To Show or Not Show: Using User Profiling to Manage Internet Advertisement Campaigns at Chitika

Radha Mookerjee, et al.

© 2012 INFORMS
To Show or Not Show: Using User Profiling to Manage Internet Advertisement Campaigns at Chitika

Radha Mookerjee  
Naveen Jindal School of Management, University of Texas at Dallas, Richardson, Texas 75080, radham@utdallas.edu

Subodha Kumar  
Mays Business School, Texas A&M University, College Station, Texas 77843, subodha@tamu.edu

Vijay S. Mookerjee  
Naveen Jindal School of Management, University of Texas at Dallas, Richardson, Texas 75080, vijaym@utdallas.edu

We study the problem of an Internet advertising firm that wishes to maximize advertisement (ad) revenue, subject to click-through rate restrictions imposed by the publisher who controls the website on which the ads are displayed. The problem is directly motivated by Chitika, an Internet advertising firm that operates in the Boston area. Chitika contracts with publishers to place relevant ads over a specified period, usually one month, on publisher websites. We develop a predictive model of a visitor clicking on a given ad. Using this prediction of the probability of a click, we develop a decision model that uses a varying threshold to decide whether or not to show an ad to the visitor. We vary the threshold depending on (1) the cumulative number of times an ad has been shown and (2) the cumulative number of clicks on the ad. The decision model’s objective is to maximize the advertising firm’s revenue subject to a click-through rate constraint. The implemented models work in real time in Chitika’s advertising network. We also discuss the implementation challenges and business impact.

Key words: Internet advertising; click-through rate; visitor profiling; revenue optimization.

The use of the Internet as a medium to deliver promotional material to prospective customers has grown tremendously in recent years (Fenton 2011, Hof 2011a). Despite the current economic slowdown, Internet advertisement (ad) revenue in the United States increased by 15 percent during 2009 to reach $26 billion in 2010 (Interactive Advertising Bureau 2011a). This trend is expected to continue; eMarketer (2011) estimated that these revenues would grow by over 20 percent in 2011 to pass the $30 billion mark, and would exceed $40 billion by 2013 and $50 billion by 2015. Another estimate predicts that by 2016 Internet advertising will reach $77 billion in 2016 and will constitute 35 percent of all advertising spending, overtaking television advertising (Hof 2011b). Finally, heavy Internet advertising is not purely a US phenomenon. In the United Kingdom, online advertising increased its share of the overall advertising market to a record 27 percent in the first half of 2011 (Fenton 2011).

The ability of advertisers to accurately quantify the success of their Internet ad campaigns has been one of the most important factors driving this growth (Monica 2005). When an ad is shown to a visitor on the website of an advertising publisher, the visitor’s click behavior can be tracked. This is important for the advertiser because a click usually signals the visitor’s interest in the ad content. Therefore, we should not be surprised to find that many ad revenue models used on the Internet are based on the cost-per-click model. In this model, the advertiser pays only for a click, and not for an impression (i.e., the event where a visitor is shown an ad). In the first half of 2011, performance-based pricing models (e.g., the cost-per-click model) accounted for 64 percent of the total online advertising spending in the United States, whereas the impression-based model accounted for only 31 percent (Interactive Advertising Bureau 2011b). Furthermore, the percentage of performance-based pricing models is increasing.

In addition to the advertiser and the publisher, an intermediary (i.e., an Internet advertising firm) usually matches an advertiser’s ads with a publisher’s
website and with a web user who visits the website. The problem we address in this study is from Chitika’s perspective. At first glance, the objectives of Chitika and a publisher appear to be completely aligned; both would like to show ad impressions to many visitors to maximize the number of clicks on that ad. However, this is often not the case. The publisher also wants to consider the number of impressions that generated a given number of clicks. To a publisher, an ad that is shown but not clicked on is a lost opportunity; a different ad could have been shown in that space. Other content, or even the same content displayed in a more attractive manner, could also have used this wasted space. Therefore, publishers typically hold advertising firms responsible for efficiently using their real estate. The publisher conveys this requirement to the advertising firm in the form of a click-through rate (CTR) constraint or a yield constraint. The CTR is the number of clicks generated divided by the number of impressions. The yield constraint is the advertising revenue to the publisher per impression. If the revenue per click to the publisher is constant, then mapping the yield constraint to a CTR constraint is easy. In practice, the publisher’s advertising revenue per click varies within a small range. However, in this paper we assume that the publisher’s revenue per click is constant; hence, we focus only on the CTR constraint.

This study extends previous work on the optimization of an advertising campaign (Hartl 1982). The key difference in a campaign that focuses on Internet use is the ability to collect information on web visitors and use this information to profile them. Past studies have extensively used such profiling to select the type of ad that should be shown to a visitor (Joshi et al. 2011, Mangalampalli et al. 2011). Previous literature has also focused on the optimal selection and placement of ads on a web page (Dawande et al. 2003; Kumar et al. 2006, 2007). Our study makes a significant contribution by its innovative use of the profiling of website visitors; it uses profiling to determine when not to show an ad to a visitor. Thus, although Internet advertising is often cited as being accountable from the perspective of the advertiser, our study takes this concept a step further by actively using click-stream data to manage an ad campaign in real time, not simply evaluating performance at the end of the campaign.

We organized this paper as follows. In the Problem Overview and Solution section, we present an overview of Chitika’s problem and our solution approach. In Solution Details, we provide analytical details of the solution. We formulate the problem of managing an ad campaign as an optimization problem, discuss four solution approaches, and compare the four approaches using real-world data collected from Chitika in Experiments. The Implementation Challenges and Impact section includes a discussion of the implementation challenges and the impact of the solutions on Chitika. In Lessons Learned, we summarize and conclude the paper.

Problem Overview and Solution

Founded in 2003, Chitika is an Internet advertising company. Since its inception, it has focused on building a large publisher network to enable advertisers to display their content on a diverse selection of websites. Chitika has more than 100,000 publishers spanning many countries; it also partners with large ad aggregators who provide it with a wide variety of ads. These partners collect revenue on a cost-per-click basis from merchants who wish to place ads. For example, if an ad has a per-click cost of $1, the merchant pays this amount to the advertising partner when a click occurs on the merchant’s ad. The partner retains a fraction of this dollar, usually about 15 cents. Chitika and the publisher share the remaining amount depending on the revenue-sharing contract between them. In addition to advertising partners, a small and growing number of merchants deal directly with Chitika to manage their ad campaigns. In such a case, the entire per-click cost flows to Chitika, which shares it in some contractual manner with the firm and the publisher.

