Task-Oriented Web User Modeling for Recommendation

Xin Jin, Yanzan Zhou, Bamshad Mobasher
{xjin,yzhou,mobasher}@cs.depaul.edu

Center for Web Intelligence
School of Computer Science, Telecommunication, and Information Systems
DePaul University, Chicago, Illinois, USA

Abstract. We propose an approach for modeling the navigational behavior of Web users based on task-level patterns. The discovered "tasks" are characterized probabilistically as latent variables, and represent the underlying interests or intended navigational goal of users. The ability to measure the probabilities by which pages in user sessions are associated with various tasks, allow us to track task transitions and modality shifts within (or across) user sessions, and to generate task-level navigational patterns. We also propose a maximum entropy recommendation system which combines the page-level statistics about users' navigational activities together with our task-level usage patterns. Our experiments show that the task-level patterns provide better interpretability of Web users' navigation, and improve the accuracy of recommendations.

1 Introduction

Web users exhibit different types of behavior according to their interests and intended tasks. These interests and tasks are captured implicitly by a collection of actions taken by users during their visits to a site. For example, in a dynamic application-based e-commerce Web site, they may be reflected by sequences of interactions with Web applications to search a catalog, to make a purchase, or to complete an online application. On the other hand, in an information intensive site, such as a portal or an online news site, they may be reflected in a series of user clicks on a collection of Web pages with related content.

Web usage mining is a collection of techniques for modeling Web users' navigational behavior. Beginning from Web server log data, Web usage mining involves the application of data mining and machine learning techniques, such as association rule mining, clustering, or sequential pattern analysis to identify usage patterns. Web usage mining has achieved great success in various application areas such as Web personalization [13, 14], link prediction and analysis [11, 20], Web site evaluation or reorganization [21, 22], Web analytics and e-commerce data analysis [10], Adaptive Web sites [17, 12].

Since users' activities are recorded in terms of visited pages at the session level, after the data mining process, the discovered *page-level patterns* only provide limited interpretability in terms of the underlying users interests and the

intended tasks. For example, we may find an association rule $A \rightarrow B$ involving pages A and B. This page-level pattern, by itself, provides no clue as to nature of association between A and B or the reasons why they are commonly visited together. However, if we are able to view user interactions at a "higher" level of abstraction, we may be able to explain the association in terms of a common task or content category involving these pages. The ability to better explain or interpret the patterns at the higher abstraction level, in turn, can lead to more accurate predictive models such as those used in personalization and recommender systems.

Some recent work has focused on methods to model users' behavior at "higher" abstraction levels. In [16], Web items (documents) are first clustered based on users' navigational data, and then user behavior models are built at this item-cluster level. In [4], an aggregated representation is created as a set of pseudo objects which characterize groups of similar users at the object attribute level. While these approaches have proved useful in some applications, they generally rely on manually pre-defined semantic features or ad hoc clustering methods which limit their flexibility.

In this paper, we propose a new Web user modeling approach called *task-oriented user modeling*. The basic assumption we make is that, in a Web site there exists a relatively small set of common navigational "tasks" which can explain the behavior of numerous individual Web users at different points in their interactions with the site. We use the Probabilistic Latent Semantic Analysis (PLSA) model [6] to probabilistically characterize these tasks, as well as the relationships between the tasks and Web pages or users. We then propose an algorithm based on Bayesian updating to discover individual user's underlying tasks, as well as the temporal changes in these tasks, thus generating *task-level usage patterns*. These task-level patterns enable us to better understand users' underlying navigational goals or interests. We also propose a Web recommendation system based on a maximum entropy model [18, 1], which can flexibly combine knowledge about Web users' navigational behavior by integrating the standard page-level and our proposed task-oriented models.

This paper is organized as follows. We present our task-oriented user model in Section 2. In Section 3, we introduce our recommendation system which takes into account both the page-level and task-level usage patterns. We evaluate this recommendation system and report our experimental results on two real data sets in Section 4. Finally, in Section 5, we identify avenues for future work and conclude this paper.

