# Web Customer Modeling for Automated Session Prioritization on High Traffic Sites

Nicolas Poggi[1], Toni Moreno[2,3], Josep Lluis Berral[1], Ricard Gavaldà[4], and Jordi Torres[1,2]

[1] Computer Architecture Department, U. Politècnica de Catalunya, Barcelona, Spain
[2] Barcelona Supercomputing Center, Barcelona, Spain
[3] Department of Management, U. Politècnica de Catalunya, Barcelona, Spain
[4] Department of Software, U. Politècnica de Catalunya, Barcelona, Spain

**Abstract.** In the Web environment, user identification is becoming a major challenge for admission control systems on high traffic sites. When a web server is overloaded there is a significant loss of throughput when we compare finished sessions and the number of responses per second; longer sessions are usually the ones ending in sales but also the most sensitive to load failures. Session-based admission control systems maintain a high QoS for a limited number of sessions, but does not maximize revenue as it treats all non-logged sessions the same. We present a novel method for learning to assign priorities to sessions according to the revenue that will generate. For this, we use traditional machine learning techniques and Markov-chain models. We are able to train a system to estimate the probability of the user's purchasing intentions according to its early navigation clicks and other static information. The predictions can be used by admission control systems to prioritize sessions or deny them if no resources are available, thus improving sales throughput per unit of time for a given infrastructure. We test our approach on access logs obtained from a high-traffic online travel agency, with promising results.

**Keywords:** Web prediction, navigation patterns, machine learning, data mining, admission control, resource management, autonomic computing, e-commerce.

## 1 Introduction

During the recent years there have been important changes in web technologies. There has been a shift from originally serving mainly static pages to fully dynamic sites. Dynamic applications have a huge demand on CPU power, opposed to network bandwidth that has been the traditional bottleneck of the web. Websites now make the use of fully featured programming languages, implementing XML-based web services for B2B communication, SSL for security, on-the-fly generated media, and technologies such as AJAX and for interactivity. While these technologies improve the user experience and privacy, they also increase the demand for CPU power [2].

C. Conati, K. McCoy, and G. Paliouras (Eds.): UM 2007, LNAI 4511, pp. 450–454, 2007.

With the increase of dynamic websites, system overload is becoming a common situation and its incidence is growing along. Improving the infrastructure of a website might not be simple; for cost reasons, scalability problems or because some peaks are infrequent, websites might not be able to adapt rapidly in hardware to user fluctuations. When a server is overloaded, it will typically refuse to serve any connections, as resources get locked and a race condition occurs. Session-based admission control systems [3] allow to maintain QoS on overloads by keeping a high throughput in terms of properly finished sessions for a limited number of users. However, by denying access to exceeding users, the website looses potential customers.

This paper proposes a novel approach consisting of generating a model for web user behavior in a real, complex website and using it to support decisions regarding the allocation of the available resources, based on a revenue-related metrics. In our user models, we try to understand how to best capture the features that make a customer more likely to make a purchase, and therefore more attractive — from the point of view of maximizing revenues — to maintain in the system even in the case of a severe overload. In this sense, we are proposing a per user-adaptive policy for admission control and session prioritization. Details on related work are given in the extended version [1].

## 2 Our Approach

Our approach consists in using web dynamic application log files to learn models that make predictions about each class of user future behavior, with the objective of assigning a priority value to every customer based on the expected revenue that s/he will generate, which in our case essentially depends on whether s/he will make a purchase. For this we have developed the AUGURES architecture, a prototype which currently implements: an access log preprocessor, to remove non-user generated actions and rewrite the log in a more convenient format; a module generating two high-order Markov chains, one for purchasing users and another for non-purchasing users; and an offline learning module (the predictor) running chosen classifiers form the WEKA machine learning package [4].

AUGURES is first trained by preprocessing a training log file, then the buying and non-buying Markov models are generated from it. Subsequently, each transaction on the training log is passed through both Markov models and their resulting probabilities added as static variables on the training log file. We then build the predictor from the training data; from this point, we can run incoming sessions from a new log file against the predictor, which will produce a probability on the users' purchasing intentions. In our generic approach we assume that from the log files we can extract at least the following information for each user transaction:

1. Date and time of transaction (discretized to a few categories).
2. Session identifier
3. "Tag", identifying the type of transaction performed by the page.

4. Whether the user is logged in at this moment.
5. Whether s/he is a returning customer, and whether s/he bought in the past.
6. Length of the current session, in number of transactions.
7. The "class", that is, the behavior we want to predict. This information is computed by looking "forward" in the logfile, so it can only be computed for the training set.

We call the previous information *static* because it reflects little information about the navigation path of the user in this session. On the other hand, it is reasonable to believe that the sequence of requests made by the user should help in predicting his/her future behavior. We call this sequence the *dynamic information* of the session; it can be identified by the sequence of tags (user clicks) in the associated transactions.

Unfortunately, most machine learning algorithms are not well adapted to dealing with variables that are themselves sequences, and some ad-hoc mechanism has to be designed. We propose to use higher-order Markov chains to obtain the extra information. More precisely, we model separately the navigation patterns of buyers and non-buyers with two order-$k$ Markov chains (we use $k = 2$). These Markov chains let us assign probabilities $\Pr[buyer|p]$ and $\Pr[nonbuyer|p]$ given that the last $k$ tags in the session are those described by the path $p$, see the extended version for details [1].

## 3 Experiments

The data for the experiment was provided by Atrapalo.com, a high traffic Spanish online travel agency, that makes use of the above mentioned state-of-the-art web technologies. It consisted of about 112,000 transactions collected over approximately one day. The data was preprocessed to remove erroneous entries, transactions clearly corresponding to automated bots and crawlers, and one-click sessions corresponding to banners. The resulting data contained 42 different tags or "pages" accessed by the users in their navigation.