Genesis of Problem and Challenges

In early 2010, in an effort to grow its publisher base and to attract highly visible publishers (e.g., the Wall Street Journal), Chitika launched Chitika Premium, a service that offers publishers an innovative value proposition—the publisher can control the average CTR of Chitika’s ads. Typically, this program requires a monthly contract. At the end of the month, Chitika pays the publisher a fraction of the revenue accrued from clicks that the ads generated. In addition, the
publisher enforces a CTR constraint on Chitika. Such a constraint ensures that the publisher’s space for ads on its website is used efficiently. If too many ads are placed but do not result in clicks, this indicates a wasted opportunity for the publisher. The publisher could either place more relevant ads or use the space for additional content. The CTR constraint essentially balances two opposing publisher goals: (1) generate as much advertising revenue as possible, and (2) keep the website content interesting so that visitors continue to patronize the website. If a publisher becomes greedy and shows an excessive number of ads, this greediness could result in sacrificing content, which is the main driver of website traffic. In the long run, the publisher would likely suffer. However, visitors who click on an ad are indicating that they like it. In such cases, the ad is almost indistinguishable from content.

This paper presents analytic models to optimize the Chitika Premium program. The problem has two aspects. First, we need to find a way to respect a publisher’s CTR constraint, while collecting as much revenue as possible for Chitika and the publisher. Thus, we need to optimize the campaign for a given CTR constraint. Second, we must find a convenient way to help the publisher choose an appropriate CTR constraint. Essentially, the publisher would like to reduce advertising clutter, but not lose too much revenue. For example, a publisher might want to impose the following rule via a CTR constraint on Chitika: reduce the number of impressions by 25 percent, but do not lose more than 5 percent of the revenue. The publisher needs a simple way to balance the loss of revenue with the reduction in advertising clutter. Therefore, for the Chitika Premium program, we must predict the publisher’s revenue for a given CTR constraint.

Solution Overview
The solution has two parts: (1) a step involving data analytics, and (2) a follow-up step involving decision analytics. The solution combines these steps and operates in real time (i.e., the delivery of ads on a website is managed in real time). We use data analytics to predict the likelihood that a visitor will click on a given ad. If this probability is low (i.e., below a threshold controlled by the decision analytic model), then the space allocated for the ad collapses and the ad is not displayed; otherwise, it is displayed. The challenge in the decision analytic model is to continuously vary the threshold so that the display criterion (i.e., the threshold) can be relaxed or tightened, depending on how much time remains in the planning horizon, which is typically one month, and whether the current CTR is above or below the target level that must be achieved at the end of the month. If we are ahead (i.e., the current CTR is above the target), we can lower the threshold and show more ads to generating more revenue. However, if we are below the target, then the threshold should be increased so that ads are shown only to more interested visitors. Increasing the threshold sacrifices advertising revenue; however, we need to take this action to achieve the target CTR at the end of the planning horizon.

Solution Details
Data Analytics
In this section, we describe how we estimate the probability of a click and the distribution of these probabilities in the visitor population.

Estimation of Click Probability Using Logit. This model uses a vector of observations collected from a cookie on the visitor’s computer and meta data available from the http header. These observations include variables (e.g., the visitor’s search string, Internet browser, operating system, and previous click data). The logit model combines information associated with the visitor with other information about the publisher’s website. Note that one of Chitika’s major strengths is its large publisher network; this allows Chitika to have repeated interactions with visitors across different websites, enabling it to develop a profile of each visitor. The predictive model uses this profile information to estimate the probability that a visitor will click on a given ad. We use logistic regression (logit) for this prediction. The regression output is an estimated probability that the user will click on the ad. The inputs are selected variables, which we found to be helpful in estimating the click probability. The model uses more than 50 variables as input (see Table 1).

Estimating the Click-Probability Distribution. Using logit regression, we estimate the click probability of any given site visitor. Figure 1 shows a representative frequency distribution of probabilities. The data
indicate that the probability increases rapidly from zero to a maximum frequency. After the maximum frequency, the frequency falls sharply. This is consistent with a beta distribution. We can estimate the click-probability distribution for a publisher using a sample of these probabilities. Using data collected for several publishers, we were able to confirm that the click probability follows a beta distribution. We calibrate the shape and scale parameters for this distribution to optimize the advertising campaign for a particular publisher. For example, the shape and scale parameters for data in Figure 1 are 2 and 0.005, respectively. We implemented a chi-square procedure to find the best fit shape and scale parameters for a given set of probability values obtained for a publisher.

The click-probability distribution provides us with vital information. We use the following rule: show an ad to a visitor only if the click probability \( p \) meets or exceeds a given threshold value, \( \alpha \). That is, show the ad only if \( p \geq \alpha \). For any given threshold, the click-probability distribution allows us to estimate the probability that an ad will be shown and the probability that a visitor who sees it will click on it. The appendix shows the mathematical expressions of these quantities. Conceptually, the probability that an ad will be shown is the upper tail of the beta distribution (i.e., the area under the density curve above \( \alpha \)), and the probability that a visitor will click on the ad is the conditional expectation of the upper tail of the distribution.

### Decision Analytics

**Model Formulation.** The objective is to maximize the number of clicks over the planning horizon. The value of the objective function depends on the decision variable \( \alpha \). This variable represents the threshold

![Histogram](https://via.placeholder.com/150)

**Figure 1:** We can approximate the click-probability distribution as a beta distribution.
used to decide whether or not to show an ad to a given visitor. The objective function is clearly decreasing in the value of \( \alpha \). That is, we can set \( \alpha \) to 0 to maximize clicks, because decreasing the value of \( \alpha \) leads to more impressions (i.e., we show ads to more visitors). A lower value of \( \alpha \) cannot reduce the number of clicks; this number will typically increase. Hence, in an unconstrained world, we should set \( \alpha \) to zero, implying that ads are shown to all visitors. However, we do not set \( \alpha \) to zero because we must consider the publisher’s CTR constraint, which we calculate as follows. At the end of the planning horizon, we calculate the CTR as the ratio of the total number of clicks to the total number of impressions. This rate must be greater than or equal to a specified fraction between 0 and 1—approximately 0.01 for a typical publisher. Because both the total clicks and the total impressions are functions of \( \alpha \), the CTR is also a function of \( \alpha \) at the end of the planning horizon. Figure 2 illustrates the overall solution process. A visitor is characterized by specific values of the predictor variables \( X \). Logistic regression then produces the estimated click probability \( p = L(X) \). The decision to show the ad depends on whether \( p \) is no lower than the threshold \( \alpha \).