2 Task-oriented User Modeling with PLSA

In [9], we introduced a general and flexible approach for Web usage mining based on Probabilistic Latent Semantic Analysis (PLSA). Here we employ the PLSA framework to quantitatively characterize users' navigational tasks, as well as the relationships between these tasks and Web pages or users.

The first step in Web usage analysis is the data preparation process. Raw Web log data is cleaned and sessionized to generate user sessions. Each user session is a logical representation of a user's single visit to the Web site (usually within certain time interval). After data preparation (see [3] for details), we have a set of user sessions $U = \{u_1, u_2, \cdots, u_m\}$, a set of pages $P = \{p_1, p_2, \cdots, p_n\}$. The Web session data can be conceptually viewed as a $m \times n$ session-page matrix $UP = [w(u_i, p_j)]_{m \times n}$, where $w(u_i, p_j)$ represents the weight of page p_j in user session u_i. The weights can be binary, indicating the existence or non-existence of the page in the session, or they may be a function of the occurrence or duration of the page in that session. We use a set of hidden (unobserved) variables $Z = \{z_1, z_2, ..., z_l\}$, which in our framework, correspond to users' common tasks. Our goal is to automatically discover and characterize these tasks, and then obtain a view of the users' behavior in the site by associating their actions with the discovered tasks.

Each usage observation, which corresponds to an access by a user to a Web resource in a particular session which is represented as an entry of the $m \times n$ co-occurrence matrix UP, can be modeled as

$$Pr(u_i, p_j) = \sum_{k=1}^{l} Pr(z_k) \bullet Pr(u_i|z_k) \bullet Pr(p_j|z_k), \tag{1}$$

summing over all possible choices of z_k from which the observation could have been generated.

Now, in order to explain a set of usage observations (U, P), we need to estimate the parameters $Pr(z_k)$, $Pr(u_i|z_k)$, $Pr(p_j|z_k)$, while maximizing the log-likelihood of the observations.

$$L(U, P) = \sum_{i=1}^{m} \sum_{j=1}^{n} w(u_i, p_j) \log Pr(u_i, p_j). \tag{2}$$

We use the Expectation-Maximization (EM) algorithm to perform maximum likelihood parameter estimation of $Pr(z_k)$, $Pr(u_i|z_k)$, $Pr(p_j|z_k)$. These probabilities quantitatively measure the relationships among users, Web pages, and common interests (tasks). For example, $Pr(p_j|z_k)$ represents the probability of page p_j being visited given a certain task z_k is pursued. While $Pr(u_i|z_k)$ measures the probabilities of a user engaging in a certain task.

Given the estimated probabilities from the PLSA model, our next goal is to identify user's tasks from individual user sessions. Taking into account the order of pages being visited in user sessions, each user session can also be represented as a sequence of pages, as $u_i = \langle p_i^1, p_i^2, \cdots, p_i^t \rangle$, where $p_i^j \in P$. Here we use a Bayesian updating method, to compute the posterior probability of each task within each individual user session, given the assumption that pages are independent given a task. These probabilities measure the relative importance of each task within the session. Also, we use a sliding window over the user's click-stream history reflecting our assumption that the most recently visited pages are better predictors of user's current interests. The algorithm is as follow.

Input: a user session u_i, $Pr(p_j|z_k)$. Each user session is represented as a page sequence, $u_i = \langle p_i^1, p_i^2, \cdots, p_i^t \rangle$, $p_i^j \in P$.

Output: a user session represented as a task sequence.

1. Get the first L pages from u_i and put them into a sliding window W.
2. Using Bayesian updating [19], we compute the probability of each task given all the pages in W:

$$Pr(z|W) = \frac{Pr(W|z)Pr(z)}{Pr(W)} \propto Pr(z) \prod_{p \in W} Pr(p|z) \tag{3}$$

3. Pick those tasks with probability exceeding a pre-defined threshold as the current tasks and output them.
4. Move the sliding window to the right (remove the first page from the window, and add the next page in u_i into the window), recompute the probability of each task given the current sliding window, and pick the dominant tasks.
5. Repeat this process until the sliding window reaches the end of this session.

