Bayesian Inference for Web Surfer Behavior Prediction

Malik Tahir Hassan, Khurum Nazir Junejo, and Asim Karim

Dept. of Computer Science, Lahore University of Management Sciences
Lahore, Pakistan
{mhassan, junejo, akarim}@lums.edu.pk

Abstract. The accurate prediction of user behavior on the Web has immense commercial value as the Web evolves into a primary medium for marketing and sales for many businesses. This broad and complex problem can be broken down into three more understandable problems: predicting (1) short and long visit sessions, (2) first three most probable categories of pages visited in a session, and (3) number of page views per category in a visit session. We present Bayesian solutions to these problems. The focus in our solutions is accuracy and computational efficiency rather than modeling the complex Web surfer behavior. We evaluate our solutions on four weeks of surfer data made available by the ECML/PKDD Discovery Challenge. Probabilities are estimated from the first three weeks of data and the resulting Bayesian models tested on last week's data. The results confirm the high accuracy and good efficiency of our solutions.

1 Introduction

Web users definitely exhibit patterns of surfing behavior. Discovering such patterns 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 has been a popular research topic. Over the years numerous approaches have been proposed for solving various aspects of the problem with varying degrees of success. In general, the problem involves the prediction of a user's sequence of page views based on previous history of the user. Oftentimes, to simplify the problem somewhat, Web pages are abstracted and grouped into categories and the problem is reduced to the prediction of a user's sequence of categories visited. Nonetheless, this is a complex machine learning problem that requires careful consideration from the technical and practical points of view.

Among the various approaches used for the modeling and prediction of Web surfer behavior, probabilistic approaches have been very common [1-5]. Borges and Levene [1] propose the use of N-gram probabilistic models which assume that the probability of a page visit depends only on the last N pages browsed. Similarly, Manavoglu et al. [2] present probabilistic user behavior models by applying maximum entropy and Markov mixture models. For prediction for known users, they propose a Markov

model. Another probabilistic solution is presented by Deshpande and Karypis [3]. They try to reduce the state complexity resulting from all kth-order Markov models by pruning many of the non-affecting states. Eirinaki et al. [4] present a hybrid probabilistic predictive model by extending the properties of Markov models with link-based methods such as PageRank. Such an approach is applicable only when structural link information of the pages is known. Lu et al. [5] group or cluster clickstream data using a pair-wise alignment algorithm. Then, a first-order Markov model is built for each cluster of sessions.

The majority of the approaches try to tackle the general Web surfer behavior modeling problem rather than specific prediction problems. This often makes the solutions complex and difficult to interpret. The Web surfer behavior prediction problem can be broken down into three sub-problems: (1) predicting short and long visit sessions by users, (2) predicting first three most probable categories of pages visited by users, and (3) predicting range of page views per category made by users. These sub-problems capture key Web surfer behaviors of practical value. Moreover, they represent simpler problems in comparison to the general Web surfer behavior prediction problem.

In this paper, we present Bayesian solutions to these problems. In particular, we develop Bayes classifiers for each sub-problem, invoking the naïve Bayes assumption of conditional independence of the input given the class. We model the sequence of page categories visited as a Markov chain. The naïve Bayes assumption and the first-order Markov property are made to improve space and time efficiency of the solutions. The performance of our solutions is evaluated on four weeks of data made available by the ECML/PKDD Discovery Challenge [6]. The results show high prediction performance identical to those produced by a support vector machine (for problem 1). Moreover, our solutions are time and space efficient.

The rest of the paper is organized as follows. We formally describe the Web surfer prediction problems in Section 2. Our solutions to the problems are described in Section 3. Experimental evaluation of the solutions, including their complexity analysis, is presented in Section 4. We conclude in Section 5.