**Solutions Considered.** The optimal value of \( \alpha \) is simply the lowest value that achieves, on an expected basis, the CTR constraint. We can approach the problem of optimizing \( \alpha \) over the planning horizon in one of two ways. The first is to use a static approach. Using this approach, we calculate the value of \( \alpha \) at the beginning of the horizon and hold it constant over the entire horizon. The static approach has less computational overhead; however, it has a drawback because it does not use real-world feedback on the actual number of impressions shown and clicks generated.

At Chitika, we implemented the second approach—a dynamic approach. Using this solution, we change the threshold each period (we will say more about how we choose the length of a period below). This approach copes well with the variability of incoming traffic and the clicking behavior of visitors to the website. We change the value of \( \alpha \) over the planning horizon; each time we change it, we account for the actual numbers of clicks and impressions that have occurred so far.

We explain the basic idea behind varying the thresholds as follows. Imagine that we are in the first period of a one-month planning horizon. We set the threshold at the smallest level such that the level of the target CTR should just be achieved at the end of the month, and we keep this threshold value constant for the current period. At the end of the period, we collect data on the numbers of ads displayed, impressions, and clicks. The current CTR is the number of clicks divided by the number of impressions. If this rate is higher than the required CTR at the end of the month, we can afford to lower the threshold and show

![Figure 2: We show an ad only if a visitor’s likelihood of clicking on it is sufficiently high.](image-url)
more ads. Conversely, if the current CTR is below the threshold, we must set the next period’s threshold slightly higher. At the beginning of each period, we set the threshold to a value that, if kept constant for the rest of the planning horizon, would just achieve the target threshold, on an expected basis, for the entire planning horizon.

An example will show how the dynamic approach works. To keep the example simple, let us set the planning horizon at three days and update the value of $\alpha$ daily. Let the publisher require us to achieve a CTR constraint of 0.01 (i.e., 1 percent CTR). At the beginning of the first day, we find the lowest value of $\alpha$ that, if held constant, would just achieve the required CTR of 0.01. Suppose this value is $\alpha_1$. We set the threshold to $\alpha_1$ for the first day. At the end of the first day, we make actual observations of the number of clicks ($r_1$) and the number of impressions ($m_1$); for example, $r_1 = 10$ and $m_1 = 2,000$. We next use these values, $r_1$ and $m_1$, to find the lowest value of $\alpha$ that, if held constant for the remaining two days, would just achieve the required CTR. Suppose this value is $\alpha_2$. Intuitively, this value should be higher than $\alpha_1$ because the CTR of 0.005 on the first day was below the required target of 0.01. After considering that we are below the constraint, we set the value of $\alpha$ to $\alpha_2$ for the second day. Similarly, at the beginning of the third day, we use the actual clicks and impressions that have occurred so far to calculate the lowest value of $\alpha$ needed to achieve the CTR constraint $\alpha_3$. On the third day, we use a threshold of $\alpha_3$. Figure 3 illustrates the difference between the static and dynamic approaches.

The dynamic approach is that it increases or reduces the threshold to account for actual feedback from the real world. If the threshold is higher than the constraint, we can afford to use a lower value of $\alpha$. Conversely, if real-world feedback tells us that the threshold is lower than the constraint, we need to tighten (increase) the threshold to catch up.

The key step in the above approaches, static or dynamic, is to calculate the expected CTR that will be achieved if a specific value of $\alpha$ is held constant for the planning horizon, or the remainder of this horizon. As the appendix shows, exactly evaluating the expected CTR can be time consuming. However, we need to evaluate it frequently if we use the dynamic approach. Because this approach outperforms the static approach, finding a fast and accurate approximation that will enable us to calculate $\alpha$ in real time becomes imperative. As we will see below, we often use an hourly updating scheme for $\alpha$. Thus, approximating the expected CTR becomes a crucial step in solving the problem in the real world. Figure 4 shows the four solution approaches that we explore.

The approximation works as follows. Rather than evaluating the constraint as the expectation of a ratio, we use a ratio of the expectations of the numerator and the denominator. The numerator is the number of clicks and the denominator is the number of impressions. The approximation is extremely accurate because the values converge to their respective means when the sample sizes, the number of impressions, and the number of clicks are large enough. Most publishers require only a few hours to generate sufficient data. Thus, we can replace all random quantities in the expression for the expected CTR by their means. This allows us to quickly calculate the expected CTR.
To provide an exact evaluation of this expected rate, we would need several hours, even on a fast computer. However, we cannot afford this luxury—we must update the value of $\alpha$ frequently, if possible every hour; thus, the approximation is a huge advantage in solving the company’s problem.

**Providing Decision Support to Publishers:**

**The Revenue Slider**

A publisher might not know how to set the CTR constraint. If the constraint is set too high, ads will be shown only to interested visitors; although the CTR will likely be high, the ad revenue will suffer. To help publishers balance the trade-off between ad revenue and clutter, Chitika provides publishers with a revenue slider. This slider allows a publisher to slide a button and observe the revenue impact of a candidate value of the CTR constraint. Each point along the slider represents different values of the CTR constraint; at each point, the slider shows the expected revenue that the publisher will generate at the end of a month if the current value of the CTR constraint is enforced. This revenue calculation uses the specific details of the publisher’s traffic and the revenue-sharing contract between Chitika and that publisher. Figure 5 illustrates the revenue output and the expected CTR from the slider for a specific publisher for different values of the CTR constraint. We see that the revenue is highest when the slider is at the extreme left, implying that the value of the CTR constraint is 0 and all visitors see ads. This setting will generate the highest number of clicks; however, the expected CTR at the end of the month could be lower than the rate the publisher desires. The revenue slider application also shows the expected CTR that results from a particular choice of the rate constraint. Beyond a certain point, meeting an arbitrary CTR constraint is impossible. Thus, we restrict the slider’s movement to between 0 and some high value, which is specific for that publisher.

