

# Dynamic Classification of Online Customers\*

Dimitris J. Bertsimas<sup>†</sup>

Adam J. Mersereau<sup>‡</sup>

Nitin R. Patel<sup>§</sup>

## Abstract

We explore methods for dynamic classification of visitors to an e-commerce web site based on visit sequences of page accesses. The time aspect is important in the processing of such data, and we require techniques that yield information before a customer's full sequence is realized. Further, we recognize that the timing of classification decisions may be important.

We focus on prediction of purchases based on site navigation paths, and explore two related problems. The first is incremental estimation of purchase probabilities. We develop a probability estimation model based on mixtures of Markov chains, and develop several extensions. Second, we consider dynamic classification of visits into "buy" and "non-buy" classes. We assume that at each click the merchant has three options: classify the visit as a "buy" visit, classify the visit as a "non-buy" visit, or await further information to be revealed. We examine dynamic decision rules—derived using dynamic programming—for generating these classifications from estimated probabilities, and compare them to schemes based on fixed probability thresholds.

We illustrate our methodologies on a real web log data set from a large retailer of computers. We demonstrate that probability estimation models based on second- and higher-order transition information outperform models of lower order. We show that both the fixed thresholds and the dynamic decision rules outperform a simple classification heuristic, and can be tuned to trade off the speed and accuracy of detection of both purchase visits and non-purchase visits.

**Keywords:** Markov models, dynamic programming, web mining.

## 1 Introduction

Despite the recent ups and downs of many companies doing business on the Internet, it is clear that electronic commerce has taken a firm hold. The questions of how best to understand and market to customers through

interactive media remain interesting and unsolved. Web personalization and electronic Customer Relationship Management (eCRM) have become buzzwords in both industry and academia.

A merchant doing business over an interactive medium faces the challenge of making use of the data available to him to dynamically manage his individual customer relationships. Although demographic and historical behavior information is valuable, the dynamics of the customer relationship are captured by evolving sequences of customer interactions. An Internet "click" sequence is a prime example of evolving data that gives timely information about a customer.

In this paper, we investigate approaches to time-sensitive classification of data sequences. We look at methods that are incremental, meaning that they can provide useful information about a customer before the customer's full data sequence is revealed. This allows the merchant to take an appropriate action while the customer is still in the system. We require decision processes that are sensitive to time and to the inherent tradeoff between gathering information and taking action. That is, we recognize that *when* to make a decision may be as important as *what* decision to make.

Our practical motivation is the problem of dynamic prediction of purchases on an e-commerce web site at the visit (or session) level. Our proposed methodology has two components: a model for estimating purchase probabilities and a time-sensitive framework for using these probabilities to make actionable classification decisions. Such classification decisions are potentially useful for personalization and timing of marketing interventions and promotions, for customization of site content, and for allocation of server resources.

The contributions of this paper are the following:

1. **A class of models for fast incremental estimation of purchase probabilities based on click sequences.** Our basic method is a Bayesian scheme involving mixtures of low-order Markov models of web navigation. Our scheme can be seen as a special case of both Hidden Markov Models [14] and Mixtures of Experts [8]. We also develop extensions of the technique for incorporating higher-order transition information, the elapsed time between clicks, and features such as demographic or browsing history variables.

\*This research was supported by a grant from MIT's Center for eBusiness.

<sup>†</sup>MIT Sloan School of Management, rm. E53-363, Cambridge, MA 02139, dbertsim@mit.edu

<sup>‡</sup>MIT Operations Research Center, rm. E40-130, Cambridge, MA 02139, ajm@mit.edu

<sup>§</sup>MIT Sloan School of Management, rm. E40-111, Cambridge, MA 02139, nitinrp@mit.edu

2. **Two frameworks for generating actionable, time-sensitive decision rules for classification.** Given an estimated probability of purchase at some point in time, does the merchant (a) classify the visit as a “buy” visit, (b) classify the visit as a “non-buy” visit, or (c) wait for more information to be revealed? The question couples a classification problem with a stopping problem. We examine it within the class of fixed threshold policies, and also through dynamic programming.
3. **Illustration of our approaches on a real web log data set from a large computer retailer.**

The organization of the paper is as follows. In Section 2 we briefly review related work in the data mining and marketing domains. In Section 3 we present and derive the models used for estimation of purchase probabilities and generation of decision rules. We describe in Section 4 experience and results from applying our methodology to a real dataset from a large computer retailer. Finally, in Section 5 we present conclusions and future research directions.

## 2 Related Work

The business value of web navigation data is widely recognized, and many methods for analyzing such data have appeared in the data mining literature. Recent conference workshops on web data mining (see [18] and [20]) highlight some of the latest research in the general area of web data mining. In addition, the paper by Srivastava et. al. [19] introduces and surveys the area of web usage mining. Kohavi [9] and references review opportunities and pitfalls in the practice of e-commerce data mining. The KDD-Cup 2000 competition showcased some of the most promising web usage mining practice through analysis of a real e-commerce data set. A summary can be found in Kohavi et. al. [10].

Popular approaches to web usage mining are those based on discovery of common sequential patterns and association rules (see Agrawal and Srikant [1]), on application of popular classification and clustering methods to feature representations of the data, and on invention of novel distance metrics for use in distance-based classification and clustering techniques. We employ a probability estimation methodology based on a Markovian model of web navigation behavior, and it has the advantages of providing a sound probabilistic framework conducive to extensions and of capturing the sequential nature of the data. Our work, in addition to the recent findings of Li et. al [11], demonstrates the value of the sequential aspect of web usage data. Popular classification methods such as support vector machines and tree-based classifiers operate in feature spaces of fixed dimension, and thus cannot naturally accommodate click sequences of variable lengths without some loss of infor-

mation.

The application of Markov chain models to web data is not new. Cadez et. al. ([4], [5]) use mixtures of Markov chain models for clustering and visualizing web usage data. In particular, our content categorization scheme and modeling assumptions are similar to theirs (see also papers by Smyth [16],[17], and references therein). Models based on Markov chains for predicting page accesses in a web navigation context have been employed recently by Deshpande and Karypis [7] and by Anderson et. al. [2]. While ours is a supervised learning task, we are interested not in predicting future page views but in predicting the class to which a given visit belongs. Markov chain-based analysis of sequences has long been used in fields such as computational biology and speech recognition, and our prediction approach can be seen as a special case of both Hidden Markov Models (see Rabiner [14]) and Mixtures of Experts (see Jordan and Jacobs [8]).

Analysis and modeling of web data is also of interest to the marketing community, and recent examples of work examining e-commerce purchase behavior are Moe and Fader [12], who develop statistical models of purchase behavior based on observed visit history, and Sismeiro and Bucklin [15], who examine purchase behavior by modeling completions of certain purchase-related tasks. Padmanabhan et. al. [13] train neural network, tree-based, and regression models to feature representations of web navigation data to illustrate the value of a customer’s navigation data external to a specific site. Perhaps closest to our work is the recent working paper of Li et. al. [11], which shares our focus on the use of detailed navigation paths to predict purchases within visits. They develop a complex statistical model of web navigation that includes a hidden Markov component. The methods proposed for model fitting and purchase prediction are more computationally intensive than ours, and the authors examine information, such as visitor demographics and browsing behavior at other sites, that may not be available to a retailer. Thus their work has a more descriptive but less operational focus than ours.