An important feature of the data is that only about 2% of the sessions end in purchase; since buying sessions are longer in average than non-buying ones, this means about 6.6% of transactions have "buying" label. We prepared a training dataset of about 7,000 transactions. These were chosen randomly, except that we forced that about 50% were buying ones so that these were sufficiently represented. Another dataset was prepared for testing, containing the rest of the buying transactions plus a sufficient number of non-buying ones not appearing in the training dataset, so that the proportion of buyers was the original 6.6%.

After building a classifier using the training dataset, we can compute for each transaction in the testing set a "true" buying/nonbuying label and a "predicted" label. Thus, we can divide them into the 4 typical categories of true positives (tp), false positives (fp), true negatives (tn), and false negatives (fn). For example, false positives are the transactions that are predicted to be followed by purchase but that in fact did not.

| | j48 classifier | NB classifier | Logistic |
|---|---|---|---|
| %recall | 78.1 | 68.5 | 72.4 |
| %precision | 9.8 | 8.4 | 9.0 |

**Fig. 1.** Models built by different classifiers admitting N=30,000 transactions

The measures we are interested in this study are the classical *recall* and *precision*, as well as one that is specific to our setting, which we call *%admitted*.

- %admitted is (tp+fp)/(tp+fp+tn+fn), or the fraction of incoming transactions that would be admitted into the system. The number if allowed transactions is the one that may be limited by the available infrastructure.
- the recall is tp/(tp+fn), the fraction of real buyers that are admitted.
- the precision is tp/(tp+fp), the fraction of admitted transactions that really end up in purchase.

For the time being, we control the %admitted quantity by the ad-hoc but simple method of assigning different weights to buyers and nonbuyers when training. In a first set of experiments, we wanted to compare different learning methods. We used the 50%buyers-50%non-buyers training dataset to train a logistic linear regression (WEKA's `Logistic` method), a decision tree (WEKA's j48 method), and a Naive Bayes classifier.

We also fixed %admitted to about 28.5%, so that 30,000 transactions in the test dataset are admitted. The results are given in Figure 1. One can see that there are noticeable, but not drastic, differences in recall and precision among the methods. An important implication can be drawn from the recall figures, which reaches 78% for the j48 method: in an overload situation where less than 30% of the transactions can be admitted, a system admitting transactions at random, also a 30% of all buying transactions would be admitted, and 70% of buyers would be unserved; by using our mechanism, we would instead accept 78% of the buying transactions and leave only 22% buyers unserved.

In a second set of experiments we wanted to simulate the effect of different infrastructure capacity. We repeated the experiment above for different values of %admitted or, equivalently, for different numbers of admitted transactions $N$. We present the results (for the `j48` method only) on Figure 2.

One can observe that as admission is made harder ($N$ decreases), both recall and precision strictly grow. In other words, when less resources are available our system tends to let in only the most promising transactions.

| | N=5,000 | N=10,000 | N=30,000 | N=50,000 |
|---|---|---|---|---|
| %admitted | 4.2 | 10.5 | 28.2 | 42.4 |
| %recall | 40.6 | 54.41 | 78.1 | 85.8 |
| %precision | 34.5 | 18.3 | 9.8 | 7.0 |

**Fig. 2.** Models built by the `j48` classifier forcing %admitted to different values

## 4 Conclusions

Websites might become overloaded by certain events such as news events or promotions, as they can potentially reach millions of users. When a peak situation occurs most infrastructures become stalled and throughput is reduced even though there are more users. To prevent this, load admission control mechanisms are used to allow only a certain number of sessions, however current session based admission systems don't differentiate between users and might be denying access to users with the intention to purchase. As a proof of concept, we have taken a dataset from high traffic online travel agency to perform experiments to approximate users purchasing intentions from their navigational patterns.

In our experiments, we are able to train a model from previously recorded navigational information that can be used to tell apart, with nontrivial probability, whether a session will lead to purchase after a few clicks. From the results, in a situation where less than 30% of the transactions can be admitted, AUGURES would admit 78% of buying customers opposed to 30% from a random strategy. By assigning different weights to false positives and false negatives, the model can adapt itself dynamically maintaining a reasonable precision. As future work we plan to investigate other models to improve predictions, classification criteria, and at the same time test the applicability of the predictor models for a production environment. For further details please refer to the extended version or research group site [1].

## Acknowledgements

This work is supported by the Ministry of Science and Technology of Spain and the European Union under contract TIN2004-07739-C02-01. R. Gavaldà is partially supported by the 6th Framework Program of EU through the integrated project DELIS (#001907), by the EU PASCAL Network of Excellence, IST-2002-506778, and by the DGICYT MOISES-BAR project, TIN2005-08832-C03-03. Experiments data and domain knowledge provided by Atrapalo.com

## References

1. Poggi, N., Moreno, T., Berral, J., Gavalda, R., Torres, J.: Web customer modeling for automated session prioritization on high traffic sites. Technical Report, UPC, Group site (2006) at `http://research.ac.upc.edu/eDragon`
2. Guitart, J., Beltran, V., Carrera, D., Torres, J., Ayguadé, E.: Characterizing secure dynamic web applications scalability. In: 19th International Parallel and Distributed Processing Symposium, pp. 166–176. Denver, Colorado, USA (2005)
3. Guitart, J., Carrera, D., Beltran, V., Torres, J., Ayguadé, E.: Session-Based Adaptive Overload Control for Secure Dynamic Web Applications. In: 34th International Conference on Parallel Processing (ICPP'05), pp. 341–349. Oslo, Norway (2005)
4. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques 2nd edn. Morgan Kaufmann, San Francisco (2005) `http://www.cs.waikato.ac.nz/~ml/weka`