**Experiments**

In this section, we discuss a series of experiments in which we compare the four solution approaches described above.

**Solution Selected**

The test data we used to compare the four approaches consist of 100,000 visits to a publisher’s website. To provide a valid comparison of the four approaches, we set the threshold to 0 for testing purposes. We then showed an ad for each of these 100,000 visits. For each

![Figure 5: The slider provides a prediction of the revenue and the CTR for any given CTR constraint.](image-url)
visit, we use the logit model to record the predicted probability of a click and to record if a click was generated. We simulated the advertising campaign for these 100,000 visits using each of the four solution approaches.

Table 2 shows a summary of the results. We vary the number of periods ($K$), the value of the CTR constraint ($\eta$), and the approach used to manage the ad campaign—dynamic-exact, static-exact, dynamic-approximate, or static-approximate. For each solution approach, we report the number of impressions shown ($m$), the number of clicks generated ($r$), and the associated CTR. For the dynamic approaches, we also report the CPU time (in seconds) needed to find the solution. An examination of the results in Table 2 reveals that the approximation works well; however, the time taken to find the optimal solution for the dynamic-exact approach grows quickly and is significant even for a modest number of periods (e.g., nine periods). Focusing on the approximate approaches, the results also show that the dynamic approach typically generates more clicks than the static approach; it also meets the CTR constraint. The static approach often overshoots the target CTR, resulting in a lower value for the number of generated clicks.

Table 3 provides more details on how a dynamic approach works. Here, we consider only the dynamic-approximate approach because the approximation works well for the size of problems that a typical Chitika publisher is likely to encounter. We illustrate the dynamic approach using a 30-day planning horizon and update the value of $\alpha$ daily. Seeing how a dynamic approach allows us to adjust the threshold value to accommodate actual events—impressions and clicks—is instructive. In the first period, the threshold chosen ($\alpha$) is always the same for the dynamic and static approaches. However, if good

<table>
<thead>
<tr>
<th>$K$</th>
<th>$\eta$</th>
<th>$m$</th>
<th>$r$</th>
<th>CTR</th>
<th>CPU secs</th>
<th>$m$</th>
<th>$r$</th>
<th>CTR</th>
<th>CPU Secs</th>
<th>$m$</th>
<th>$r$</th>
<th>CTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.005</td>
<td>17.377</td>
<td>166</td>
<td>0.009553</td>
<td>294</td>
<td>17.377</td>
<td>166</td>
<td>0.009553</td>
<td>0</td>
<td>16.438</td>
<td>165</td>
<td>0.010038</td>
</tr>
<tr>
<td>3</td>
<td>0.0075</td>
<td>14.207</td>
<td>158</td>
<td>0.011121</td>
<td>621</td>
<td>14.207</td>
<td>158</td>
<td>0.011121</td>
<td>0.01</td>
<td>7.142</td>
<td>132</td>
<td>0.018482</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>10.973</td>
<td>137</td>
<td>0.012485</td>
<td>889</td>
<td>10.973</td>
<td>137</td>
<td>0.012485</td>
<td>0.016</td>
<td>2.392</td>
<td>79</td>
<td>0.033027</td>
</tr>
<tr>
<td>3</td>
<td>0.0125</td>
<td>5.927</td>
<td>74</td>
<td>0.012485</td>
<td>1.276</td>
<td>5.927</td>
<td>74</td>
<td>0.012485</td>
<td>0.026</td>
<td>1.493</td>
<td>53</td>
<td>0.035499</td>
</tr>
<tr>
<td>3</td>
<td>0.015</td>
<td>4.681</td>
<td>65</td>
<td>0.013866</td>
<td>1.455</td>
<td>4.681</td>
<td>65</td>
<td>0.013866</td>
<td>0.032</td>
<td>867</td>
<td>41</td>
<td>0.04729</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>3.204</td>
<td>54</td>
<td>0.016854</td>
<td>4.934</td>
<td>3.204</td>
<td>54</td>
<td>0.016854</td>
<td>0.098</td>
<td>513</td>
<td>26</td>
<td>0.050682</td>
</tr>
<tr>
<td>6</td>
<td>0.005</td>
<td>35.135</td>
<td>325</td>
<td>0.0092</td>
<td>1.730</td>
<td>35.135</td>
<td>325</td>
<td>0.0092</td>
<td>0</td>
<td>32.905</td>
<td>321</td>
<td>0.009755</td>
</tr>
<tr>
<td>6</td>
<td>0.0075</td>
<td>31.174</td>
<td>316</td>
<td>0.010137</td>
<td>3.113</td>
<td>31.174</td>
<td>316</td>
<td>0.010137</td>
<td>0</td>
<td>14.210</td>
<td>252</td>
<td>0.017734</td>
</tr>
<tr>
<td>6</td>
<td>0.01</td>
<td>26.258</td>
<td>288</td>
<td>0.010968</td>
<td>4.967</td>
<td>26.258</td>
<td>288</td>
<td>0.010968</td>
<td>0</td>
<td>4.783</td>
<td>149</td>
<td>0.031152</td>
</tr>
<tr>
<td>6</td>
<td>0.0125</td>
<td>20.542</td>
<td>260</td>
<td>0.012657</td>
<td>6.035</td>
<td>20.542</td>
<td>260</td>
<td>0.012657</td>
<td>0.01</td>
<td>3.009</td>
<td>107</td>
<td>0.03556</td>
</tr>
<tr>
<td>6</td>
<td>0.015</td>
<td>14.181</td>
<td>226</td>
<td>0.015937</td>
<td>7.817</td>
<td>14.181</td>
<td>226</td>
<td>0.015937</td>
<td>0.015</td>
<td>1.754</td>
<td>78</td>
<td>0.04447</td>
</tr>
<tr>
<td>6</td>
<td>0.02</td>
<td>8.819</td>
<td>183</td>
<td>0.020751</td>
<td>14.035</td>
<td>8.819</td>
<td>183</td>
<td>0.020751</td>
<td>0.04</td>
<td>1.039</td>
<td>47</td>
<td>0.045236</td>
</tr>
<tr>
<td>9</td>
<td>0.005</td>
<td>52.854</td>
<td>486</td>
<td>0.009195</td>
<td>5.156</td>
<td>52.854</td>
<td>486</td>
<td>0.009195</td>
<td>0</td>
<td>49.335</td>
<td>483</td>
<td>0.00979</td>
</tr>
<tr>
<td>9</td>
<td>0.0075</td>
<td>48.299</td>
<td>477</td>
<td>0.009876</td>
<td>9.574</td>
<td>48.299</td>
<td>477</td>
<td>0.009876</td>
<td>0</td>
<td>21.509</td>
<td>384</td>
<td>0.017853</td>
</tr>
<tr>
<td>9</td>
<td>0.01</td>
<td>40.790</td>
<td>445</td>
<td>0.01091</td>
<td>14.170</td>
<td>40.790</td>
<td>445</td>
<td>0.01091</td>
<td>0.019</td>
<td>7.256</td>
<td>227</td>
<td>0.031284</td>
</tr>
<tr>
<td>9</td>
<td>0.0125</td>
<td>28.016</td>
<td>358</td>
<td>0.012778</td>
<td>18.650</td>
<td>28.016</td>
<td>358</td>
<td>0.012778</td>
<td>0.031</td>
<td>4.607</td>
<td>170</td>
<td>0.0369</td>
</tr>
<tr>
<td>9</td>
<td>0.015</td>
<td>21.097</td>
<td>326</td>
<td>0.015452</td>
<td>22.166</td>
<td>21.097</td>
<td>326</td>
<td>0.015452</td>
<td>0.047</td>
<td>2.716</td>
<td>125</td>
<td>0.046024</td>
</tr>
<tr>
<td>9</td>
<td>0.02</td>
<td>12.736</td>
<td>275</td>
<td>0.021592</td>
<td>29.033</td>
<td>12.736</td>
<td>275</td>
<td>0.021592</td>
<td>0.016</td>
<td>1.595</td>
<td>74</td>
<td>0.046395</td>
</tr>
</tbody>
</table>