Our work is distinguished by our development of a dynamic classification framework, in which we solve a decision problem to trade off the quality and timeliness of our classification decisions. Cost-sensitive data analysis is a line of research that accounts for the tradeoffs among the prediction accuracies achievable for different classes. A bibliography on cost-sensitive data analysis can be found in [21], and a classic reference on statistical decision problems is [6].

## 3 Models and Methodology

**3.1 Problems and Notation.** We will use the term “click” to denote a single page view on an e-commerce

web site. We assume that we can attribute these page views to a user or consumer of the web site (e.g., through the use of cookie identifiers), and that we have a mechanism for segmenting the page views attributed to a single user into meaningful consecutive subsequences, or visits.

Our work takes the point of view of a merchant who observes a single visit sequence of clicks  $(i_1, i_2, \dots, i_T)$ , where the length  $T$  of the sequence is assumed unknown. We assume discrete time and that each click is taken from a finite alphabet  $\mathcal{I}$ , which in our implementation will represent a set of URL categories.

As mentioned in the paper introduction, we address two related problems. The first is estimation of the probability that a visit results in a purchase. We develop models that estimate these probabilities based only on the sequence of clicks  $I_t = (i_1, \dots, i_t)$  up to the  $t$ th click. We also develop a model that estimates purchase probabilities based on  $I_t$  and  $\theta$ , a variable whose value is known at the outset of the visit. Examples of such variables include demographic information and browsing or purchasing histories. We develop another model that estimates purchase probabilities based on  $I_t$  and  $E_t = (e_1, e_2, \dots, e_{t-1})$ , where  $e_t$  is the elapsed number of seconds between click  $i_t$  and  $i_{t+1}$ .

We also consider dynamic classification of visits into “buy” and “non-buy” visits. We assume that a merchant observes a visit evolving in the system. Immediately following click  $t$ , the merchant can classify the visit as a “buy” visit, classify the visit as “non-buy” visit, or make no classification. Once the visit has been classified, the decision is assumed irreversible. As the length of the visit sequence is unknown, we observe that failing to take an action at some point in time may result in the visit ending (i.e., the customer leaving the system) before any action is taken.

### 3.2 Model-Based Estimation of Purchase Probabilities

**The First-Order Model.** Our basic model probability estimation model is a mixture of two Markov chains: a “buy” chain that models visits that result in purchases and a “non-buy” chain that models visits that do not result in purchases. We indicate by  $B$  the event that an incoming visit is generated by the “buy” chain and by  $N$  the event that the visit is generated by the “non-buy” chain. We assume that, prior to the first click, nature assigns a visit randomly to the “buy” chain with probability  $\rho = \Pr\{B\}$  and to the “non-buy” chain with probability  $(1 - \rho)$ . The ensuing sequence of clicks is then assumed to be generated by the respective Markov chain. The Markov chain models include starting and ending states to model sequences of finite length, and are parameterized by transition probability matrices  $P^B$  and  $P^N$ . As an illustration, Figure 1 is the state dia-

gram for such a mixture model with  $\mathcal{I} = \{\text{“a”}, \text{“b”}\}$ .

We can think of the model as a special case of a hidden Markov model (HMM) with the transition matrix constrained to be block-diagonal. Fitting such a model from a set of training data is straightforward. The maximum likelihood estimates of the starting and transition probabilities are given simply by normalized transition counts observed among purchase visits and non-purchase visits in a training data set. The mixture probability  $\rho$  can be estimated by the fraction of visits in the training data that result in a purchase.

Once we have fit the model parameters, we can then dynamically score new sequences easily. Given the mixture probability  $\rho$  and the parameters of the Markov chain models, it is a simple application of Bayes’ Rule to compute the probability that a new incoming sequence is being generated by the “buy” chain. If we have observed the partial sequence  $I_t = (i_1, i_2, \dots, i_t)$ , our estimate of the probability of purchase is then:

$$\Pr\{B|I_t\} = \frac{\rho \cdot \Pr\{I_t|B\}}{\rho \cdot \Pr\{I_t|B\} + (1 - \rho) \cdot \Pr\{I_t|N\}},$$

where:

$$\begin{aligned} \Pr\{I_t|N\} &= P_{0,i_1}^N P_{i_1,i_2}^N \cdots P_{i_{t-1},i_t}^N, \\ \Pr\{I_t|B\} &= P_{0,i_1}^B P_{i_1,i_2}^B \cdots P_{i_{t-1},i_t}^B. \end{aligned}$$

$\Pr\{B|I_t\}$  can thus be quickly updated for each click in the visit sequence, thus providing at all points in time an updated estimate of the purchase probability.

The basic model as we have developed it is a mixture of first-order Markov chains. That is, conditional on whether a visit is generated by the “buy” or “non-buy” chain, we assume that the category of the next click in the visit depends only on the category of the current click.

**The Zero-Order Model.** An even simpler model than the mixture of first-order Markov chains involves replacing the first-order Markov chains with zero-order Markov models. Such models assume that each click is an independent random variable chosen from one of two multinomial distributions. Call the vectors of multinomial probabilities  $p^N$  and  $p^B$  for the “non-buy” and “buy” models respectively. We can then find the estimated probability of purchase as follows:

$$\Pr\{B|I_t\} = \frac{\rho \cdot \Pr\{I_t|B\}}{\rho \cdot \Pr\{I_t|B\} + (1 - \rho) \cdot \Pr\{I_t|N\}},$$

where:

$$\begin{aligned} \Pr\{I_t|N\} &= p_{i_1}^N p_{i_2}^N \cdots p_{i_t}^N, \\ \Pr\{I_t|B\} &= p_{i_1}^B p_{i_2}^B \cdots p_{i_t}^B. \end{aligned}$$

**Higher-Order Models.** The first-order model makes use of one-step transitions in the visit sequence. We

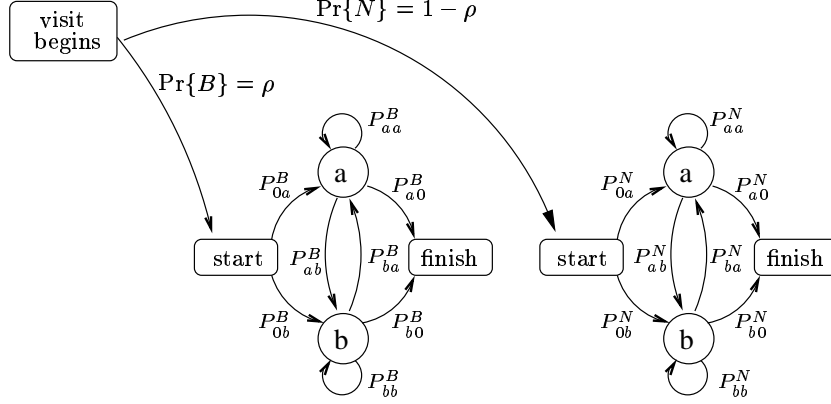


Figure 1: Mixture of First-order Markov Chains.

can obtain richer models by considering mixtures of higher order Markov models. For instance, a mixture of second-order Markov models assumes that, conditional on the visit being generated by either the “buy” or “non-buy” model, the category of the next page request in a visit depends on the previous two page requests (Note that the  $t = 1$  case must be handled specially).