After running this algorithm, each user session can be represented as a sequence of tasks, $u_i = \langle z'_x, \cdots, z'_y \rangle$, where z' is either a single task or a small set of dominant tasks. This representation gives us a direct understanding of users' interests and the temporal changes in these interests.

Given these task sequences, we can discover task-level patterns from these task sequences. For example, we can identify the most popular task(s) and the least popular task(s), or we can run simple first-order Markov model to compute probabilities of task transitions, $Pr(z_b|z_a)$, etc. These task-level patterns demonstrate the Web site's functionalities from the users' point of view, and may help the site designer to evaluate and reorganize the site. Another direct application is to generate recommendations based on users' tasks and task transitions. In the next section, we will present a flexible and accurate recommendation system which combines both the page-level and task-level patterns.

3 Recommendations Based on Maximum Entropy

In this section, we present our maximum entropy based recommendation system. The system consists of two components. The offline component accepts constraints from the training data and estimates the model parameters. The online part reads an active user session and runs the recommendation algorithm to generate recommendations (a set of pages) for the user.

To make predictions or generate recommendations for active users, we aim to compute the conditional probability $Pr(p_d|H(u_i))$ of a page p_d being visited next given a user's recent navigational history $H(u_i)$, here $p_d \in P$, and $H(u_i)$ represents a set of recently visited pages by u_i. Since the most recent activities have greater influence on reflecting a user's present interests, we only use the last several pages to represent a user's history.

3.1 The Maximum Entropy Model

To discover $Pr(p_d|H(u_i))$, we adopt a maximum entropy model, a powerful statistical model, which has been widely applied in many fields, including statistical language learning [18], information retrieval [8] and text mining [15]. The goal of a maximum entropy model is to find a probability distribution which satisfies all the constraints in the observed data while maintaining maximum entropy. One main advantage of using maximum entropy model is that one or more knowledge sources can be integrated at different levels of abstraction in a natural way. Here we extract two levels of statistics about Web users' navigational behavior, and use them as features and constraints to fit our model.

In our model, we use page-level and task-level usage patterns extracted from Web users' navigation data. For each type of patterns, we define features that capture certain statistics of users' navigational behavior.

1. Features Based on Page-level Usage Patterns
   For simplicity, here we adopt the first-order Markov model to generate page transitions. For each page transition $p_a \rightarrow p_b$ where $Pr(p_b|p_a) \neq 0$, we define a feature function as:

$$f_{p_a,p_b}(H(u_i), p_d) = \begin{cases} 1 \text{ if page } p_a \text{ is the last page in } H(u_i), \text{ and } p_b = p_d, \\ 0 \text{ otherwise} \end{cases}$$

2. Features Based on Task-Level Usage Patterns
   Similarly, given a task transition $z_a \rightarrow z_b$, we define a feature as $f_{z_a,z_b}(H(u_i), p_d)$:

$$f_{z_a,z_b}(H(u_i), p_d) = \begin{cases} 1 \text{ if the dominant task of } H(u_i) \text{ is } z_a, \text{ and after moving sliding} \\ \quad \text{window right to include } p_d, \ z_b \text{ will be the dominant task,} \\ 0 \text{ otherwise} \end{cases}$$

   Based on each feature f_s, we represent a constraint as:

$$\sum_{u_i} \sum_{p_d \in P} Pr(p_d|H(u_i)) \bullet f_s(H(u_i), p_d) = \sum_{u_i} f_s(H(u_i), D(H(u_i))) \qquad (4)$$

where $D(H(u_i))$ denotes the page following u_i's history in the training data. Every constraint restricts that the expected value of each feature w.r.t. the model distribution should always equal its observation value in the training data. After we have defined a set of features, $F = \{f_1, f_2, \cdots, f_t\}$, and accordingly, generated constraints for each feature, it's guaranteed that, under all the constraints, a unique distribution exists with maximum entropy [7]. This distribution has the form:

$$Pr(p_d|H(u_i)) = \frac{exp(\sum_s \lambda_s f_s(H(u_i), p_d))}{Z(H(u_i))} \qquad (5)$$

where $Z(H(u_i)) = \sum_{p_d \in P} Pr(p_d|H(u_i))$ is the normalization constant ensuring that the distribution sums to 1, and λs are the parameters needed to be estimated. Thus, each source of knowledge can be represented as constraints (captured by the corresponding features) with associated weights. By using Equation 5, all such constraints are taken into account to make predictions about users' next action given their past navigational activity.

There have been several algorithms which can be applied to estimate the λs. Here we use the Sequential Conditional Generalized Iterative Scaling (SCGIS) [5], which seems to be more efficient especially when the number of outcomes is not very large (in our case, the number of pages in a site). We also apply the smoothing technique suggested by Chan and Rosenfeld [2], which has shown to be able to improve the model by using some prior estimates for model parameters.

3.2 Generating Recommendations

After we have estimated the parameters associated with each feature, we can use Equation 5 to compute the probability of any unvisited page p_i being visited next given certain user's history, and then pick the pages with highest probabilities as recommendations. Given an active user u_a, we compute the conditional probability $Pr(p_i|u_a)$. Then we sort all pages based on the probabilities and pick the top N pages to get a recommendation set. The algorithm is as follows:

Input: active user session u_a, parameters λs estimated from training data.
Output: Top N pages sorted by probability of being visited next given u_a.

1. Consider the last L pages of the active user session as a sliding window (these pages are considered as this user's history), and identify all the task(s) with probability exceeding a pre-defined threshold using Equation 3 in Section 2. (L is the size of the sliding window, also the length of the user history.)
2. For each page p_i that does not appear in the active session, assume it is the next page to be visited, and evaluate all the features based on their definitions (above).
3. Using Equation 5 to compute $Pr(p_i|u_a)$.
4. Sort all the pages in descending order of $Pr(p_i|u_a)$ and pick the top N pages to get a recommendation set.

4 Experimental Evaluation

We used two real Web site data sets to empirically measure the accuracy of our recommendation system. Our accuracy metric is called *Hit Ratio* and is used in the context of top-N recommendation framework: for each user session in the test set, we take the first K pages as a representation of an active session to generate a top-N recommendations. We then compare the recommendations with page $K + 1$ in the test session, with a match being considered a *hit*. We define the Hit Ratio as the total number of hits divided by the total number of

user sessions in the test set. Note that the Hit Ratio increases as the value of N (number of recommendations) increases. Thus, in our experiments, we pay special attention to smaller number recommendations (between 1 and 10) that result in good hit ratios.

The first data set is based on the server log data from the host Computer Science department. The site is highly dynamic, involving numerous online applications, including admissions application, online advising, online registration, and faculty-specific Intranet applications. After data preprocessing, we identify 21,299 user sessions (U) and 692 pageviews (P), with each user session consisting of at least 6 pageviews. The number of tasks is set to 30, and the length of user history is set to 4. This data set is referred to as the "CTI data". Our second data set is based on *Network Chicago* Web servers representing the Chicago Public Television and Radio. A total of 4,987 user sessions and 295 Web pageviews were selected for analysis. Each user session consists of at least 6 pageviews. In contrast to the CTI data, this site is comprised primarily of static pages grouped together based on their associations with specific content areas. In this case, we expect users' navigational behavior reflect their interests in one or more programs represented by the content areas. In the experiment, task number is set to 15 and the length of user history is set to 4. We refer to this data set as the "NC data".

We tried different number of tasks and picked the one achieving the maximum likelihood of the training data. User history length was determined based on the average session length in the training data. Each data set was randomly divided into multiple training and test sets to use with 10-fold validation.

4.1 Examples of Task and Task-level Usage Patterns

We applied the PLSA model to each of the training data sets, and then ran the task-level pattern discovery algorithm to generate task-level usage patterns. An example of a task-level usage pattern from the CTI data, and the associated task transition tracking within a selected user session, is depicted in Figure 1.