Table 2: A comparison of the four solution approaches shows that using the dynamic-approximate approach is best.
<table>
<thead>
<tr>
<th>Day</th>
<th>Dynamic Alpha Impressions Clicks CTR</th>
<th>Static Alpha Impressions Clicks CTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0082 1,036 38 0.03668</td>
<td>0.0082 1,036 38 0.03668</td>
</tr>
<tr>
<td>2</td>
<td>0.006   3,850 68 0.017662</td>
<td>0.0082 2,226 58 0.026056</td>
</tr>
<tr>
<td>3</td>
<td>0.0059 5,874 119 0.020259</td>
<td>0.0082 3,090 91 0.02945</td>
</tr>
<tr>
<td>4</td>
<td>0.005   9,845 167 0.016963</td>
<td>0.0082 4,236 120 0.028329</td>
</tr>
<tr>
<td>5</td>
<td>0.0048 13,996 220 0.015719</td>
<td>0.0082 5,312 143 0.02692</td>
</tr>
<tr>
<td>6</td>
<td>0.0046 17,655 269 0.015236</td>
<td>0.0082 6,129 168 0.027411</td>
</tr>
<tr>
<td>7</td>
<td>0.0043 22,551 320 0.01419</td>
<td>0.0082 7,361 197 0.026763</td>
</tr>
<tr>
<td>8</td>
<td>0.0042 27,164 355 0.013069</td>
<td>0.0082 8,309 216 0.025996</td>
</tr>
<tr>
<td>9</td>
<td>0.0043 31,468 424 0.013474</td>
<td>0.0082 9,298 255 0.027425</td>
</tr>
<tr>
<td>10</td>
<td>0.0039 36,681 496 0.013522</td>
<td>0.0082 10,609 298 0.028089</td>
</tr>
<tr>
<td>11</td>
<td>0.0036 41,710 543 0.013018</td>
<td>0.0082 11,461 322 0.028095</td>
</tr>
<tr>
<td>12</td>
<td>0.0035 46,865 606 0.012931</td>
<td>0.0082 12,495 354 0.028331</td>
</tr>
<tr>
<td>13</td>
<td>0.0033 52,338 661 0.012629</td>
<td>0.0082 13,747 393 0.028588</td>
</tr>
<tr>
<td>14</td>
<td>0.0032 57,598 703 0.012205</td>
<td>0.0082 14,522 409 0.028164</td>
</tr>
<tr>
<td>15</td>
<td>0.0032 63,006 765 0.012142</td>
<td>0.0082 15,642 445 0.028449</td>
</tr>
<tr>
<td>16</td>
<td>0.003  68,626 810 0.011803</td>
<td>0.0082 16,762 465 0.027741</td>
</tr>
<tr>
<td>17</td>
<td>0.0031 73,944 863 0.011671</td>
<td>0.0082 17,581 498 0.028326</td>
</tr>
<tr>
<td>18</td>
<td>0.003  79,527 918 0.011543</td>
<td>0.0082 18,763 532 0.028354</td>
</tr>
<tr>
<td>19</td>
<td>0.0028 85,139 955 0.011193</td>
<td>0.0082 19,775 550 0.027813</td>
</tr>
<tr>
<td>20</td>
<td>0.003  90,515 1,021 0.01128</td>
<td>0.0082 20,731 585 0.028219</td>
</tr>
<tr>
<td>21</td>
<td>0.0026 96,285 1,070 0.011113</td>
<td>0.0082 21,938 608 0.027714</td>
</tr>
<tr>
<td>22</td>
<td>0.0026 101,945 1,132 0.011104</td>
<td>0.0082 22,837 636 0.02785</td>
</tr>
<tr>
<td>23</td>
<td>0.0023 107,689 1,178 0.010939</td>
<td>0.0082 23,798 657 0.027607</td>
</tr>
<tr>
<td>24</td>
<td>0.0023 113,529 1,228 0.010817</td>
<td>0.0082 25,054 687 0.027421</td>
</tr>
<tr>
<td>25</td>
<td>0.0022 119,316 1,271 0.010652</td>
<td>0.0082 25,891 709 0.027384</td>
</tr>
<tr>
<td>26</td>
<td>0.0022 125,127 1,351 0.010797</td>
<td>0.0082 26,994 752 0.027858</td>
</tr>
<tr>
<td>27</td>
<td>0.0013 131,053 1,421 0.010843</td>
<td>0.0082 28,237 789 0.027942</td>
</tr>
<tr>
<td>28</td>
<td>0.0006 137,011 1,477 0.010708</td>
<td>0.0082 29,061 817 0.028113</td>
</tr>
<tr>
<td>29</td>
<td>0.0001 143,008 1,534 0.010727</td>
<td>0.0082 30,230 853 0.028217</td>
</tr>
</tbody>
</table>
| 30  | 0.0001 149,005 1,589 0.010664     | 0.0082 31,299 882 0.02818           

