generalized iterative scaling. In *Proceedings of ACL*, 2002.

[11] D. Heckerman. Bayesian networks for data mining. *Data Mining and Knowledge Discovery*, 1(1):79–119, 1997.

[12] D. Heckerman, D. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for density estimation, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49—75, 2000.

[13] F. Jelinek. *Statistical Methods for Speech Recognition*. Cambridge. MA:MIT Press, 1998.

[14] S. Lawrence, C. L. Giles, and K. Bollacker. Digital libraries and Autonomous Citation Indexing. *IEEE Computer*, 32(6):67–71, 1999.

[15] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 80–96, 1998.

[16] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on Web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.

[17] D. Pavlov. Sequence modeling with mixtures of conditional maximum entropy distributions. In *Proceedings of the Third IEEE Conference on Data Mining (ICDM'03)*, 2003.

[18] D. Pavlov and D. Pennock. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. In *Proceedings of Neural Information Processing Systems*, 2002.

[19] M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically synthesizing web pages. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 727—732, 1998.

[20] S. D. Pietra, V. D. Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, April 1997.

[21] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.