The user visited 12 pages in the selected session (the upper table in Figure 1). After we ran the algorithm in Section 2 for this session, we generated a task sequence depicted in the lower table. The sliding window moves from the beginning of the session to the end. At each position, we show the top two tasks with corresponding probabilities (in each row). We can see that the user was mainly performing two tasks in this session: Task 10 and Task 21. The figure also shows the top pages associated with Task 10 and Task 21. Task 10 clearly represents prospective international students looking for admission information, while Task 21 represents the activity of prospective students who are actually engaged in the process of submitting an application. As evident from the task transition model, this user gradually finished Task 10 and moved to Task 21. The changes of the probabilities associated with these tasks clearly reflect the change of the user's interests.

A real user session (page listed in the order of being visited)	**Task 10** (in descending order of Pr(page|task))	**Task 21** (in descending order of Pr(page|task))
1. Admission main page	Department main page	Online application – start
2. Welcome information – Chinese version	Admission requirements	Online application – step1
3. Admission info for international students	Admission main page	Online application – step2
4. Admission - requirements	Admission costs	Online application - finish
5. Admission – mail request	Programs	Online application - submit
6. Admission – orientation info	Online application – step 1	Online application - newstart
7. Admission – F1 visa and I20 info	Admission – Visa & I-20 information	Department main page
8. Online application - start	Admission – international students	Admission requirements
9. Online application – step 1		
10. Online application – step 2		
11. Online application - finish		
12. Department main page		

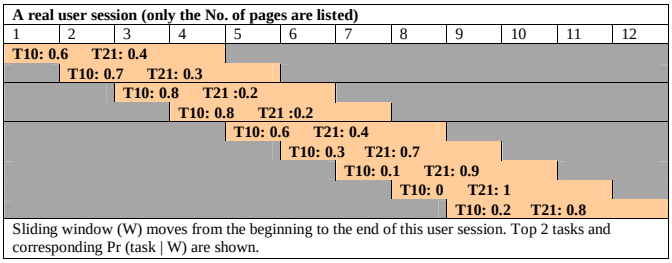

Fig. 1. Task Transition Example from CTI Data

After identifying the tasks and their temporal changes in each user session, we are able to generate recommendations not only based on the page transitions, but also based on users' current task and next possible task.

4.2 Experimental Results

We identified page-level and task-level features as discussed in Section 3 and built our recommendation system according to the algorithm of Section 3. For comparison, we built another recommendation system based on the standard first-order Markov model to predict and recommend which page to visit next. The Markov-based system models each page as a state in a state diagram with each state transition labeled by the conditional probabilities estimated from the actual navigational data from the server log data. It generates recommendations based on the state transition probability of the last page in the active user session.

Figure 2 depicts the comparison of the *Hit Ratio* measures for the two recommender systems in each of the two data sets. The experiments show that the maximum entropy recommendation system has a clear overall advantage in terms of accuracy over the first-order Markov recommendation system on the CTI data, which demonstrates that the incorporation of task-level usage

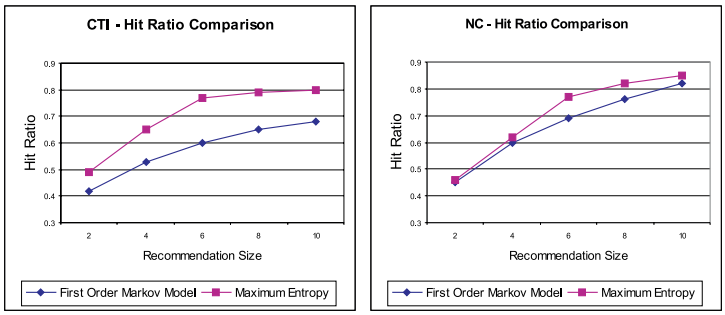

Fig. 2. Recommendation accuracy: maximum entropy model vs. Markov model

patterns actually help achieve better recommendation accuracy. As for the NC data, our model also performs better in general, but the overall advantage is not as apparent as that in the CTI data. One explanation of the difference in the results is that the CTI Web site provides many distinct functionalities for users. Student users can collect admission information, submit online applications, and perform degree audits, while faculty members can do online advising, etc. While the content-based functionalities from NC site are not so distinct as those of CTI site. Therefore, the benefit of using tasks there is not so impressing.