Table 3: Although both approaches meet the CTR constraint, the dynamic approach produces more clicks than the static approach.

progress is made toward meeting the CTR constraint in the first period, a dynamic approach can risk lowering the threshold with the goal of generating more clicks. For example, we see that the threshold is lowered from its first-day value of 0.0082 to 0.006 for the second day. The static approach uses the same threshold (0.0082) for all 30 periods. In this problem, we set the value of the CTR constraint to 0.01. As we can see, although both approaches meet this constraint, the dynamic approach generates more clicks (1,589) than the static approach (882) because it varies the threshold across periods. We also observe that the static approach can overshoot the CTR constraint, resulting in missed revenue opportunities.

**Implementation Challenges and Impact**

Before we discuss implementation challenges, we must provide some background on Chitika’s existing advertising data flow and network architecture (see Figure 6).

**Advertising Data Flow and Architecture**

Currently, Chitika delivers ads using ad servers in five data centers located across the United States. Each data center handles traffic from a specific geographical region (e.g., Midwest, South, East). A geo-balancer placed at the entry to each network ensures that ad servers in the correct region are contacted for ad delivery. Once the ad request goes to a data center in a
Figure 6: Chitika's network architecture creates several implementation challenges for optimally managing an ad campaign.
particular geographical area, a load balancer within the data center ensures that no ad server in the data center becomes overloaded. Each data center houses several hundred ad servers; each server typically receives about 30–40 requests per second. These requests originate from a script that executes on the publisher’s web page at the time the page is being rendered for the incoming visitor.

All ad requests do not result in an ad display; an ad server must first decide whether or not to call the advertising partner. To make this decision, the ad server calculates the probability of a click by the specific visitor and compares this value to the publisher’s ad display threshold. If the click probability is higher than the threshold, the ad server requests that the partner supply an ad. Within approximately 200 milliseconds, the partner returns the ad as an ad unit, which is ready for rendering on the publisher’s page. Sometimes, an image is associated with the ad. To ensure fast processing in such cases, the ad server might obtain the image by calling an Akamai (www.Akamai.com) edge server, which stores the image. In some cases, the image might be available locally at the data center, making a call to an Akamai server unnecessary. After the ad has been displayed, a log of this impression is created. This log records detailed data about it, including time and information concerning the visitor. If the visitor clicks on the ad, the ad server logs this event. Note that the ad server that records the impression may differ from the one that records the click (although both are likely to be in the same data center). Each data center has a local log processor; once every hour, each ad server purges its logs to its local log processor, which then broadcasts its contents to a master log processor that collects the entire log for the past hour. At the end of each hour, it takes about 10–15 minutes to collect the log data from each data center to get a consolidated picture of the events that transpired during the hour. At the end of each hour, the raw log data is processed to match impressions with clicks. The entire process, starting from the call to Chitika by a script on the publisher’s page to the rendering of the ad on the page, typically takes less than 0.5 seconds. If the ad rendering takes more than a second, the impression is likely to be wasted (i.e., a click is unlikely to be generated for such a delayed ad).

Implementation Challenges
The architecture poses several implementation challenges. One of the main challenges is to decide how to implement the logic to calculate the value of the click probability to compare it against the threshold $\alpha$. Although the value of $\alpha$ stays constant for a period (e.g., a few hours), we need to calculate the value of the click probability ($p$) in real time for each visitor. Recall that we need to calculate the click probability using $p = L(X)$. This function is a polynomial, which we can quickly evaluate using our knowledge of $X$ for every visitor. Given the time constraints, a database lookup for $X$ is infeasible. Fortunately, we can extract the contents of the visitor’s cookie and other details from the http header and calculate the value of $p$ in about 50 milliseconds. Thus, the evaluation of the rule $p \geq \alpha$ is feasible, given the constraints of the problem. However, we need to implement the entire logic at the front end, implying that every ad server in Chitika’s network must replicate this logic. We must consider the time taken for replication in deciding how frequently to update the value of $\alpha$. Note that to find a new value of $\alpha$, we must know the actual events, impressions, and clicks that occurred in the last period. As mentioned above, because Chitika processes the logs of these events only once each hour, updating the value of $\alpha$ more than once an hour is infeasible. In reality, implementing even an hourly updating scheme may be impossible for some low-traffic publishers. As we noted earlier, the dynamic approach of frequently updating $\alpha$ is only possible if we approximate the expected CTR that results for any given choice of $\alpha$. This approximation is extremely fast and accurate, but with a proviso—the approximation requires that the sample sizes of the random quantities, specifically the number of impressions shown, are in thousands ($\geq 5,000$). This requires that the traffic to the publisher must be sufficiently large. The updating frequency must match the traffic of the publisher; an hourly updating scheme is possible for large publishers; however, for relatively low-traffic publishers, we must wait until a sufficient number of impressions have been shown before doing an update.

Benefits and Impact
The implementation, which began in March 2010, has resulted in a revenue increase of about $3,000
per day for Chitika. Based on the data collected between March 2010 and September 2010, we estimate the total increase in revenue to be approximately $1.2 million per year. This revenue increase occurred because Chitika was able to sign up more publishers under the Chitika Premium program. Over the past year, Chitika used its Premium program to partner with a large advertising aggregator to show ads in the United Kingdom. As part of the trial process, Chitika was asked to demonstrate a CTR of 0.015 (1.5 percent). Our methodology provided a CTR of 0.0151 (1.51 percent), and Chitika won the contract, which has contributed to a huge revenue increase for the company.