Fitting a mixture of  $n$ th-order Markov models requires estimation of  $O(|\mathcal{I}|^n)$  probabilities from the training data. A significant complication with such models is that of poor coverage. That is, there may be many transitions in the model for which there are few or no corresponding examples in the training data set, making it difficult to accurately estimate model parameters.

An approach which attempts to avoid this difficulty is to use a model that at each click uses transition information of different order depending on the coverage in the training data set. Deshpande and Karypis [7] study an approach with this flavor for predicting the next web page request in a visit.

We have implemented a version of our model that includes higher-order transition information in a related way. The model’s transition probabilities can be stored in an  $(|\mathcal{I}| + 1)$ -ary tree  $\mathcal{T}$  in which a node at depth  $m > 1$  corresponds to a unique set of  $m - 1$  previous clicks. At the node corresponding to the sequence  $\{i_{j-1}, \dots, i_{j-m-1}\}$ , we store the transition probabilities:

$$\begin{aligned} \Pr(i_j | i_{j-1}, \dots, i_{j-m-1}, \{i_{j-1}, \dots, i_{j-m-2}\} \notin \mathcal{T}, N) \\ \text{for each } i_j \in \mathcal{I}, \\ \Pr(i_j | i_{j-1}, \dots, i_{j-m-1}, \{i_{j-1}, \dots, i_{j-m-2}\} \notin \mathcal{T}, B) \\ \text{for each } i_j \in \mathcal{I}. \end{aligned}$$

After generating a full tree up to a prespecified depth, we prune the tree based on the number of training set observations available to estimate the transition

probabilities at each node. The pruned tree represents a mixture of Markov models of variable order.

When we score a new sequence  $I_t$ , the probabilities  $\Pr(i_j | I_{j-1}, N)$  and  $\Pr(i_j | I_{j-1}, B)$  are obtained by finding the deepest node in the tree consistent with the sequence  $I_{j-1}$ . The estimated probability of purchase can then be obtained using Bayes’ Rule as before, where:

$$\begin{aligned} \Pr\{I_t | N\} &= P_{0,i_1}^N \Pr(i_2 | I_1, N) \cdots \Pr(i_t | I_{t-1}, N), \\ \Pr\{I_t | B\} &= P_{0,i_1}^B \Pr(i_2 | I_1, B) \cdots \Pr(i_t | I_{t-1}, B). \end{aligned}$$

**Incorporating Other Covariates.** The models we have discussed so far take into account only the sequence of page views, but in general we may want the capability to incorporate other types of information into the model. We briefly discuss ways to do this.

Suppose we would like to account in the first-order Markov chain mixture model for a variable  $\theta$ , a (possibly multivariate) random variable that takes a value at the outset of each visit. Examples of  $\theta$ ’s of this type include demographic variables and features derived from a customer’s historical browsing or purchasing behavior (e.g. the amount of time spent by a customer on the site in the past, the number of previous visits, and the time since the last purchase). We assume the following quantities are functions of  $\theta$ :

$$\begin{aligned} \rho(\theta) &= \Pr\{B | \theta\}, \\ P_{i_{j-1}, i_j}^N(\theta) &= \Pr\{i_j | i_{j-1}, N, \theta\}, \\ P_{i_{j-1}, i_j}^B(\theta) &= \Pr\{i_j | i_{j-1}, B, \theta\}. \end{aligned}$$

With this notation established, our estimated purchase probability for the click sequence  $I_t = (i_1, \dots, i_t)$  is given by the same application of Bayes’ Rule as before, but conditional on the value of  $\theta$ :

$$\Pr\{B|I_t, \theta\} = \frac{\rho(\theta) \cdot \Pr\{I_t|B, \theta\}}{\rho(\theta) \cdot \Pr\{I_t|B, \theta\} + (1 - \rho(\theta)) \cdot \Pr\{I_t|N, \theta\}},$$

where:

$$\begin{aligned} \Pr\{I_t|N, \theta\} &= P_{0,i_1}^N(\theta) P_{i_1,i_2}^N(\theta) \cdots P_{i_{t-1},i_t}^N(\theta), \\ \Pr\{I_t|B, \theta\} &= P_{0,i_1}^B(\theta) P_{i_1,i_2}^B(\theta) \cdots P_{i_{t-1},i_t}^B(\theta). \end{aligned}$$

If  $\theta$  takes values of zero or one, then we can fit this model by estimating  $\rho(\theta)$  and the Markov chain transition probabilities separately for each value of  $\theta$ . If  $\theta$  represents continuous, categorical, or multiple covariates, we replace  $\rho(\theta)$  and the Markov chain transition probabilities with functions. We have chosen logistic functions because they take values between zero and one, they are often used to model discrete choice, and they can be fit using well-studied logistic and polytomous regression techniques. A similar use of logistic functions in Markov chain mixture models was noted in Smyth [17] in a clustering framework.

We use  $\alpha$  to denote the vector of weights in the logistic function approximating  $\rho$ . For every  $i, k \in \mathcal{I}$ , we use  $\beta_{i,k}^N$  and  $\beta_{i,k}^B$  to denote the vector of weights in the logistic functions approximating  $P_{i,k}^N$  and  $P_{i,k}^B$  respectively. These approximations are then as follows:

$$\begin{aligned} \rho(\theta) &= \frac{\exp(\alpha \cdot \theta)}{1 + \exp(\alpha \cdot \theta)}, \\ P_{i_{j-1},i_j}^N(\theta) &= \frac{\exp(\beta_{i_{j-1},i_j}^N \cdot \theta)}{\sum_{i \in \mathcal{I}} \exp(\beta_{i_{j-1},i}^N \cdot \theta)}, \\ P_{i_{j-1},i_j}^B(\theta) &= \frac{\exp(\beta_{i_{j-1},i_j}^B \cdot \theta)}{\sum_{i \in \mathcal{I}} \exp(\beta_{i_{j-1},i}^B \cdot \theta)}. \end{aligned}$$

(Here we assume that  $\theta$  includes a component of ones, such that the expressions  $\alpha \cdot \theta$ ,  $\beta_{i_{j-1},i}^B \cdot \theta$ , and  $\beta_{i_{j-1},i}^N \cdot \theta$  include intercept terms.)

A potentially valuable set of information about a visitor's web usage behavior is the elapsed time between clicks. We discuss a simple extension to the probability estimation model that incorporates the time between clicks into the procedure. Assume that at some point in time we have observed a partial sequence of clicks  $I_t = (i_1, i_2, \dots, i_t)$  and a corresponding set of inter-click times  $E_t = (e_1, e_2, \dots, e_{t-1})$ . In this notation, we assume that  $e_j$  time units transpire between clicks  $i_j$  and  $i_{j+1}$ . We make the modeling assumption that, given the data is generated by either the "buy" or "non-buy" model, the joint distribution of  $e_j$  and  $i_{j+1}$  depends on the past only via  $i_j$ .

