Targeted Advertising on the Web With Inventory Management

David Maxwell Chickering and David Heckerman

Microsoft Research

Redmond, WA 98052-6399

dmax@microsoft.com, heckerma@microsoft.com

Abstract

Companies that maintain Web sites can make considerable revenue by running advertisements, and consequently they compete to attract advertisers. The ability to deliver high click-through rates on a site can attract advertisers and under an appropriate pricing model can also increase revenue directly. Consequently, companies can benefit from delivery systems that display advertisements selectively to those visitors most likely to click though. To satisfy contractual obligations, however, these systems must simultaneously manage inventory. We developed a delivery system that maximizes click-through rate given inventory-management constraints in the form of advertisement quotas. The system uses predictive segments in conjunction with a linear program to perform the constrained optimization. Using a real Web site (msn.com), we demonstrated the efficacy of the system. We can generalize our system to find revenue-optimal advertisement schedules under a wide variety of pricing models.

(Marketing:Advertising and Media. Probability: Applications. Programming: Linear: Applications.)

Advertising revenue on the Web is important for many companies that host Web sites. The revenue can provide those companies with profits without their charging visitors for using their sites. According to the internet advertising revenue report (Interactive Advertising Bureau 2002), Internet-advertising revenues in the United States totaled $2.98 billion for the first six months of 2002. Hoffmann et al. (1997) discuss the business model of sponsored content sites.

In a typical advertising scenario, a hosting Web site or *publisher* will have a system in place for delivering advertisements so that a *visitor* (a person using a browser that is directed to the hosting site) asking to view a page will obtain the normal content from the Web site plus one or more advertisements provided by a third-party advertiser. A visitor can *click through* the advertisement (direct their browser to the link associated with the advertisement) by either clicking on the advertisement with the mouse or by some other means.

The method the advertiser follows to pay the publisher to display advertisements is called the pricing model. A number of pricing models for Web advertising have been developed that differ by the relative degree of risk that the publisher and the advertiser take (Hoffman and Novak 2000). At one extreme, using the *cost-per-thousand* or *CPM* pricing model, the advertiser pays a fixed amount to the publisher every time an advertisement is displayed. In this model, the advertiser assumes all of the risk; the publisher is guaranteed *per-impression* (per-page-view) revenue regardless of the effectiveness of the advertisement, whereas the advertiser benefits only if the advertisement is effective in influencing those customers who view it. In contrast, *performance-based* models move some or most of the risk to the publisher; under this general class of pricing models, the advertiser pays the Web site only when a customer takes some action in response to the advertisement. For example, the advertiser could pay (1) a fixed amount for every click, (2) a fixed amount for every purchase, or (3) a fixed percent of the purchase price for every purchase.

As described in the Internet advertising revenue report (Interactive Advertising Bureau 2002), roughly 40 percent of the 2002 second-quarter advertising revenue came from deals based on a so-called hybrid pricing model that combines CPM with a performance-based model. An example is a CPM deal that gives a bonus to the publisher for every click. Such a combination has the advantage of providing the publisher with guaranteed revenue and an incentive to place advertisements in a way that benefits the advertiser.

As Baudisch and Leopold (2000) discuss, many companies have turned to *targeting* to compete for advertising dollars; they employ advertisement-delivery systems that use information collected about the visitors to decide which advertisements to show. This information can include demographic information that the visitor has previously entered, it can include the set of pages previously visited on the publisher's site, or it can be simply the specific area of the publisher's site (the so-called content channel) that the user is navigating. For example, the advertisement-delivery system could serve sports-related advertisements on any sports-related page on the site.

Both the publisher and the advertiser can benefit from targeting, regardless of the pricing model. In a pure CPM pricing model, the publisher can implement differential (per impression) pricing based on the advertiser's desire to reach a particular type of visitor. For example, a tennis-clothing company advertising on a news site might be willing to pay a large amount for any advertisement shown in the sports section; the publisher can gain large per-impression revenue, while the advertiser pays to reach only those visitors of interest. In a hybrid pricing model or in a pure performance-based pricing model, the publisher can maximize revenue by showing advertisements based on expected revenue, and the advertiser can maximize revenue by adjusting per-click (or per-purchase or percent-of-purchase) payments.