In December 2010, Chitika began offering another service, Chitika Select. Most of Chitika’s publishers had come on board to use Chitika Premium with the expectation that Chitika would show ads only to visitors coming to the site from search engines (i.e., search traffic). This was a good starting point. Although Chitika had ads for visitors who came to its website from other sources (e.g., by directly typing in the company’s URL), it chose not to show the ads to these visitors because doing so might dilute the CTR. Chitika Select offers publishers the opportunity to expand their ad usage, hence driving more revenue, with the assurance that the expanded coverage will not dilute the Premium CTR by more than 25 percent. Without a way to control the CTR dilution, the coverage expansion could seriously hurt the CTR and risk losing some publishers completely. However, with Chitika Select, Chitika was able to guarantee a specific CTR level, thus giving the publishers this option with assurance. The Select offering expanded the use of Chitika’s service across a large percentage of the network traffic. Whereas with Chitika Premium, Chitika accepted only search traffic and collapsed the ads for nonsearch traffic, the Select service allowed it to show ads to a much larger traffic base. As a result, Chitika generated an additional 25 percent in revenue.

**Potential Opportunities for Chitika**

**Inclusion of Advertiser Constraints.** Although this problem is similar to the one described above, its solution must also respect the performance constraints of both the publisher and the advertiser. In addition to the publisher’s constraint about exceeding a given CTR, the advertiser often poses an additional constraint, a cost-per-conversion constraint. The advertiser’s constraint requires that the cost-per-conversion value is below a specified constant. The cost-per-conversion value is the ratio of the total advertising cost, that is, the per-click cost times the number of clicks, divided by the number of conversions (e.g., a sale or registration) that are generated from the clicks. Once again, for a constant cost-per-click value, we can express the advertiser’s constraint as one that imposes a lower limit on the ratio of the number of conversions to the number of clicks (or conversion ratio). To solve this problem, we use the data analytics step to predict both the probability of a click and the probability of a conversion. The decision to show an ad depends on both probabilities. That is, we use two thresholds (one for the click probability and the other for the conversion probability), both of which must be satisfied to display an ad. Figure 7 depicts the overall solution process. Again, we vary these thresholds over the planning horizon, depending on current CTR and the current conversion rate. The inclusion of advertiser constraints often applies to situations in which Chitika contracts directly with the merchant for ad display, rather than obtaining these ads from a partner. Using the extended model, Chitika was able to win a contract to show ads for a partner who specializes in conversion-based payment schemes. This partner charges merchants on a cost-per-conversion basis, but pays Chitika on a per-click basis. However, to retain this business, Chitika must deliver a sufficient number of conversions to the merchant and must also keep the publisher satisfied by using its website space efficiently.

**Real-Time Media Buying.** To extend its access to publishers, Chitika is actively considering buying impressions at advertising exchanges. These exchanges allow advertising networks such as Chitika to place a bid to buy the rights to show ads to a particular visitor on a particular website, which is outside the advertising network’s current publisher network. Because of its large publisher network, Chitika has often previously encountered these visitors within its network. Therefore, it has visitor profiles on them, which it can leverage to target them with relevant ads. The main difference between the problem we address in this paper and the real-time media buying
problem is that in real-time bidding, Chitika pays for the impression whether or not a visitor clicks. Thus, instead of a CTR constraint, the cost of the impression becomes part of Chitika’s objective function. Chitika’s ability to bid in an informed fashion on impressions that are likely to yield a click has supported its entry into the media buying space. In this problem, the CTR constraint disappears, but a cost-per-impression value is included in the objective function. Thus, Chitika’s objective in the media buying case is to maximize profit—revenue from clicks minus cost of impressions—by choosing an optimal value of the threshold $\alpha$. If the advertiser is a direct client of Chitika, a conversion constraint might still be applicable in the real-time media buying problem. A huge market opportunity awaits the firm that is able to fine-tune and adapt the ideas in Chitika Premium and Chitika Select to media buying.

**Fading Ads.** The concept behind fading ads is to show an ad, but fade away after a specific amount of time (e.g., $\tau$), depending on the visitor’s previous click history. This is a relaxation of the model discussed above, in which either the time for which the ad is shown is zero (for the case $p < \alpha$) or the ad is shown for a fixed amount of time. This click history includes data on whether or not the visitor has clicked on ads and the time it took the visitor to generate these clicks. Such an innovation, if correctly implemented, would provide Chitika with a very distinctive product in the Internet advertising space.

**Lessons Learned**

In this study, we develop several approaches for using visitor profiling to manage an ongoing Internet advertising campaign for Chitika. The basic concept is to show ads only to those visitors who have a reasonable chance of generating a click—not to every visitor. Most publishers would like to maximize the revenue earned from ads via clicks, but not clutter their websites with many impressions that do not generate a click. Thus, Chitika, which does not directly suffer from unclicked impressions, must optimize the revenue it earns from an advertising campaign, but also respect a CTR constraint that the publisher supplies. We formulate this optimization problem and find the optimal value of the decision variable, a probability threshold that governs the display of ads, by using a simple numerical search. Because an ad campaign usually lasts several weeks, or even months, our process has the potential to allow a company to dynamically manage its campaign (i.e., by changing the probability threshold across periods.
during the planning horizon). Although the dynamic approach is attractive, it requires us to solve the problem in each period of the planning horizon; its use could raise concerns about the time required to solve the optimization problem. Motivated by the need to quickly solve the problem in each period, we created an approximation version of the dynamic approach. In this approach, we approximate the expected CTR by taking the ratio of expectations, rather than accurately computing this rate by taking the expectation of a ratio. Because we are dealing with large numbers of visitors (often thousands) per period, the approximation works extremely well and the optimization problem can be rapidly solved in each period, typically in a few milliseconds. Therefore, we are optimistic about the use of a dynamic approach to optimize ongoing ad campaigns. As we mention above, the benefit of a dynamic approach is that it can make the probability threshold more or less strict, depending on the progress made toward the final goal of achieving the CTR constraint at the end of the planning horizon.