Given these assumptions, incorporation of inter-click times into the model is straightforward. Given  $(I_t, E_t)$ , our estimate of the purchase probability is:

$$\Pr\{B|I_t, E_t\} = \frac{\rho \cdot \Pr\{I_t, E_t|B\}}{\rho \cdot \Pr\{I_t, E_t|B\} + (1 - \rho) \cdot \Pr\{I_t, E_t|N\}}$$

where:

$$\begin{aligned} \Pr\{I_t, E_t|B\} &= \Pr\{i_1, \dots, i_t, e_1, \dots, e_{t-1}|B\} \\ &= \Pr\{i_1|B\} \cdot \Pr\{e_1, i_2|i_1, B\} \\ &\quad \cdot \Pr\{e_2, i_3|i_2, B\} \cdots \\ \Pr\{I_t, E_t|N\} &= \Pr\{i_1, \dots, i_t, e_1, \dots, e_{t-1}|N\} \\ &= \Pr\{i_1|N\} \cdot \Pr\{e_1, i_2|i_1, N\} \\ &\quad \cdot \Pr\{e_2, i_3|i_2, N\} \cdots \end{aligned}$$

It remains to estimate the terms  $\Pr\{e_j, i_{j+1}|i_j, B\}$  and  $\Pr\{e_j, i_{j+1}|i_j, N\}$ . If we discretize the  $e_j$  (say, into "small," "medium," and "large" inter-click times) such that we can write  $e_j \in \mathcal{E}$  for a finite set  $\mathcal{E}$ , then these values can be estimated simply by counting in the training data and normalizing such that the probability estimates  $\hat{\Pr}\{e_j, i_{j+1}|i_j, B\}$  and  $\hat{\Pr}\{e_j, i_{j+1}|i_j, N\}$  satisfy  $\sum_{e \in \mathcal{E}} \sum_{k \in \mathcal{I}} \hat{\Pr}\{e, k|B, i\} = \sum_{e \in \mathcal{E}} \sum_{k \in \mathcal{I}} \hat{\Pr}\{e, k|N, i\} = 1$  for all  $i \in \mathcal{I}$ .

**3.3 Generation of Decision Rules.** Once we are able to estimate at each click the probability that the visit results in a purchase, the question remains how to use this information to make classification decisions in time. We assume that immediately following each click, we may either classify the visit as a "buy" visit, classify the visit as a "non-buy" visit, or make no classification (and await further information to be revealed). Once a visit has been classified, the classification is assumed irreversible.

We examine two classes of decision rules for making timely classification decisions based on an evolving sequence of probability estimates. The first class of rules we consider are based on fixed probability thresholds that remain constant through time. A second class of decision rules are allowed to vary in time.

**Fixed Threshold Rules.** As there are three possible decisions available at any click, we require two thresholds to define a set of fixed threshold rules. Assume that we have probability estimates  $\Pr\{B|I_t\}$  (or,  $\Pr\{B|I_t, E_t\}$  or  $\Pr\{B|I_t, \theta\}$ ) for some visit for each  $t \leq T$ . Define  $s_N$  and  $s_B$  (Assume  $s_N \leq s_B$ .) such that we classify a visit as a "buy" visit or a "non-buy" visit after the first click  $t$  such that  $\Pr\{B|I_t\} \leq s_N$  or  $\Pr\{B|I_t\} \geq s_B$  respectively. A sequence unclassified prior to click  $t$  and with  $s_N \leq \Pr\{B|I_t\} \leq s_B$  is assigned no classification at click  $t$ . In our implementation, we will discretize the probability estimates so that

probabilities are discrete quantities from a finite set  $\mathcal{P}$ . In this framework there are  $(|\mathcal{P}| + 2)(|\mathcal{P}| + 1)/2$  sets of fixed threshold rules possible. We can apply all possible sets of fixed threshold rules to a validation set and choose the one that optimizes a performance measure of interest.

**Dynamic Decision Rules.** In the class of dynamic decision rules, we allow the classification decision to depend on both the estimated probability  $\Pr\{B|I_t\}$  and on  $t$  itself. This is a much larger class of decision rules than in the fixed threshold case, and there are too many possible sets of decision rules to search exhaustively. We approach the generation of dynamic decision rules through dynamic programming. (A reference on dynamic programming is [3].) The output of the dynamic program is, at each click  $t$  and for every possible probability estimate, one of “classify as buy,” “classify as non-buy,” or “make no classification.” We generate such decision rules up to a planning horizon  $H$ , which we define to be the last click at which we allow ourselves to make a classification decision. We assume that the consequences of our classification actions are the following costs, which we can view either as tuning parameters or as real costs in a dynamic marketing system:

- $c_B^B(t), c_B^N(t) :=$  cost of classifying as “buy” or “non-buy” respectively immediately following click  $t$  when visit results in at least one purchase.
- $c_N^B(t), c_N^N(t) :=$  cost of classifying as “buy” or “non-buy” respectively immediately following click  $t$  when visit does not result in a purchase.
- $c_0(t) :=$  cost of the customer leaving the system after click  $t$  with no classification applied.

As we assume that classifications are irreversible, exactly one of these costs is realized at some point during each visit. We derive decision rules by minimizing the expected visit cost given this cost structure.

We consider the evolution of a single visit sequence, and formulate the dynamic program by specifying the state, randomness, available controls, system dynamics, and cost structure of the system over which we are optimizing.

#### State

The state  $x_t$  of the visit at time  $t$  is defined as follows. Let  $x_t$  equal the current purchase probability estimate if the visit is eligible for classification at click  $t$ , and take the value -1 if the visit is not eligible for classification at click  $t$ . This can occur either because it has already been labeled or because the sequence is shorter than  $t$  clicks (i.e., the customer has left the system). In our implementation, we discretize the probability estimates so that the estimates are

quantities from a finite set  $\mathcal{P}$ . Thus  $x_t$  will take values on  $\mathcal{P} \cup \{-1\}$ .

#### Randomness

Define the following random variable for each click  $t$ :

$$P_t = \begin{cases} \Pr\{B|I_{t+1}\}, & \text{if } t + 1 \leq T \\ -1, & \text{if } t + 1 > T \end{cases}.$$

Thus, if the visit includes a  $(t + 1)$ th click, then  $P_t$  gives the (discretized) purchase probability estimate immediately after click  $t + 1$ . If the visit has fewer than  $t + 1$  clicks, then  $P_t = -1$ . We assume that the sequence  $\{P_0, \dots, P_{H-1}\}$  evolves according to a non-stationary Markov process with the following transition probabilities:

$$Q_{p_1, p_2}^t = \Pr\{P_{t+1} = p_1 | P_t = p_2\}, \\ \forall t = 1, \dots, H, \forall p_1, p_2 \in \mathcal{P} \cup \{-1\}.$$

Note that  $Q_{-1, -1}^t = 1, \forall t = 1, \dots, H$ .

Having trained one of the models of Section 3.2 using a training data set, we use the fitted model to compute the value  $P_t$  for each click in a validation data set. These validation set estimates are then used to estimate the elements of the transition probability matrices  $Q^t$ . Our evaluation of the resulting classification model will rely on an independent test set.