2 Problem Description and Notation

Let variable $X = \{U, T\}$ identify a visit session, where variables U and T denote the user ID and the starting timestamp of the visit session, respectively. A visit session or path is described by a sequence of page categories visited during that session. Variable C_i identifies the category visited in position i of the sequence, and a visit session has one or more positions in the sequence. A particular visit session can have the same page category visited at different positions; however, two consecutive page categories must be different. Individual Web pages are abstracted and grouped into a finite number of page categories. The range of the number of page views made for a given page category is captured by the variable R_i, where i denote the ith position in the sequence. All variables have discrete and finite sets of possible values. The variable T is discretized into time slots. The historical training data available to the learning system contains unique visit sessions represented by instantiations of the

variables X, C_i's, and R_i's. The test data contain different instantiations of the variable X only.

The Web surfer behavior prediction problem is divided into three sub-problems. Problem 1 is to learn to predict whether a visit session X is short or long. A visit session is said to be short if it contains one sequence position only; otherwise, it is said to be long. Problem 2 is to learn to predict the first three page categories for a given visit session X. Problem 3 is to learn to predict the range of page views for each page category in positions 1, 2, and 3 for a visit session X. All three problems are classification problems. The objective in each is to predict the output as accurately as possible.

3 Our Solution

We present Bayesian solutions to the three problems described in the previous section. The Bayesian approach has been adopted for the following reasons: (1) it is simple and intuitive, providing insight into the problem and its solution, (2) it is adaptable to concept drift, and (3) it is computationally efficient and acceptably accurate. In particular, we use a Bayesian classifier for each of the three problems, as described in the following sub-sections.

3.1 Problem 1

This is a two-class classification problem. We present a naïve Bayes classifier for its solution. Given a visit session X, the most probable class $z = Z \in \{long, shart\}$ is given by

$$z = \underset{z \in \{long, short\}}{\arg\max} \; P(Z = z)P(X \mid Z = z) \tag{1}$$

where $P(.)$ denotes the probability of the enclosed event. If we assume that the user ID U and the timestamp T are conditionally independent of each other given class Z, we get the naïve Bayes classification:

$$z = \underset{z \in \{long, short\}}{\arg\max} \; P(Z = z)P(U \mid Z = z)P(T \mid Z = z) \tag{2}$$

This represents the most probable class under the naïve Bayes assumption. If we do not consider the timestamp T of a visit session, the last probability in Equation (2) drops out further simplifying the solution.

3.2 Problem 2

This problem involves the prediction of the first three page categories of a visit session. To solve this problem, we model the sequence of page categories visited as a Markov chain. The chain start state (the first page category) is determined by a Bayes

classifier. Subsequent states are determined by combining the posterior probability estimates given by the Markov chain with that of the Bayes classifier for that particular position. The reason for selecting the first-order Markov model over the kth-order model is two fold: (1) The problem involves the prediction of only the first three states for which a first-order Markov model is sufficient. and (2) The first-order Markov model is computationally efficient. Moreover, we believe that a kth-order model is more realistic for modeling page view transitions rather than page category transitions.

According to the Bayes rule, the posterior probability of page category C_i visited in position i (i = 1, 2, or 3) of a visit session X is given by

$$P^B(C_i \mid X) = P(C_i)P(X \mid C_i) / P(X) \tag{3}$$

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

$$c_1 = \arg\max_{c_1} P(C_1 = c_1)P(X \mid C_1 = c_1) \tag{4}$$

This fixes the start state of the Markov chain. The subsequent states can be found by combining the predictions of the Bayes classifier (Equation 3) and the Markov model. According to the Markovian property, for a given visit session X the posterior probability of page category C_i visited in position i (i = 2, 3) depends only on C_{i-1} and can be expressed as

$$P^M(C_i \mid C_{i-1}, X) = P(C_i \mid C_{i-1})P(X \mid C_i, C_{i-1}) / P(C_{i-1}, X) \tag{5}$$

The page category visited at position i (i = 2, 3) is then given by

$$c_i = \arg\max_{c_i} P^B(C_i = c_i \mid X)P^M(C_i = c_i \mid C_{i-1}, X) \tag{6}$$

Notice that in evaluating Equation (6), we do not need to estimate the unconditional probabilities in the denominator of Equations (3) and (5).