Deals based on CPM or hybrid pricing models typically include *quotas* (minimum-impression guarantees). Furthermore, when they are targeting, publishers can face per-targeted-group quotas. That is, the publisher will promise to deliver a minimum number of impressions to visitors that interest the advertiser (for example, people reading a sports story). Given these quotas and given that a limited number of visitors come to the publisher's Web site during any period, the publisher will face inventory-management constraints: (1) a publisher must not oversell to advertisers, and (2) the delivery system must deliver all of the advertisements sold, regardless of the type of visitors that come to the site.

In the context of the CPM pricing model, Adler et al. (2002) consider the problem of allocating banner advertisements, given inventory-management constraints, when each impression is modeled as a constrained area into which varying numbers of banner advertisements of different sizes can be placed. Finding the revenue-optimal schedule, which Adler et al. (2002) show is NP-hard, is related to the well-known (and NP-hard) bin packing problem; Adler et al. (2002) provide a 2-approximation for the optimal solution. Amiri and Menon (2001) and Kumar et al. (2001) consider variations of this problem and alternative heuristic solutions.

We developed a system for serving advertisements to maximize overall number of clicks on a Web site; that is, our system maximizes revenue under a hybrid pricing model that gives the publisher a constant bonus for every advertisement clicked. We used information about the visitor in conjunction with a linear program to construct an advertisement-delivery system that maximizes expected number of clicks given the inventory-management constraints. We developed the system because the managers of a real-world Web site (msn.com) wanted to increase click-through rates on their advertisements; the pricing model at the time was strict CPM, but the managers believed that the advertisers would appreciate higher click-through rates.

The use of a linear program to solve a similar advertisement-delivery problem was developed independently by Langheinrich et al. (1999). Our system contains novel extensions to this approach as well as—to our knowledge—its first empirical validation in a real setting.

The Basic Approach

An important concept underlying our approach is the idea of an impression context. Intuitively, an impression context is the information about the impression (page view) that the targeting system uses to decide what advertisements to show. We can define impression contexts in a number of ways. An impression context may depend on the entire history of the visitor to that site who is about to receive the impression. Alternatively, an impression context may correspond to the area of the Web site on which the impression is being delivered. For example, a news site may have contexts corresponding to advertisements shown on front-page, finance, sports, entertainment, and weather sections of the site. An advantage to using this simple context is that we obtain some information about each user (that he or she is reading a story in a particular section of the site) without the need to track users across the site.

At the core of our approach is the use of predictive segments or clusters. We partition the impression contexts into a small number of segments, and then estimate for every advertisement in each segment the click-through probability (the probability that a visitor shown the advertisement in the segment will click through on the advertisement). We then use these individual click-through probabilities to target delivery so as to increase the overall click-through probability on the site. In the approach we discuss, we use the simple impression context corresponding to the area of the Web site on which the impression is being delivered.

Our approach consists of two phases. In the first phase, the system delivers each advertisement with probability proportional to its quota (and without regard to segment) and collects statistics about click through. In particular, for each advertisement and segment the system records (1) the number of times that advertisement was shown in the segment, and (2) the number of times that a visitor shown the advertisement in the segment clicks through. Using these counts, we estimate the click-through probability for each advertisement in each segment. We need to run the first phase only long enough to get accurate probability estimates. The greater the number of segments, the longer we must run the collection phase.

In the second phase of our approach, we use the estimated click-through probabilities to construct a new schedule that maximizes the expected overall click-through probability for the site. To describe this phase in more detail, we need some notation. Assume there are m segments and n advertisements.

We use p_{ij}, $i = 1 \ldots, n, j = 1 \ldots, m$, to denote the probability, estimated in the first phase, that a visitor will click on advertisement i shown in segment j. We define a particular delivery schedule by the set $\mathbf{X} = \cup_{i,j}\{x_{ij}\}$, where x_{ij} is the number of impressions of advertisement i to be shown in segment j in some specified amount of time T (for example, one day, one week, or one month).