We distinguish the assumption that the probability estimates evolve in a Markovian fashion from the Markov models of web navigation used in Section 3.2 to generate these probability estimates. In developing the dynamic programming model, we assume that probabilities evolve according to a Markov process, but we do not rely on the details of the probability estimation model. Dynamic programming models that specifically rely on the Markovian nature of our probability estimation models may yield improved improved results, and we leave this topic for future work.

#### Control

The control  $u_t$  at click  $t$  takes one of three values:  $u^B$  (classify as “buy”),  $u^N$  (classify as “non-buy”), and  $u^0$  (make no classification). If  $0 \leq x_t \leq 1$  then we allow the control  $u_t$  to take values on  $\{u^B, u^N, u^0\}$ , and for the case  $x_t = -1$  we require  $u_t = u^0$ .

#### System Dynamics

Based on the control  $u_t$  and the random quantity  $P_t$ , the state  $x_t$  of the visit evolves according to the following system dynamics:

$$x_1 = P_0 \\ x_{t+1} = \begin{cases} -1 & \text{if } u_t = u^B, u_t = u^N, \text{ or } x_t = -1 \\ P_t & \text{otherwise.} \end{cases}$$

### Expected Costs

In terms of the cost structure mentioned previously, we can write the expected costs  $g(x_t, u_t)$  at click  $t$  as a function of the state  $x_t$  and the control  $u_t$ :

$$\begin{aligned} g_t(-1, u^0) &= 0, \\ g_t(x_t, u^B) &= x_t c_B^B(t) + (1 - x_t) c_B^N(t), \\ &\quad \text{for } 0 \leq x_t \leq 1, \\ g_t(x_t, u^N) &= x_t c_N^B(t) + (1 - x_t) c_N^N(t), \\ &\quad \text{for } 0 \leq x_t \leq 1, \\ g_t(x_t, u^0) &= Q_{x_t, -1}^t c_0(t), \text{ for } 0 \leq x_t \leq 1, t < H, \\ g_H(x_H, u^0) &= c_0(H), \text{ for } 0 \leq x_H \leq 1. \end{aligned}$$

### Dynamic Programming Iteration

We can now write the dynamic programming iteration that solves the problem of selecting the optimal set of thresholds. We define the value function  $J_t(x_t)$  as follows. Set  $J_t(-1) = 0$  for  $t = 1, \dots, H$ . For  $x_t \in \mathcal{P}$  we have:

At click  $H$ :

$$J_H(x_H) = \min \left\{ \begin{array}{l} x_H c_B^B(H) + (1 - x_H) c_B^N(H), \\ x_H c_N^B(H) + (1 - x_H) c_N^N(H), \\ c_0(H) \end{array} \right\}.$$

At clicks  $t = H - 1, \dots, 1$ :

$$J_t(x_t) = \min \left\{ \begin{array}{l} x_t c_B^B(t) + (1 - x_t) c_B^N(t), \\ x_t c_N^B(t) + (1 - x_t) c_N^N(t), \\ \sum_{p \in \mathcal{P}} Q_{x_t, p}^t J_{t+1}(p) + Q_{x_t, -1}^t c_0(t) \end{array} \right\}.$$

In each of the two expressions above, the first term in the minimization gives the expected cost of a “buy” classification, the second term gives the expected cost of a “non-buy” classification, and the third term gives the expected cost of postponing the classification decision. The optimal decision rule for state  $x_t$  at click  $t$  is determined by which of the three terms achieves the minimum in these expressions. We can solve for these decision rules in time proportional to  $H|\mathcal{P}|^2$ .

## 4 Implementation and Results

**4.1 Data.** We demonstrate our methodology on a data set derived from the web logs of a large retailer of computers. As is typical among online computer retailers, the site includes separate but parallel sections targeted towards home users, small business customers, educational customers, etc. Each section includes pages

devoted to general and specific product information, special offers, and order status. Also available are “configuration” pages where customers can customize product specifications and view prices, a shopping cart function, and a series of checkout pages.

The raw data corresponds to page views from a representative sample of purchasing users and non-purchasing users from all sections of the web site over a 2-month period. Users were identified using cookie information, and then separated into visits at 30-minute breaks in activity.

The web site in question is vast. As is often the case with real web log data, the data required several cleaning steps to improve its consistency and to focus on the types of users of value to the retailer. We eliminated users with very many or very few page views on the site (This filtering was done at the *user* level, meaning that many *visits* with few clicks remained.), and with significant numbers of clicks in sections of the web site less interesting to the retailer. We realize that dynamic classification requires that any data filtering of new data be implementable in real time, although we have not strictly adhered to this constraint here.

The resulting data set includes approximately 200,000 visits with clicks among approximately 7,000 unique URLs. There are an average of 10.7 clicks per visit, and approximately 2% of visits include a purchase. We note that these numbers are not necessarily representative of the raw data due to the cleaning and filtering steps we employed.

We assigned the data randomly by user to a training set (50%), validation set (33%), and test set (17%). The training set is used to fit the various probability estimation models described in Section 3.2. We use these fitted models to estimate purchase probabilities for the visits in the validation set. These probabilities are used in Section 4.4 to evaluate the probability estimation models, and are also used to estimate the parameters required for the decision rule generation procedure described in Section 3.3. The test set is used to generate the results of Section 4.5.

**4.2 Content Categorization.** As the original data set includes on the order of 7,000 distinct URLs, it is essential to reduce this set of symbols in order to meaningfully apply techniques based on Markov chains. To this end we have summarized this URL information using a reduced set of content categories. In web mining practice this is commonly approached by “stemming,” which makes use of the directory information in the URL address string. As the web site is not organized in such a way that “stemming” gives consistently useful categories, we have taken a modified approach.

We assign URLs to categories by searching for keywords in the URL addresses themselves. The results

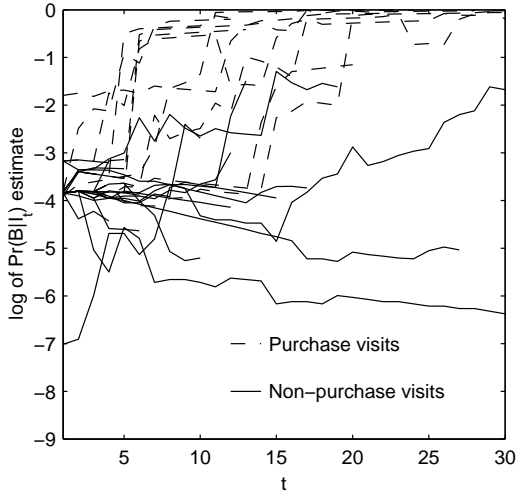


Figure 2: Examples of estimated probability trajectories produced by a mixture of second-order Markov models. Note: The numbers of purchase and non-purchase visits were not chosen proportional to their incidence in the data.

are 11 content categories that include categories such as “Homepages,” “Special offers,” “Choose and Configure,” “Checkout,” and “Order Status.” We associate each category with a letter from “a” to “k.”

