# A Poisson Model for User Accesses to Web Pages

Şule Gündüz[1] and M. Tamer Özsu[2]

[1] Computer Engineering Department, Istanbul Technical University
Istanbul, Turkey, 34390
gunduz@cs.itu.edu.tr

[2] School of Computer Science, University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
tozsu@db.uwaterloo.ca

**Abstract.** Predicting the next request of a user as she visits Web pages has gained importance as Web-based activity increases. There are a number of different approaches to prediction. This paper concentrates on the discovery and modelling of the user's aggregate interest in a session. This approach relies on the premise that the visiting time of a page is an indicator of the user's interest in that page. Even the same person may have different desires at different times. Although the approach does not use the sequential patterns of transactions, experimental evaluation shows that the approach is quite effective in capturing a Web user's access pattern. The model has an advantage over previous proposals in terms of speed and memory usage.

## 1 Introduction

Web mining is defined as the use of data mining techniques to automatically discover and extract information from Web documents and services [5]. With the rapid growth of the World Wide Web, the study of modelling and predicting a user's access on a Web site has become more important. There are three steps in this process [2]. Since the data source is Web server log data for Web usage mining, the first step is to clean the data and prepare for mining the usage patterns. The second step is to extract usage patterns, and the third step is to build a predictive model based on the extracted usage patterns. The prediction step is the real-time processing of the model, which considers the active user session and makes recommendations based on the discovered patterns.

An important feature of the user's navigation path in a *server session*[3] is the time that a user spends on different pages [12]. If we knew the desire of a user every time she visits the Web site, we could use this information for recommending pages. Unfortunately, experience shows that users are rarely willing

[3] The term *server session* is defined as the click stream of page views for a single visit of a user to a Web site [2]. In this paper we will use this term interchangeably with "user session" and "user transaction".

to give explicit feedback. Thus, the time spent on a page is a good measure of the user's interest in that page, providing an implicit rating for that page. If a user is interested in the content of a page, she will likely spend more time there compared to the other pages in her session. However, the representation of page visit time is important. If the representation is not appropriate for the model, the prediction accuracy will decrease.

In [3] we proposed a new model that uses only the visiting time and visiting frequencies of pages without considering the access order of page requests in user sessions. Our experiments showed that Poisson distribution can be used to model user behavior during a single visit to a Web site. In that paper we examine the effect of several representation methods of time that a user spent on each page during her visit. In our previous work we employed a model-based clustering approach and partitioned user sessions according to the similar amount of time spent on similar pages. In this paper, we present a key extension to the representation of user transactions that improves the resulting accuracy for predicting the next request of a Web user. To confirm our findings, the results are compared to the results of two other well known recommendation techniques.

The rest of the paper is organized as follows. Section 2 briefly reviews the work related to model based clustering. Section 3 presents the proposed model. Section 4 provides detailed experimental results. In Section 5, we examine related work. Finally, in Section 6 we conclude our work.

## 2 Model-Based Cluster Analysis

In this section, we first describe the mixture model for clustering objects and then describe how the parameters of the clusters are derived in the context of the mixture model.

Model-based clustering methods optimize the fit between the given data and some mathematical model. Such methods are often based on the assumption that the data are generated by a mixture of underlying probability distributions, defined by a set of parameters, denoted $\mathbf{\Theta}$ [6]. An observation $\mathbf{x}_i$ in a data set of $K$ observations, $\mathbf{D} = \{\mathbf{x}_1, ..., \mathbf{x}_K\}$, is generated by a mixture of $G$ components as follows:

$$p(\mathbf{x}_i|\mathbf{\Theta}) = \sum_{g=1}^{G} p(c_g|\mathbf{\Theta}) p(\mathbf{x}_i|c_g, \mathbf{\Theta}_g) = \sum_{g=1}^{G} \tau_g p(\mathbf{x}_i|c_g, \mathbf{\Theta}_g) \tag{1}$$

where $\mathbf{\Theta}_g$ ($g \in [1...G]$) is a vector specifying the probability distribution function (pdf) of the $g^{th}$ component, $c_g$, and $\sum_{g=1}^{G} p(c_g|\mathbf{\Theta}) = \sum_{g=1}^{G} \tau_g = 1$.