Assuming that the click-through probabilities do not depend on the schedule, an observation we verify experimentally, we can express the expected overall click-through probability on the site, for any schedule, as

$$E(\text{overall click-through probability}) = \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij} \cdot \frac{x_{ij}}{N} \,. \tag{1}$$

where N is the total number of impressions to be delivered in time T.

Let q_i denote the *quota* for advertisement i—the minimum number of all impressions (in time T) of advertisement i to be shown. In addition, let s_j denote the *capacity* of segment j—the maximum number of all impressions (in time T) that could be shown in segment j. The quantities q_i are determined by the advertisers, whereas the quantities s_j are determined by the amount of traffic on the site. Also, the capacities are uncertain, because they are determined by future traffic. Nonetheless, at least for large sites and for optimization problems in which T is fairly small (less than a month), capacities are stable over time and can be estimated with little error.

For each advertisement i, the quota q_i imposes the following constraint on the delivery schedule:

$$\sum_{j=1}^{m} x_{ij} \geq q_i \,. \tag{2}$$

That is, the number of impressions in which advertisement i is delivered must be at least the number promised to the advertiser. Similarly, to avoid overbooking any section on the site, we have the constraint, for each segment j:

$$\sum_{i=1}^{n} x_{ij} \leq s_j \,. \tag{3}$$

We would like find the schedule $\mathbf{X} = \cup_{i,j}\{x_{ij}\}$ that maximizes Equation 1, subject to the inventory-management constraints expressed in Equations 2 and 3. Because of the enormous number of hits that typical Web sites receive per day, it is reasonable to treat each x_{ij} as a continuous variable. For example, using a time unit of a day, the average x_{ij} in our experiments was in the thousands; the difference in overall click-through probability between serving, say, 2,342.7 impressions of a particular advertisement per day to a segment versus serving 2,343 such impressions is insignificant. As a result—

because the objective function is a linear function of $\mathbf{X}$, and both constraints are linear functions of $\mathbf{X}$—we identify the optimal schedule using a linear program (Chvátal, 1983).

Once we have identified the optimal schedule $\mathbf{X}$, the delivery system must deliver x_{ij} impressions of advertisement i to segment j. A straightforward way to determine approximately the right number of each advertisement to show follows. When delivering an impression in segment j, we randomly choose to serve advertisement i with probability

$$\frac{x_{ij}}{\sum_{i'} x_{i'j}}.$$

With this approach, the system does not need to keep track of which advertisements it has already served. Furthermore, the random nature of the algorithm ensures that any particular visitor is likely to be shown a variety of advertisements.

Simple Extensions

Our basic approach has a potential problem: the solution to the linear program can be sensitive to small errors in the estimates of p_{ij}. For example, suppose that, for two different segments, the true click-through probabilities for a particular advertisement are identical and equal to 0.5. Even with a reasonably large sample, we are almost guaranteed to have two different estimates for the two probabilities. Suppose that one of the estimated probabilities is 0.501 and the other is 0.499. In this case, the linear program is likely to place all impressions of this advertisement in the segment with the higher probability. We would prefer a more uniform placement for two reasons. First, visitors in a given segment will get a greater variety of advertisements. The managers of msn.com, the site we studied, found this property highly desirable. Second, we expect this approach to yield higher overall click-through probabilities, because we avoid overfitting the training data by making the solution less sensitive to fluctuations in individual click-through probabilities.

Extending the basic linear-program approach, Tomlin (2000) solves this problem by optimizing a non-linear function of $\mathbf{X}$ that trades off expected overall click-through probability with the uniformity of the solution. Our approach is to bucket the p_{ij} values. In particular, we partition the p_{ij} values into sets or *buckets* of similar value and replace each p_{ij} with the mean of the bucket into which it falls. We then optimize the delivery of advertisements with these new click-through probabilities.

Given a desired number of buckets k, we use a simple agglomerative clustering algorithm to identify the buckets. Initially, we place each p_{ij} value in a separate bucket. Then, as long as we have more than k buckets, we merge the two buckets whose means are the closest. The best choice for k will depend on the domain and should be chosen empirically.