If a visit session X is described by user ID U and timestamp T, the naïve Bayes assumption can be invoked to simplify the expressions above, as shown for problem 1.

3.3 Problem 3

This problem involves the prediction of the range of the number of page views for the first three page categories visited in a visit session. The page categories c_i (i = 1, 2, 3) visited have been determined in problem 2. We use a Bayes classifier to predict the range $r_i = R_i$ of page views made at position i (i = 1, 2, 3) in visit session X as

$$r_i = \arg\max_{r_i} P(R_i = r_i \mid C_i = c_i)P(X \mid R_i = r_i, C_i = c_i) \tag{7}$$

The page category c_i is the one predicted in problem 2.

3.4 Estimating the Probabilities

The various probabilities used in our solution are estimated from the historical training data by maximum likelihood estimation. Since all variables are observed in the training data, maximum likelihood estimates are equivalent to the frequencies in the data. Specifically, the probability estimate of $P(X = x|Y = y)$ is given by

$$P(X = x \mid Y = y) \approx \frac{\text{no. of examples with } X = x, Y = y}{\text{no. of examples with } Y = y} \tag{8}$$

For an unconditional probability, the denominator will be the total number of examples in the training data. To estimate the transition probabilities in Equation (5), we count an example if it contains the given transition at any position of the sequence.

4 Evaluation

We carry out a number of experiments to demonstrate the efficiency and effectiveness of our solution to the Web surfer behavior prediction problem. The evaluations are performed on a desktop PC with an Intel 2.4 GHz Pentium 4 processor and 512 MB of memory.

4.1 Data and its Characteristics

We use the data provided by the 2007 ECML/PKDD Discovery Challenge [6]. The data were collected by Gemius SA, an Internet market research agency in Central and Eastern Europe, over a period of 4 weeks through use of scripts placed in code of the monitored Web pages. Web users were identified using cookies technology. The first 3 weeks of data are used for training while the last week of data are reserved for testing.

The data records individual visit sessions described by the fields: path_id, user_id, timestamp, {category_id, pageviews_number},…. An example visit session is shown below:

path_id	user_id	timestamp	path (category_id, pageviews_number) …, … …				
27	1	1169814548	7,1	16,2	17,9	16,1	…

The timestamp field records the time at which a visit session starts and the category ID field identifies a group of Web pages with similar theme such as entertainment, technology, or news. There are 20 page categories in the data. The entire data contain 545,784 visit sessions from which 379,485 visit sessions are used for training and the remaining 166,299 visit sessions are used for testing. There are 4,882 distinct users in the data.

An analysis of the training and test data reveals non-uniform data 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