The content categorization scheme allows us to represent a visit sequence in a compact manner (e.g., “aadfgggg”), and it sharply reduces the number of states in the Markov model and hence the number of model parameters to be fit.

**4.3 Overview of a Dynamic Classifier.** We have discussed two modeling components that constitute the dynamic classifier: models for estimating purchase probabilities and decision rules for time-sensitive classification decisions. To illustrate the output of the probability estimation models, in Figure 2 we plot estimated probability trajectories for several purchase and non-purchase visits randomly selected from the validation data. Results from a mixture of second-order Markov models are presented. Figure 2 provides a reality check of our probability estimation models, and we see that our model appears capable of achieving some separation between most purchase visits and most non-purchase visits by the end of the visits. We observe that identifying purchase visits is straightforward once we have observed a sequence of “Checkout” activity. Thus the interesting challenges are to identify purchase visits before they enter the checkout sequence and to identify non-purchasers before they leave the system.

Trajectories like those in figure 2 represent the output of the probability estimation models and the input to the decision rule models. Table 1 illustrates

Prob. Est. \ t	1	2	3	4	5	6	7	8	9	10
> 0.6065	Classify as “buy”									
0.3679-0.6065	Make no classification									
0.2231-0.3679										
0.1353-0.2231										
0.0821-0.1353										
0.0498-0.0821										
0.0302-0.0498										
0.0183-0.0302										
0.0111-0.0183										
0.0067-0.0111										
0.0041-0.0067										
0.0025-0.0041	Classify as “non-buy”									
0.0009-0.0025										
< 0.0009										

Table 1: Example dynamic decision rules generated using dynamic programming and based on probability estimates generated using second-order Markov models.

the concept of dynamic decision rules. For each  $t$  (up to a planning horizon of  $H = 10$ ) and every possible probability estimate (discretized on a grid of 14 bins), Table 1 shows example decision rules produced by the dynamic programming algorithm. Note that fixed threshold rules would appear on such a table as straight horizontal classification boundaries. Using this table, we classify a visit as a “buy” visit the first  $t$  for which the estimated probability enters the “Classify as buy” region, and classify “non-buy” visits analogously.

This table says that a given click we should classify high probability visits as “buy,” low probability visits as “non-buy,” and postpone classification of the remaining visits until more information is revealed. At the end of the planning horizon approaches, the “buy” and “non-buy” regions touch, since in our formulation there is no value to postponing classification beyond the planning horizon.

**4.4 Evaluation of Probability Estimates.** For purposes of comparison, we have implemented several purchase probability estimation schemes. The parameters of each are estimated using the training data set.

- Zero : Mixture of zero-order Markov models with no covariates.
- One : Mixture of first-order Markov models with no covariates.
- Two : Mixture of second-order Markov models in which we estimate purchase probabilities for all possible second-order transitions. This is the model represented in Figure 2.
- Pruned : Mixture of Markov models that include transitions up to fourth order, pruned back

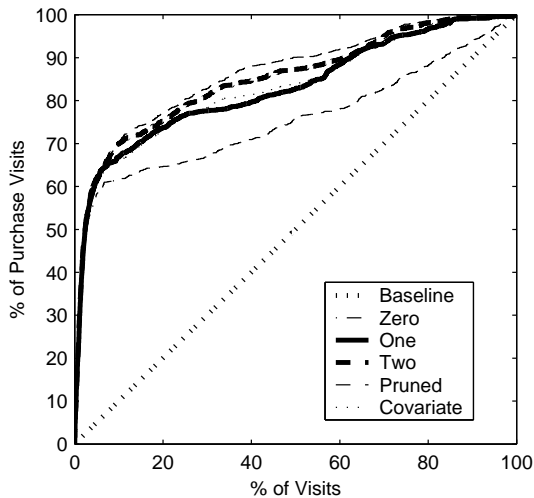


Figure 3: Cumulative lift curves for various probability estimation models.

as described in Section 3.2 under the subheading “Higher Order Models.” We found that including transitions higher than fourth order did not yield noticeable improvements in the quality of the probability estimates.

- **Covariate** : Mixture of first-order Markov models incorporating a numerical covariate, as described in Section 3.2. The covariate used is the number of clicks in the “Choose and Configure” page category observed in the user’s previous visit (zero if the current visit is the first observed visit associated with the user). The particular covariate was chosen by running logistic regressions of purchase probability on several potential covariates and choosing the covariate with coefficient most significantly different from zero.
- **Time** : The extension of the first-order model that allows us to incorporate inter-click times into the prediction model. For modeling purposes, we discretize inter-click times into “small,” “medium,” and “large,” as described in Section 3.2.

For the purpose of comparing the various probability estimation models, we clip each visit at a randomly selected click, and generated the cumulative lift curve of Figure 3 based on probability estimates generated for the clipped visits. The reason for the clipping is that we are interested in predicting a purchase during a visit rather than at the end of the visit (by which time the purchase would have already taken place). The “Baseline” benchmark shows expected performance if we were to assign probability estimates to visits randomly.

We observe that the performance of the probability estimation model seems to improve with the order

of the model. The second-order model achieves higher lifts than the first-order model, which in turn outperforms the zero-order model. The higher order pruned model performs marginally better than the second-order model. The “Covariate” model improves little on the first-order model that does not include the covariate information. We note that the “Time” model, not included in the figure, gives results almost indistinguishable from the “One” model. Because it is conceptually simple and performs well, we use the estimates produced by the “Two” model in future steps of the analysis.

Table 2 shows the accuracy of the probability estimates produced by the “Two” model. Using a 14-bin discretization scheme, we allocate the visits of the validation data to bins based on their estimated purchase probabilities after various numbers of clicks. The entries in Table 2 show, for each such bin, the fraction of visits in that bin that actually result in a purchase. For instance, the entry corresponding to click 15 and estimated probability range “> 0.607” in Table 2 indicates that of the validation set visits for which the second-order Markov model’s purchase probability estimate at click 15 is greater than 0.607, 62.2% of them actually include a purchase.

We notice that for each column of Table 2, the observed purchase frequencies tend to increase in the estimated probabilities, indicating that our probability estimation methods are achieving some separation between purchase visits and non-purchase visits. However, they do not consistently match the estimated probabilities in magnitude.

Recall that when we use the dynamic programming formulation to determine dynamic decision rules, we require a representative purchase probability estimate  $P_t$  for each probability bin at each  $t$ . One method for generating this estimate for a given bin is to average the model-based probability estimates that fall into the bin. Another method is to use the purchase frequency observed for validation set visits in that bin. We have generated dynamic decision rules using both methods, and observed the latter to give slightly better results. Thus we use this method in the results of Section 4.5.