Statisticians refer to such a model as *mixture model with G components.* The *maximum likelihood* (*ML estimation*) approach maximizes the *log* likelihood of the training data in order to learn the model parameters:

$$L(\mathbf{\Theta}_1, ..., \mathbf{\Theta}_G; \tau_1, ..., \tau_G|D) = \sum_{i=1}^{K} \ln \left( \sum_{g=1}^{G} \tau_g p(\mathbf{x}_i|c_g, \mathbf{\Theta}_g) \right) \tag{2}$$

## 3 Web Page Recommendation Model

In this research, we use three sets of server logs. The first one is from the NASA Kennedy Space Center server over the months of July and August 1995 [8]. The second log is from ClarkNet (C.Net)Web server which is a full Internet access provider for the Metro Baltimore-Washington DC area [7]. This server log was collected over the months of August and September, 1995. The last server log is from the Web server at the University of Saskatchewan (UOS) from June to December, 1995 [11]. For each log data set we apply the same pre-processing steps. Since the cleaning procedure is beyond the scope of this paper, the details of this procedure are not given here.

In this work, *visiting page times*[4], which are extracted during pre-processing step, are represented by four different normalization values in order to evaluate the effect of time to the prediction accuracy. The visiting times are normalized across the visiting times of the pages in the same session, such that the minimum value of normalized time is 1. We try 4 different maximum values: 2, 3, 5 and 10. If a page is not in the user session, then the value of corresponding normalized time is set to 0. This normalization captures the relative importance of a page to a user in a transaction. The *aggregate interest* of a user in a transaction is then defined by a vector which consists of the normalized visiting times of that transaction. The details of this step is given in [3].

Our previous work has presented a new model that uses only the visiting time and visiting frequencies of pages. The resulting model has lower run-time computation and memory requirements, while providing predictions that are at least as precise as previous proposals [3]. The key idea behind this work is that user sessions can be clustered according to the similar amount of time that is spent on similar pages within a session without considering the access order of page requests. In particular, we model user sessions in log data as being generated in the following manner: (i) When a user arrives to the Web site, his or her current session is assigned to one of the clusters, (ii) the behavior of that user in this session, in terms of visiting time, is then generated from a Poisson model of visiting times of that cluster. Since we do not have the actual cluster assignments, we use a standard learning algorithm, the Expectation-Maximization (EM) [4], to learn the cluster assignments of transactions as well as the parameters of each Poisson distribution. The resulting clusters consist of transactions in which users have similar interests and each cluster has its own parameters representing these interests. Our objective in this paper is to assess the effectiveness of *non-sequentially ordered* pages and the representation methods of normalized time values in predicting navigation patterns.

In order to obtain a set of pages for recommending and rank these pages in this set, *recommendation scores* are calculated for every page in each cluster using the Poisson parameters of that cluster. The cluster parameters of a cluster $c_g$ are then in the form:

$$pc_g = \{\tau_g; (rs_{g1}, ..., rs_{gn})\}$$

[4] It is defined as the time difference between consecutive page requests.

where $\tau_g$ is the probability of selecting the cluster $c_g$ and $rs_{gj}, j = [1...n]$ is the recommendation score of cluster $c_g$ at dimension[5] $j$. Those are the only parameters that the system needs in order to produce a set of pages for recommendation. We define the number of parameters stored in the memory as *model size*. It is clear that the smaller the model size the faster the online prediction.