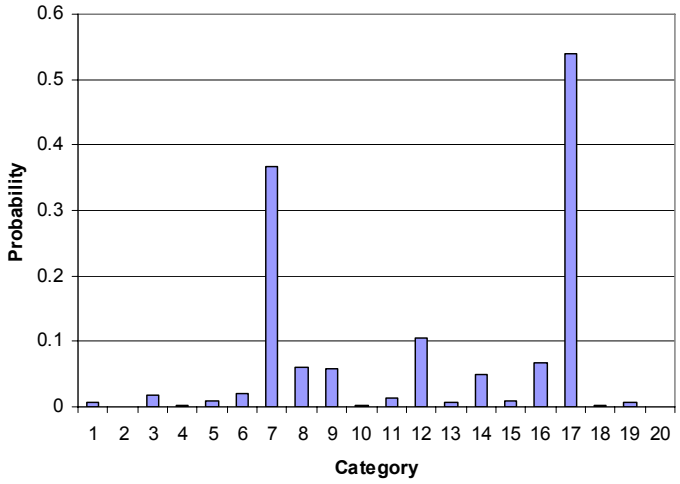

Fig. 1. Probability of categories in training data

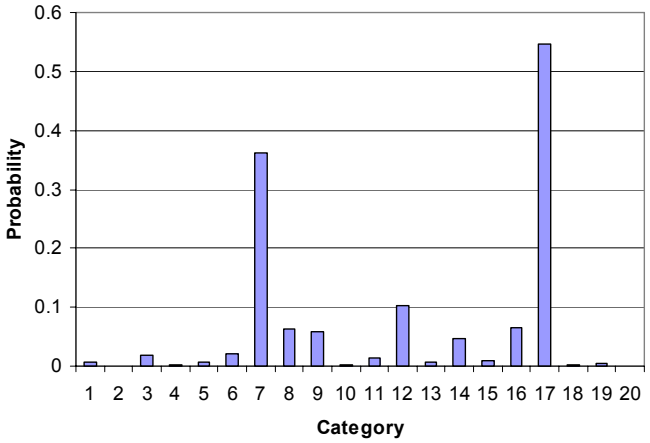

Fig. 2. Probability of categories in test data

number of visits by a user in the test data is 1 and 215, respectively. Similarly, the distribution of page categories is uneven. Some categories are being visited more frequently than others. This is evident from Figures 1 and 2 which show the probability of the categories in the training and test data, respectively. About 73% of the visit sessions in the training and data data are short, i.e., a visit where only one category is surfed. These statistics confirm that the data distributions of the test and training sets are similar.

4.2 Evaluation Criteria

Problem 1 is a two-class classification problem. The classification score, defined as the number of correct classifications, is used to evaluate this problem. Problem 2 is evaluated by computing a score. This score is the sum of scores of each prediction, where each prediction score is defined as follows: The prediction score is the sum of weights assigned to the 3 predicted categories. 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, 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 zero 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 2 cannot be greater than 4). Problem 3 is also evaluated by computing a score. This score computation is identical to that for problem 2 except that the weights are incremented by one if the predicted range is correct; otherwise they are not incremented. For all problems, higher scores signify better performance. The maximum possible score for each problem is also presented in our results

We present time and space complexity results in Sections 4.3 and 4.4.

4.3 Results

We present results for problems 1, 2 and 3 under two settings. In the first setting, we consider only the user ID as input while in the second setting we consider both the user ID and timestamp as input. We discretize the timestamp field into four values: weekday-day, weekday-night, weekend-day, and weekend-night. Daytime starts from 8AM and ends at 6PM. We tried several discretizations for timestamp but present results for the above defined discretization only. Problem 1 is also solved using a support vector machine (SVM) through SVM^{Light} [7]. The default parameters' settings of SVM^{Light} are used for this result.

The results for problems 1, 2, and 3 without and with timestamps are given in Tables 1 and 2, respectively. The accuracy of our solution without considering timestamps for problem 1 is 76.64%. The SVM also produces an accuracy of 76.64% for the same setting. Our achievement here is in terms of computational efficiency. For our hardware setup, our solution takes less than 1 minute to learn from the training data and classify the test data. In contrast, the SVM takes several hours to learn. When both user ID and timestamp are considered, the prediction performance of our solution drops slightly to 76.60% while that of SVM increases slightly to 76.68%. Including the timestamp field, however, decreases the time and space efficiency of the solutions.

A similar pattern of results is seen for the two settings of problems 2 and 3. For problem 2, the percentage score drops slightly from 83.2% to 83.17% when timestamp is considered together with user ID. On our hardware setup, it takes about 6 minutes without timestamps and about 15 minutes with timestamps to solve this problem (learning plus testing). For problem 3, the percentage score drops slightly from 72.53% to 72.42% when both timestamp and user ID are considered. Similarly,

Table 1. Prediction performance results for our solution without considering timestamp

	Problem 1	Problem 2	Problem 3
Score	127457	903145	958643
Max. possible score	166299	1085494	1321706
Percentage score	76.64%	83.20%	72.53%

Table 2. Prediction performance results for our solution when considering timestamp

	Problem 1	Problem 2	Problem 3
Score	127383	902849	957235
Max. possible score	166299	1085494	1321706
Percentage score	76.6%	83.17%	72.42%

the running time increases from about 1 minute to about 1.5 minutes when both timestamp and user ID are considered.