With bucketing added, there will likely be many optimal schedules because many click-through probabilities are equal. We break these ties by finding the most uniform of schedules among the optimal ones. That is, we first run the original linear program to identify an optimal schedule and note its expected overall click-through probability (C). Then, we define a second optimization that chooses an optimal schedule (which is likely to be different from the previously chosen optimal schedule) that is closest to the schedule in which each advertisement is shown the same number of times in each segment. In particular, we minimize the following objective function:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}|x_{ij} - \frac{q_i}{m}| \tag{4}$$

subject to constraints to be discussed. Recall that q_i is the number of impressions promised for advertisement i and that m is the total number of segments. Thus, if we place an equal number of impressions of a particular advertisement i' in each of the m segments, we will have $x_{i'j} = \frac{q_{i'}}{m}$ for all j. Term 4 measures a distance in impressions, for each advertisement, from this uniform configuration.

The constraints of this optimization problem include the constraints from the original problem (Equations 2 and 3) and the added constraint that the new schedule have the same (optimal) overall click-through probability as that identified by the first linear program. In particular, we include the constraint

$$\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij} \cdot \frac{x_{ij}}{N} = C \, .$$

The secondary optimization thus identifies the most uniform delivery schedule, subject to the inventory-management constraints and the constraint that the schedule must have the optimal overall expected click-through probability. It is well known that the second optimization, which involves absolute-value terms, can be solved by a linear program as well.

Earlier we stipulated that advertisements should be served uniformly across segments in the data-collection phase of the process so that we would have estimates of p_{ij} for all i and j. In fact, we do not need to estimate the probability p_{ij} if we do not plan to show advertisement i in segment j. For example, suppose a segment corresponds to impressions in the sports area of a Web site, and an advertiser makes a specific request *not* to show any of its advertisements in this segment. We can implement this request as the linear constraint $x_{ij} = 0$. Equivalently (and more efficiently) we can simply remove all instances of x_{ij} from the optimization.

We can add new advertisements dynamically to our system easily as long as the current schedule has not consumed the segments' capacity. In particular, we can collect data (phase one) for a new set of advertisements, while an existing (optimal) delivery schedule is in effect. After collecting these statistics, we can find a new optimal schedule that includes the new advertisements. Removing current advertisements from the schedule is even easier: we simply reoptimize with fewer advertisements using the p_{ij} values that are still relevant.

Experimental Results

We used our approach to deliver banner advertisements on the msn.com Web site. At the time, the msn.com site was organized in about 20 sections, with each section corresponding to a broad class of news stories. The site was running roughly 500 advertisements at a time. Also, the site and the advertisers scheduled the advertisements manually. In particular, the advertisers chose, for each advertisement and for each segment, a daily quota that did not violate the capacity constraints of the site. In our experiments, we used segments that corresponded to these Web-site sections. For example, if msn.com delivered an impression in the sports section, we labeled that impression as belonging to the sports segment.

In a preliminary experiment, we performed a passive test of our approach. That is, without implementing our schedule on the site, we estimated the improvement in overall click-through probability that would have resulted had we implemented the schedule. We simulated the rescheduling of advertisements across the entire site.

We collected statistics from the Web logs of December 21 and 22, 1998. Approximately 1.5 million impressions were delivered on each day. We used counts extracted from the December 21 logs to estimate the click probabilities and segment capacities. We estimated each probability using an average of 4,000 impressions. Then we ran the linear program to identify the schedule (with $T = 1$ day) that maximized the expected overall click-through probability. The linear program identified the optimal schedule in less than a minute on a Pentium II 200 MHz computer running the Windows NT 4.0 operating system.

We used counts extracted from the December 22 logs to estimate how well the resulting schedule would have worked. In particular, we used the data from the second day to reestimate each click-through probability p_{ij}, and then calculated the expected overall click-through probability for the optimized schedule via Equation 1 under the assumption that changes in schedule do not influence click-through probabilities. We compared this number to the actual overall click-through probability seen on the second day and found that our approach yielded an improvement of between 20 and 30 percent depending on the method used to estimate (smooth) the p_{ij} values. In addition, the schedule successfully avoided overbooking (because the expected and actual capacities were close) and thus fulfilled all quotas.