We use five different methods for calculating recommendation scores for every page. The recommendation scores are then normalized such that the maximum score has a value of 1. These methods can be briefly summarized as follows: For the first method, we only use the Poisson parameters of the active cluster as recommendation scores. In the second method we use only the popularity of each page, which we define as the ratio of the number of the requests of a page in a cluster to the total number of page requests in that cluster. The intuition behind this is to recommend pages that are most likely visited in a cluster. For the third method, we calculate recommendation scores by multiplying the popularity by the Poisson parameter. For the last two methods we take advantage of a technique used in decision theory called the *entropy*. We calculate the entropy for each page using the relative frequency of each of the ten possible values of normalized times. A low entropy value means that the visiting time of that page mostly has one of the normalized values. High entropy value, on the other hand, indicates wide divergence in page visiting times among transactions. We calculate the recommendation scores of the fourth method by multiplying the inverse of entropy by popularity and Poisson parameters. For the last calculation, the *log* of the popularity is taken in order to decrease the effect of the popularity in recommendation score and is multiplied by the inverse of entropy and Poisson parameters.

The real-time component of the model calculates cluster posterior probability $P(c_g|w)$ for every cluster $c_g \in C = \{c_1, ..., c_G\}$ where $w$ is the portion of a transaction in test set that is used to find the most similar cluster. The active transaction is assigned to the cluster that has the highest probability. We define this cluster as the *active cluster*. A *recommendation set*, which is the set of predicted pages by the model, is then produced ranking the recommendation scores of active cluster in descending order.

## 4 Experimental Results

In this research we use three different transaction sets prepared for experiments as mentioned in Section 3. We measure the performance of our technique using the proposed methods for calculating recommendation scores. Approximately 30% of these cleaned transactions are randomly selected as the test set, and the remaining part as the training set. The experiments are repeated with different number of clusters and with different initial parameters for EM algorithm.

We define the following metrics to evaluate our method:

**Hit-Ratio** Given the visiting time of a page in the current transaction, the model recommends three pages that have the highest recommendation score

[5] Each page in the Web site corresponds a dimension in the model

in the active cluster. A hit is declared if any one of the three recommended pages is the next request of the user. The hit-ratio is the number of hits divided by the total number of recommendations made by the system.

**Precision** For each transaction $t$ in the test set we select the first $w$ requests in $t$. These $w$ requests are used to calculate the active cluster and produce the recommendation set. The recommendation set contains all the pages that have a recommendation score greater than the threshold $\xi$ and that are not in the first $w$ requests. We denote this set as $PS(w, \xi)$ and the number of pages in this set that match with the remaining part of active transaction as $m$. Then the precision for a transaction is defined as:

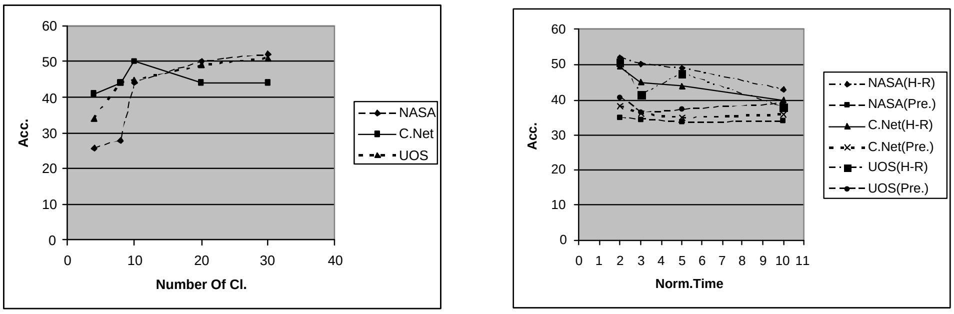
$$precision(t) = \frac{m}{|PS(w, \xi)|} \tag{3}$$