**4.5 Evaluation of Dynamic Classifiers.** We evaluate our classification schemes on a test set of approximately 31,000 visits. We build dynamic classifiers based on the probability estimates produced by the mixture of second-order Markov models and on the decision rules generated by the methods in Section 3.3. For the purpose of generating decision rules, we discretize the estimated purchase probabilities using 15 bins chosen based on percentiles in the full set of probability estimates in the validation data. We generate the dynamic decision rules using a planning horizon of  $H = 40$ , a horizon longer than the majority of visit sequences. (We

Pr. Est. \ t	1	3	5	10	15
> 0.607			0.635	0.649	0.622
0.368-0.607			0.477	0.462	0.458
0.223-0.368			0.236	0.263	0.309
0.135-0.223	0.194	0.181	0.172	0.209	0.181
0.082-0.135		0.099	0.089	0.108	0.106
0.050-0.082		0.048	0.061	0.058	0.078
0.030-0.050	0.031	0.042	0.040	0.051	0.057
0.018-0.030	0.022	0.026	0.029	0.033	0.045
0.011-0.018		0.022	0.026	0.025	0.030
0.007-0.011		0.010	0.013	0.018	0.031
0.004-0.007		0.002	0.006	0.024	0.021
0.003-0.004	0.005	0.007	0.006	0.009	0.016
0.001-0.003	0.001	0.002	0.003	0.006	0.016
< 0.001		0.003	0.004	0.007	0.008

Table 2: Comparison of observed purchase frequencies in the validation data set and purchase probabilities estimated using a mixture of second-order Markov models. Empty entries indicate few data are available for the corresponding bin.

have experimented with various choices of  $H$  and found that the models behave qualitatively similar for a broad range of choices.) Except where otherwise mentioned, we use cost structures that are constant in time. That is,  $c_{BB}(t) = c_{BB}$ ,  $c_{BN}(t) = c_{BN}$ , etc.

We refer to the dynamic classifiers that make use of the mixture of second-order Markov models and fixed threshold rules as “FT classifiers” and those that make use of the mixture of second-order Markov models and dynamic decision rules as “DDR classifiers.” For comparison, we also implement the following naive classification heuristic: we classify a visit as “buy” the first time we observe a page view in the “Checkout” section of the site. Otherwise, we classify the visit as “non-buy” at the  $N$ th click. This heuristic is motivated by the observation that a page view in the “Checkout” section of the site is a strong indicator of purchase. Since nearly all purchase visits must include a sequence of “Checkout” clicks, this simple heuristic is effective at detecting purchase visits, although not necessarily in a timely fashion. We observe that the performance of the naive heuristic depends substantially on the choice of  $N$ , thus we implement the heuristic with  $N = 20$  (“Naive20”) and with  $N = 40$  (“Naive40”).

An ideal dynamic classifier would detect a large number of “buy” visits, detect a large number of “non-buy” visits, make classifications quickly, and make few erroneous classifications. These various objectives can be traded off in numerous ways in the selection of decision rules, and thus evaluation and comparison of dynamic classifiers are difficult. In this section we explore a few of these tradeoffs and try to identify the relative merits of the various classification methods.

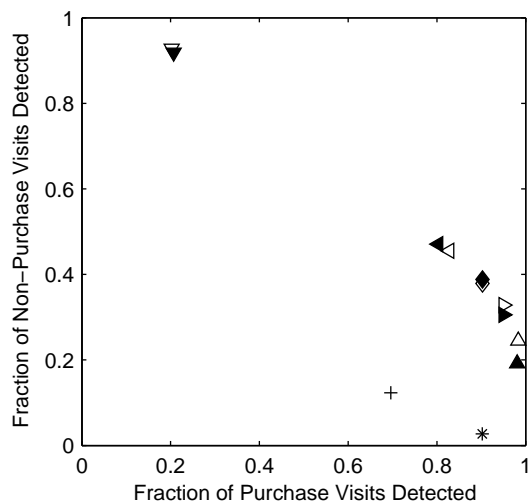


Figure 4: The tradeoff between detecting purchase visits and detecting non-purchase visits. Solid shapes represent FT classifiers, while hollow shapes represent DDR classifiers. “+” and “\*” represent the Naive20 and Naive40 classifiers respectively.

As we have mentioned, in our data set a page view in the “Checkout” section of the site is a good indication that the visit will result in a purchase. Thus our naive heuristic is somewhat effective at detecting purchase visits precisely and accurately. For instance, the “Naive40” heuristic detects 90% of the purchase visits in the test data set, while 48% of its “buy” classifications are correct. In contrast, among fixed threshold and dynamic decision rules tuned to detect 90% of the purchase visits, at most 38% of their “buy” classifications are correct. The disadvantages of the naive heuristic include its poor performance in detecting non-purchasing visits, and the fact that it cannot be adjusted to meet a wider range of objectives.

The FT and DDR classifiers, making use of our purchase probability estimates, are quite flexible. Figure 4 illustrates the tradeoff between the fraction of purchase visits detected and the fraction of non-purchase visits detected. The classifiers represented are those classifiers observed to achieve the highest percentage of non-purchase visits detected in the test data set, given that they detected approximately 20%, 80%, 90%, 95%, and 98% of the purchase visits respectively. In the case of the FT classifiers (indicated by the solid shapes), the decision rules were identified through exhaustive search over the space of fixed threshold rules possible in our probability discretization scheme. In the case of the DDR classifiers (indicated by the hollow shapes), the plotted points represent dynamic decision rules whose tuning parameters were chosen through a trial-and-error process. We also include the performance of the naive classifiers on the plot.

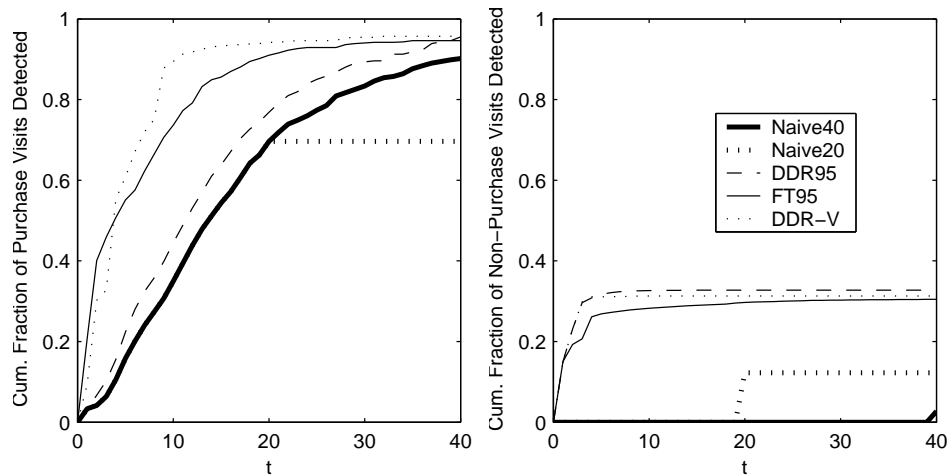


Figure 5: Performance of several classifiers, cumulative in discrete time.

We observe that not only can the FT and DDR classifiers be tuned to detect a larger number of purchase visits than the naive heuristics, they are considerably more adept at detecting non-purchase visits. We observe that the FT and DDR classifiers appear similar in this performance space, although the DDR classifier appears to have a slight edge in detecting non-purchase visits.

We would like not only to detect as many purchase and non-purchase visits as possible, but to do so in a timely fashion. We consider the FT and DDR classifiers in Figure 4 that detect 95% of purchase visits. We will refer to these classifiers as FT95 and DDR95 respectively. The classifiers’ performance detecting purchase visits and non-purchase visits, accumulated by  $t$ , is plotted in Figure 5. We also show the results from a DDR classifier (DDR-V) tuned using cost parameters that vary in time.