The managers of the msn.com site were quite pleased with these results and, in conjunction with a particular advertiser, authorized an active experiment. The advertiser had five advertisements running on msn.com, and was interested in how much we could increase the overall click-through probability on these advertisements. For this experiment, we estimated the click-through probabilities using statistics from the entire weekend of May 15, 1999. We estimated each probability using roughly 15,000 impressions. We partitioned these probabilities into 10 buckets. (We chose $k = 10$ buckets by repeating the passive experiment for many values of k using earlier data from the site.) Then, we used our approach to identify a uniform schedule with maximum overall click-through probability. Because the number of advertisements was small, the linear programs ran in under a second. Finally, we implemented this schedule during the following weekend of May 22 (Figure 1).

The overall click-through probability for the nontargeted advertisements did not change significantly between the two weekends. In contrast, our approach yielded a 30 percent increase for the targeted advertisements. As in the passive experiment, the schedule avoided overbooking. In addition, we found that the click-through probabilities p_{ij} were almost identical before and after the schedule change, thus validating one of the assumptions underlying our method.

As with the passive experiment, the managers of msn.com were quite pleased with these results. At their request, however, we do not report a net monetary gain from our approach.

Additional Extensions

There are several straightforward extensions to our approach. We can use our method to optimize any linear function of $\mathbf{X}$, not just the overall click-through probability. For example, we could add a constant α_{ij} to each term in Equation 1 that weighs the importance of showing the given advertisement. The site could then give preferential treatment to, for example, advertisers who pay more.

Assuming the data is available, it is straightforward to construct an appropriate (linear) objective function to maximize for almost any pricing model. For example, if we redefine each p_{ij} term from Equation 1 to denote the probability that an impression of advertisement i in segment j will result in a purchase, our approach can be applied directly to find the revenue-optimal schedule under a hybrid pricing model in which the publisher is paid a fixed bonus for every purchase. More generally, suppose that for each advertisement i and segment j, we can estimate the expected profit r_{ij} that will result from showing the advertisement in the segment. We can then use our process to maximize the total expected revenue across advertisers ($\sum_{ij} r_{ij} \cdot x_{ij}$) using the same inventory-management constraints used in the original formulation of the problem. We obtain a revenue-optimal schedule under a pure performance-based pricing model by further removing the quota constraints.

The schedule that maximizes the overall click-through probability across all advertisers may reduce the number of clicks for a particular advertiser. In another extension, we can explicitly prevent this from happening (in expectation) by adding the constraint that the expected net click-through probability for each particular advertiser be no less than this probability in the pretargeted schedule. As another example, we can include targeted branding in our system by allowing advertisers to require that a certain number of advertisement impressions remain in particular segments while allowing the remaining impressions to be optimized for click throughs.

Finally, DoubleClick (1996) shows that the click-through probability for an advertisement will depend (in a nonlinear way) on the number of times a user has seen that advertisement. In our approach, we do not model this effect. This omission likely does little harm in our msn.com application, because a user is unlikely to see the same advertisement more than once on this large site. Nonetheless, it would be interesting to extend our approach to include nonlinear optimization that could take these effects into account.

Adler, M., P. Gibbons, Y. Matias. 2002. Scheduling space-sharing for Internet advertising. *Journal of Scheduling*, **5** 103–119.

Amiri, A., S. Menon. 2001. Internet banner advertisement scheduling via Lagrangean decomposition. In *Proceedings of the Sixth INFORMS Conference on Information Systems and Technology (CIST 2001),* Miami Beach, FL. 133–140.

Baudisch, P, D. Leopold. 2000. Indifference, Dislike, Action: Web Advertising Involving Users. *Netnomics Journal*, **2** 75–83.

Chvátal, V. 1983. *Linear Programming.* W. H. Freeman and Company, New York.

DoubleClick. 1996. Frequency and banner burnout. DoubleClick research report.