5 Conclusions and Future Work

In this paper, we have presented a framework for modeling Web users' navigational behavior based on the automatically discovered task-level patterns. We have employed a probabilistic approach which can characterize the user tasks as a set of latent variables. We also demonstrated that the task-level patterns can be seamlessly integrated with the standard page-level patterns in Web usage mining to provide the basis for a more flexible and accurate Web recommender system. To achieve this integration, we employed a maximum entropy model which can be used to effectively combine knowledge from multiple sources described as various constraints imposed on the data.

In the future, we plan to conduct work in several directions. We intend to further explore the work on modeling Web users' navigational behavior at the task-level, such as modeling the hierarchical structure of users' interest and tasks. We also intend to incorporate other source of knowledge, such as from the Web site content and linkage into the user modeling process.

References

1. A. Berger, S. Della Pietra, and V. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1), 1996.
2. S. Chen and R. Rosenfeld. A guassian prior for smoothing maximum entropy models. Technical report, CMU, 1999.

3. R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1(1), 1999.
4. H. Dai and B. Mobasher. Using ontologies to discover domain-level web usage profiles. In *Proceedings of the 2nd Semantic Web Mining Workshop at ECML/PKDD 2002*, Helsinki, Finland, August 2002.
5. J. Goodman. Sequential conditional generalized iterative scaling. In *Proceedings of NAACL-2002*, 2002.
6. T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning Journal*, 42(1):177–196, 2001.
7. F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, MA, 1998.
8. J. Jeon and R. Manmatha. Using maximum entropy for automatic image annotation. In *Proceedings of the International Conference on Image and Video Retrieval (CIVR-2004)*, 2004.
9. X. Jin, Y. Zhou, and B. Mobasher. Web usage mining based on probabilistic latent semantic analysis. In *Proceedings of the Tenth ACM SIGKDD Conference(2004)*, 2004.
10. R. Kohavi, L. Mason, R. Parekh, and Z. Zheng. Lessons and challenges from mining retail e-commerce data. *To appear in Machine Learning*, 2004.
11. N. Kushmerick, J. McKee, and F. Toolan. Towards zero-input personalization: Referrer-based page prediction. In P. Brusilovsky, O. Stock, and C. Strapparava, editors, *Proceedings of the Adaptive Hypermedia and Adaptive Web-Based Systems International Conference (AH 2000)*, LNCS 1892, pages 133–143. Springer, 2000.
12. B. Mobasher, R. Cooley, and J. Srivastava. Creating adaptive web sites through usage-based clustering of urls. In *Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99)*, Chicago, Illinois, November 1999.
13. B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.
14. B. Mobasher, H. Dai, and M. Nakagawa T. Luo. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery*, 6:61–82, 2002.
15. K. Nigram, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proceedings of IJCAI-1999*, 1999.
16. 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)*, 2002.
17. M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically synthesizing web pages. In *Proceedings of the 15th National Conference on Artificial Intelligence*, Madison, WI, July 1998.
18. R. Rosenfeld. Adaptive statistical language modeling: A maximum entropy approach. Phd dissertation, CMU, 1994.
19. S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, 2002.
20. R.R. Sarukkai. Link prediction and path analysis using markov chains. In *Proceedings of the 9th International World Wide Web Conference*, Amsterdam, May 2000.
21. M. Spiliopoulou. Web usage mining for web site evaluation. *Communications of the ACM*, 43(8):127–134, 2000.
22. R. Srikant and Y. Yang. Mining web logs to improve website organization. In *Proceedings of the 10th International World Wide Web Conference*, Hong Kong, May 2001.