We observe that the FT95, DDR95, and DDR-V classifiers detect purchase visits sooner than the naive heuristics, while detecting considerably more non-purchase visits. FT95, for example, detects “buy” visits quite quickly. DR95 detects purchase visits more slowly, but it is better at detecting non-purchase visits. Its “buy” visit classifications are also more accurate, with 14% of its “buy” visit classifications correct versus only 7% for the FT95 model. The DDR-V classifier shows the versatility of the DDR method, as it detects purchase visits more quickly than even the FT95 classifier without sacrificing performance at detecting non-purchase visits. However, only 5% of its “buy” classifications are correct. We include the DDR-V classifier as an illustration of how different cost structures can be used to tune the classifiers for accurate and timely performance.

A final way we consider comparing the various dynamic classification methods is through a set of costs like those used in the derivation of the dynamic decision

rules in Section 3.3. If these represent real expected costs in a marketing system, then the total cost of a classifier measured on a test data set provides a useful assessment of the model. We have used various choices of cost parameters to choose fixed threshold rules and fit dynamic decision rules, then measured their performance on the test set using the same cost structures. While both the FT classifiers and DDR classifiers consistently outperform the naive heuristic, they give similar results to each other over a wide range of cost parameters. In the absence of a set of realistic cost parameters, this comparison method has been inconclusive in distinguishing the FT and DDR classifiers.

## 5 Conclusions and Future Research

We have posed the interesting and relevant problem of dynamic classification of customers in an online environment. Our specific interest is in predicting purchases at the visit level of web navigation sequences. Using Markovian models of web navigation, we have presented a methodology for incrementally mapping web navigation patterns to estimates of purchase probability. We have developed extensions of our models to account for second- and higher-order transition information and to incorporate covariates summarizing the customer’s past behavior and the elapsed time between page views. We have also developed time-sensitive methods for generating decision rules based on the purchase probability estimates. These decision rules include fixed thresholds and dynamic decision rules that we generate using dynamic programming.

We have illustrated our techniques on real web log data from a large computer retailer. Our probability estimation methods yield usable results, with second- and higher-order Markov navigation models yielding better lifts than simpler models based on low-order

Markov models. Our dynamic classification methods outperform a heuristic based on domain knowledge, and have the added advantage of being tunable so that we can identify models that balance the conflicting objectives of detecting purchase visits, detecting non-purchase visits, producing accurate classifications, and making classifications quickly.

We recognize several directions for further research. The first is to see if our methods perform well on similar data sets. It remains to be seen how well our methods apply to different types of web sites selling different types of items (e.g. non-durable goods).

In addition, we believe that our methods merit application to a wider range of data sets and classification problems. Although we developed a simple extension of our model to incorporate information on demographics and browsing histories, it would be interesting to examine more closely prediction on a user level rather than just on a visit level. It would also be interesting to consider classification targets other than the “buy”/“non-buy” classification we consider here. We are particularly interested in dynamic assignment of customers to multi-way targets such as clusters or marketing segments.

Finally, we believe that our methods mark a first step towards time- and cost-sensitive classification of customer sequences, where customer sequences are more broadly defined to include shopping cart activities, navigation information from multiple sites, and even interactions through other channels such as call centers.

## References

- [1] Agrawal, R. and Srikant, R., “Fast Algorithms for Mining Association Rules,” *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, Morgan Kaufman, 1994.
- [2] Anderson, C., Domingos, P., and Weld, D., “Relational Markov Models and their Application to Adaptive Web Navigation,” *Proceedings of The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2002)*, 143-152, 2002.
- [3] Bertsekas, D. *Dynamic Programming and Optimal Control*, Belmont, MA: Athena Scientific, 1995.
- [4] Cadez, I., Gaffney, S., and Smyth, P., “A General Probabilistic Framework for Clustering Individuals and Objects,” *Proceedings of The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2000)*, 140-149, 2000.
- [5] Cadez, I., Heckerman, D., Meek, C., Smyth, P., and White, S., “Visualization of Navigation Patterns on a Web Site Using Model-Based Clustering,” *Proceedings of The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2000)*, 280-284, 2000.
- [6] DeGroot, M., “Optimal Statistical Decisions,” New York: McGraw Hill, 1970.
- [7] Deshpande, M. and Karypis, G. “Selective Markov Models for Predicting Web-Page Accesses,” *Proceedings of the First SIAM International Conference on Data Mining*, 2001.
- [8] Jordan, M. and Jacobs, R., “Hierarchical Mixtures of Experts and the EM Algorithm,” *Neural Computation* 6, 181-214, 1994.
- [9] Kohavi, R. “Mining E-Commerce Data : The Good, the Bad, and the Ugly.” *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2001)*, 8-13, 2001.
- [10] Kohavi, R., Brodley, C.E., Frasca, B., Mason, L., and Zheng, Z. “KDD-Cup 2000 Organizer’s Report : Peeling the Onion,” *SIGKDD Explorations* 2 (2), 86-98, 2000.
- [11] Li, S., Montgomery, A.L., Srinivasan, K., and Liechty, J.C. “Predicting Online Purchase Conversion Using Web Path Analysis,” Working Paper, 2002.
- [12] Moe, W., and Fader, P., “Dynamic Conversion Behavior at e-Commerce Sites,” Working Paper, 2002.
- [13] Padmanabhan, B., Zheng, Z., and Kimbrough, S., “Personalization from Incomplete Data : What You Don’t Know Can Hurt,” *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2001)*, 154-163, 2001.
- [14] Rabiner, L., “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proceedings of the IEEE* 77(2), 257-286, 1989.
- [15] Sismeyro, C., and Bucklin, R., “Modeling Purchase Behavior at an E-Commerce Website: A Conditional Probability Approach,” Working Paper, 2002.
- [16] Smyth, P. “Clustering Sequences with Hidden Markov Models,” in *Advances in Neural Information Processing Systems* 9, eds. M. Mozer, M. Jordan, and T. Petsche, MIT Press, 1997.
- [17] Smyth, P. “Probabilistic Model-Based Clustering of Multivariate and Sequential Data,” in *Proceedings of the Seventh International Workshop in AI and Statistics*, eds. D. Heckerman and J. Whittaker, Morgan Kaufman, 1999.
- [18] Spiliopoulou, M., Srivastava, J., Kohavi, R., and Masand, B.M. “WEBKDD 2000 – Web Mining for E-Commerce,” *SIGKDD Explorations* 2 (2), 106-107, 2000.
- [19] Srivastava, J., Cooley, R., Deshpande, M., and Tan, P., “Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data,” *SIGKDD Explorations*, 1(2), 1-12, 2000.
- [20] Srivastava, J. and Ghosh, J., organizers, *Workshop on Web Mining (held in conjunction with First SIAM Conference on Data Mining)*, 2001.
- [21] Turney, P. Cost-sensitive learning bibliography. Institute for Information Technology, National Research Council, Ottawa, Canada. <http://extractor.iit.nrc.ca/bibliographies/cost-sensitive.html>