The interesting result is that considering timestamp decreases prediction performance (very) slightly (this drop may not be statistically significant). This is probably due to the greater chance of a probability estimate in our solution turning out to be zero adversely affecting the prediction. The SVM for problem 1 did show a slight increase in prediction performance. However, our solution is orders of magnitude more efficient than SVM.

4.4 Complexity Analysis

In this section, we discuss the computational complexity of our solution and demonstrate its efficiency.

The time complexity of our solution for all three problems is $O(N)$ where N is the total number of visit sessions in the data. The model is learned in $O(N)$ time and constant time is required to classify every test example as all the probabilities have been pre-computed.

The space complexity of our solution is defined by the number of probability estimates required. For problem 1, we require $2 + (4882 \times 2)$ estimates when timestamp is not considered and $2 + (4882 \times 2) + (4 \times 2)$ when timestamp is considered. In these expressions, 2 is the number of classes, 4882 is the number of distinct users, and 4 is the number of distinct timestamps. For problem 2 when timestamp is not considered, the number of probability estimates required is $(20 \times 3) + (4882 \times 20 \times 3) + (20 \times 20) + (4882 \times 20 \times 20)$. The first two terms correspond to the probabilities in Equation (3) and the last two terms correspond to the probabilities in Equation (5). When timestamp is considered an additional $(4 \times 20 \times 3) + (4 \times 20 \times 20)$ estimates are required. In these expressions, 20 is the number of categories and 3 is the number of positions.

For problem 3 without timestamp, the number of probability estimates required is $(3 \times 20) + (4882 \times 3 \times 20)$, where 3 is the number of page view ranges. When timestamp is considered, an additional $(4 \times 3 \times 20)$ estimates are required.

From the above results we see that space complexity is $O(N)$. As discussed earlier, the data is sparse and many probability estimates are zero. Hence smart selection of data structures can reduce the space requirements further. In our implementations, we use hash maps instead of matrices to store the non-zero probability values only.

5 Conclusion

In this paper, we present our solution to the 2007 ECML/PKDD Discovery Challenge on Web surfer behavior prediction. We adopt Bayesian approaches for all three problems of the challenge. For problems 1 and 3, which are standard classification problems, we use Bayes classifiers for their solution. For problem 2, which requires predicting the sequence of page categories visited, we combine Bayesian classification with Markov chain prediction. The solutions are evaluated on four weeks of data collected from Polish websites. The results show that our solutions are accurate and efficient. In particular, our solution to problem 1 has the same prediction accuracy as SVM but is orders of magnitude faster. We also find that incorporating the start time of visit sessions does not have any practical impact on prediction accuracy.

The problem of Web surfer behavior prediction is of immense commercial value. We believe that a direct solution to the problem is more practical than those involving complex Web surfer behavior modeling. As part of future work, we will explore other probability estimating approaches suitable for limited data and ways of boosting prediction performance.

References

1. J. Borges and M. Levene: Data mining of user navigation patterns. In *Proc. Of International Workshop on Web Usage Analysis and User Profiling* (1999)
2. E. Manavoglu, D. Pavlov, and C.L. Giles: Probabilistic user behavior models. In *Proc. Of International Conference on Data Mining (ICDM)* (2003)
3. M. Deshpande and G. Karypis: Selective Markov models for predicting Web page accesses. *ACM Transactions on Internet Technology* 4(2) (2004) 163-184
4. M. Eirinaki, M. Vazirgiannis, and D. Kapogiannis: Web path recommendations based on page ranking and Markov models. In *Proc. of the 7th Annual ACM international Workshop on Web Information and Data Management (WIDM)* (2005)
5. L. Lu, M. Dunham, and Y. Meng: Discovery of significant usage patterns from clusters of clickstream data. In *Proc. of WebKDD* (2005)
6. ECML/PKDD: Discovery challenge. http://www.ecmlpkdd2007.org/challenge (2007)
7. T. Joachims: Making large-scale SVM learning practical. In *Advances in Kernel Methods – Support Vector Learning*. MIT Press (1999)