In our experiments, we try different values for the threshold, $\xi$, of recommendation scores ranging from 0.1 to 0.9. If the threshold is high then fewer recommendation are produced. If it is small then irrelevant pages are recommended with a low recommendation score. Our experiments show that setting $\xi$ to 0.5 and $w$ to 2 produces few but highly relevant recommendations. We perform the experiments with different number of clusters changing from 4 to 30. These experiments show that normalizing time between 1 and 2 improves the prediction accuracy. Due to lack of space, we just present the results of the experiments in which the normalized time has a value between 1 and 2. We identify that the values for the number of clusters in Table 1 are best among the other values we consider if page time is normalized between 1 and 2. For these numbers we have a higher *log* likelihood for the training sets as well as a better prediction accuracy for the test sets. The increase of the *log* likelihood means that the model fit better to the data. Figure 1(a) presents the prediction accuracy of the model for different number of clusters where time is normalized between 1 and 2. Figure 1(b) presents the prediction accuracy for different normalization values of time. As can be seen from Figure 1(a), the model is insensitive to the number of clusters in a reasonable range around the best numbers of clusters. The remarkable changes in the number of clusters results in a decrease of the performance of the model.

| Data Set | No.Of | Method 1 | | Method 2 | | Method 3 | | Method 4 | | Method 5 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Clusters | H-R | Pre. | H-R | Pre. | H-R | Pre. | H-R | Pre. | H-R | Pre. |
| NASA | 30 | 51.5 | 34.4 | 51.3 | 34.7 | 52 | 35 | 51.1 | 33.8 | 47.5 | 33.8 |
| C.Net | 10 | 48.7 | 37.9 | 49.2 | 37.6 | 49.6 | 38.2 | 48.2 | 35.4 | 46.6 | 32.9 |
| UOS | 30 | 50.8 | 40.6 | 50.6 | 40.7 | 50.8 | 40.7 | 50.5 | 39.3 | 50.1 | 38.7 |

**Table 1.** Results (in %) of the model. Visiting time is normalized between 1 and 2.

As mentioned in the previous section, we use 5 different methods for calculating recommendation scores. The application of methods that calculate the



(a) Number of Clusters-Accuracy

(b) Normalization values-Accuracy

**Fig. 1.** Impacts of number of clusters and normalization values on prediction accuracy

recommendation scores using popularity term results in marked improvement of the prediction accuracy. This is not surprising, because the popularity represents the common interest among transactions in each cluster. The results show that using entropy during calculation of recommendation score does not improve the accuracy. This is not surprising for the experiments where page time is normalized in a narrow range. However, even for a wide change in normalized time the entropy does not improve the prediction accuracy. This may be due to the fact that the popularity of some pages in most of the clusters are zero due to the sparse and scattered nature of the data. Thus, we can not calculate entropy values for most of the pages in a cluster. All of our experiments show that in general we can use method 3 for calculating recommendation scores discarding the metric we use for evaluation.

| Data Set | Poisson Model | Model 1 | Model 2 |
|---|---|---|---|
| NASA | 52 | 4 | 47.84 |
| C.Net | 49.6 | 15 | 49.3 |
| UOS | 50.8 | 5 | 44.59 |

**Table 2.** Comparison of recommendation models.

For evaluating the effect of the Poisson model, we repeated the experiments with the same training and test sets using two other recommendation methods[9, 10]. The recommendation model proposed in [10] (Model 1 in Table 2) is comparable to our model in terms of speed and memory usage. Since the hit-ratio metric has not performed well for the model in [10], we use the precision metric for evaluation. The C.Net data set has a precision of 15%, whereas the NASA data set has 4% and the UOS has 5%. Since the model in [9] is based on association rule discovery, it has obviously a greater model size than our model. We select this model in order to compare our results to the results of a model that uses a different approach. For the method in [9] (Model 2 in Table 2) we use a sliding window with a window size 2. The sliding window is the last portion of the active user session to produce the recommendation set. Thus, the model is able to produce the recommendation set only after the first two pages of the active user session. We set the support for association rule generation to a low value such as 1 % discarding the model size in order to have a good prediction accuracy. The hit ratio for the NASA, C.Net and UOS data sets are 47.8%, 49.3%, 44.50% respectively. These results prove that modelling the user transaction with a mixture of Poisson distributions produces satisfactory prediction rates with an acceptable computational complexity in real-time and memory usage when page time is normalized between 1 and 2.