Hoffman, D., T. Novak, P. Chatterjee. 1997. Commercial scenarios for the Web: Opportunities and challenges. R. Kalakota and A. Whinston, eds., *Readings in Electronic Commerce.* Addison-Wesley. 29–54.

Hoffman, D. L., T. P. Novak. 2000. Advertising pricing models for the World Wide Web. D. Hurley, B. Kahin, V. Varian, eds., *Internet*

Publishing and Beyond: The Economics of Digital Information and Intellectual Property, Cambridge, MA. MIT Press.

Interactive Advertising Bureau. 2002. IAB internet advertising revenue report: Second quarter results. `Http://www.iab.net/resources/ad_revenue.asp`.

Tomlin, J.A. 2000. An entropy approach to unintrusive targeted advertising on the web. In *Proceedings of the 9th International World Wide Web conference*. 767–774.

Kumar, S., V. S. Jacob, C. Sriskandarajah. 2001. Hybrid genetic algorithms for scheduling advertising on a Web page. *Proceedings of the Twenty-Second International Conference on Information Systems (ICIS),* New Orleans, LA. 461–468.

Langheinrich, M., A. Nakamura, N. Abe, T. Kamba, Y. Koseki. 1999. Unintrusive customization techniques for web advertising. *Proceedings of the 8th International World Wide Web Conference*. 181–194.

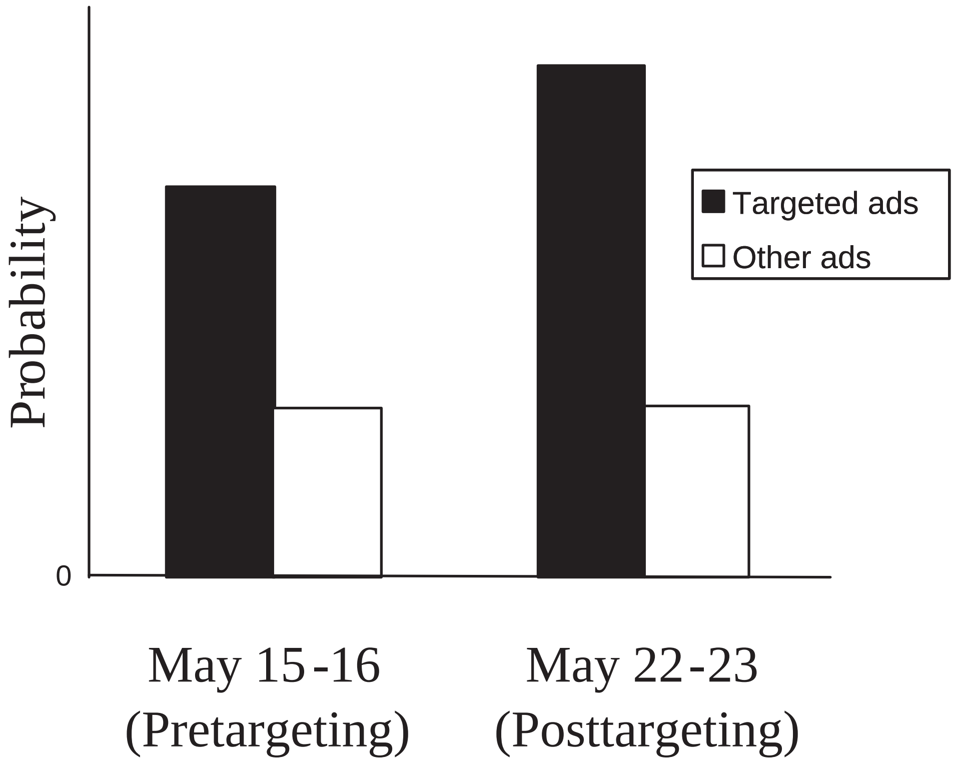

Figure 1: An active implementation of our system on msn.com showed the relative overall click-through probabilities of targeted and nontargeted advertisements during the weekends of May 15 to 16, 1999 and May 22 to 23, 1999. (At the request of msn.com, we do not show the absolute magnitudes of the overall click-through probabilities.)