## 5 Related Work

The major classes of recommendation services are based on collaborative filtering techniques and the discovery of navigational patterns of users. The main techniques for pattern discovery are sequential patterns, association rules, Markov models, and clustering.

Collaborative filtering techniques predict the utility of items of an active user by matching, in real-time, the active user's preferences against similar records (nearest neighbors) obtained by the system over time from other users [1]. One shortcomings of these approaches is that it becomes hard to maintain the prediction accuracy in a reasonable range while handling the large number of items (dimensions) in order to decrease the on-line prediction cost.

Some authors have used association rules, sequential patterns and Markov models in recommender systems. These techniques work well for Web sites that do not have a complex structure, but experiments on complex, highly interconnected sites show that the storage space and runtime requirements of these techniques increase due to the large number of patterns for sequential pattern and association rules, and the large number of states for Markov models. It may be possible to prune the rule space, enabling faster on-line prediction.

Page recommendations in [10] are based on clusters of pages found from the server log for a site. The system recommends pages from clusters that most closely match the current session. Two crucial differences between our approach and the previous one are that we consider the user interest as a statistical model and we partition user sessions using a model-based approach. As the experiments demonstrate, our model's precision and robustness is superior. Furthermore, our model has the flexibility to represent the user interest with a mixture of binomial distributions (or with different distributions) if one wishes to ignore the visiting time in determining the navigational pattern. We provide some intuitive arguments for why our model has an advantage in terms of speed and memory usage. The online prediction time correlates strongly with the model size. The smaller the model size the faster the online recommendation. Since we only store the cluster parameters for the prediction of the next page request, our model size is very small. The model size only increases with the number of clusters or the number of pages in the Web site when the Web site has a complex structure.

However, it is clear that in that case the application of methods such as sequential pattern mining, association rules or Markov models generate more complex models due to the increasing size of rules or states. Thus, all of these models require some pruning steps in order that they be effective. However, our model provides a high prediction accuracy with a simple model structure.

## 6 Conclusion

We have considered the problem of representing page time in a user session. In this article, the mixture of Poisson model is used for modelling the interest of a user in one transaction. The experiments show that the model can be used on Web sites with different structures. To confirm our finding, we compare our model to two previously proposed recommendation models. Results show that our model improves the efficiency significantly.

## References

1. J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.
2. 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.
3. Ş. Gündüz and M. T. Özsu. A user interest model for web page navigation. In *Proc. of Int. Workshop on Data Mining for Actionable Knowledge*, Seoul, Korea, April 2003. to appear.
4. A. P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statistical Society*, 39(1):1–38, 1977.
5. O. Etzioni. The world wide web: Quagmire or gold mine. *Communications of the ACM*, 39(11):65–68, 1996.
6. D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining.* The MIT Press, 2001.
7. ClarkNet WWW Server Log. http://ita.ee.lbl.gov/html/contrib/ClarkNet-HTTP.html.
8. NASA Kennedy Space Center Log. http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html.
9. B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd ACM Workhop on Web Information and Data Management*, pages 9–15, November 2001. Atlanta, USA.
10. B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Improving the effectiveness of collaborative filtering on anonymous web usage data. *Proceedings of the IJCAI 2001 Workshop on Intelligent Techniques for Web Personalization (ITWP01)*, Aug. 2001. Seattle.
11. The University of Saskatchewan Log. http://ita.ee.lbl.gov/html/contrib/Sask-HTTP.html.
12. C. Shahabi, A. Zarkesh, J. Adibi, and V. Shah. Knowledge discovery from users web-page navigation. *Proceeding of the IEEE RIDE97 Workshop, pages 20-29, Birmingham, England*, April 1997.