Our paper represents a part of a current study with Chitika, an ad firm that is interested in using visitor profiling to optimize its ad placement activities. We can improve the dynamic approach presented in this study by considering the problem of optimizing a decision vector whose components are the probability thresholds for the rest of the planning horizon. Another level of sophistication in the optimization would be to solve the problem by characterizing the form of an optimal strategy that depends on the cumulative number of impressions and the cumulative number of clicks.

Appendix

Predictive Model

The value of the click probability for a given visitor to the publisher’s site is estimated using a logit model. This model uses a vector of observations collected from a cookie on the visitor’s computer and other data available from the http header. These observations include variables (e.g., the visitor’s search string, Internet browser, operating system, and previous click data). We use more than 50 predictor variables that combine information associated with the visitor with other information about the publisher’s website.

The logit model can be expressed as a function \( \hat{p} = L(X) \), where \( \hat{p} \) is the estimated click probability and \( X \) is the vector of variables used for prediction. To estimate the logit function, we use the logit estimation procedure in STATA.

### Symbol Definition

- \( K \): Number of periods in the planning horizon
- \( \lambda \): Number of visitors per period
- \( m_0 \): Initial number of impressions
- \( r_0 \): Initial number of clicks
- \( \bar{m} \): Random variable for the number of impressions in the planning horizon
- \( \bar{r} \): Random variable for the number of clicks in the planning horizon
- \( p \): Click probability for a visitor
- \( f(p) \): Density function for the click probability
- \( \alpha \): Click probability threshold (decision variable)
- \( \gamma(\alpha) \): Probability of impression for threshold \( \alpha \)
- \( \delta(\alpha) \): Probability of click for threshold \( \alpha \)
- \( \eta \): Publisher’s CTR constraint

### Table A.1: This table lists model parameters and variables.

<table>
<thead>
<tr>
<th>Table A.1: This table lists model parameters and variables.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a statistical and econometric software package, and obtain fitted coefficients in the regression model below:</td>
</tr>
<tr>
<td>( Z = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \cdots + \alpha_n X_n ).</td>
</tr>
</tbody>
</table>

In the above, the variable \( Z \) is either 0 or 1 and \( X_i \) is a predictor variable. To ensure that the values of the predicted probabilities lie between 0 and 1, we transform a predicted value \( \hat{Z} \) to a probability value between 0 and 1 as

\[ \hat{p} = \frac{e^{\hat{Z}}}{1 + e^{\hat{Z}}}. \]

### Optimization Model

Table A.1 summarizes the details and definitions of model parameters and variables. The question of interest in this study is to use the click probability \( p \) to decide if we should show an ad to a visitor. Let \( \alpha \) denote a probability threshold, a decision variable, such that the ad is shown only if \( p \geq \alpha \). Let \( \gamma(\alpha) \) and \( \delta(\alpha) \) denote, respectively, the probability of the event that \( p \geq \alpha \), and the event that the visitor clicks on an ad, given the probability threshold \( \alpha \). For a given value of the threshold \( \alpha \) and for a given period, we denote the random variable for the total number of clicks (respectively, total number of impressions) as \( R \) (respectively, \( \bar{m} \)).

Consider a planning horizon of \( K \) periods, with a uniform visitor arrival rate of \( \lambda \) per period. The objective is to maximize the number of clicks during a given planning horizon. With appropriate scaling, this objective amounts to maximizing the revenue accrued to an advertising firm. The expected number of clicks (\( E[R] \)) can be evaluated as

\[ E[R] = \sum_{i=0}^{K} \sum_{\bar{m}=0}^{i} i \Pr[R = j \mid \bar{m} = i] \Pr[\bar{m} = i]. \]

The probability that \( i \) impressions are shown in the planning horizon is binomially distributed and given by

\[ \Pr[\bar{m} = i] = \binom{\sum_{i=0}^{K} \sum_{\bar{m}=0}^{i} i \Pr[R = j \mid \bar{m} = i] \Pr[\bar{m} = i]}{i} \gamma(\alpha)^i (1 - \gamma(\alpha))^{K-i}. \]
In the above expression, \( \gamma(\alpha) \) is the probability that an ad is shown to a visitor. Denote \( f(p) \) as the probability density of \( p \), the probability of a click. Then, we can calculate this probability as
\[
\gamma(\alpha) = \int_a^1 f(p) dp.
\]

The conditional probability that \( j \) clicks are generated from \( i \) impressions is also binomially distributed and given by
\[
Pr[\hat{r} = j \mid \hat{m} = i] = \binom{i}{j} (\delta(\alpha))^j (1 - \delta(\alpha))^{i-j}.
\]

In the above, the probability that a visitor will click on an ad for a given threshold \( \alpha \) is given by
\[
\delta(\alpha) = \frac{\int_a^1 pf(p) dp}{\gamma(\alpha)}.
\]

**Properties of \( \gamma(\alpha) \) and \( \delta(\alpha) \).** From the expression for \( \gamma(\alpha) \), we see clearly that the probability of showing an impression decreases with the threshold \( \alpha \). We can also show that the probability of a click for a given value of \( \alpha \), \( \delta(\alpha) \), increases with \( \alpha \):
\[
\frac{d\delta(\alpha)}{d\alpha} = -\frac{\left(\int_a^1 f(p) dp\right)\alpha f(\alpha) + \int_a^1 pf(p) dp}{\gamma(\alpha)^2}.
\]

The above expression is strictly positive because \( \int_a^1 \alpha f(p) dp < \int_a^1 pf(p) dp \). Thus, the probability of a click increases with \( \alpha \).

**Problem P.** Problem P can be described as one of maximizing the number of clicks in a given planning horizon, subject to an expected CTR constraint. The decision variable is the probability threshold \( \alpha \). Using the above definitions, we can formally express the optimization problem as
\[
\max_{\alpha} \mathbb{E}[\hat{r}], \quad \text{subject to } \mathbb{E}\left[\frac{r_0 + \hat{r}}{m_0 + \hat{m}}\right] \geq \eta.
\]

In the above problem, we denote the initial number of clicks by \( r_0 \) and the initial number of impressions by \( m_0 \). At the beginning of the planning horizon, we set \( r_0 = m_0 = 0 \). However, at any other stage of the planning horizon, these values could be positive, based on actual observations. If we are solving problem P at an intermediate stage (e.g., period \( j \)), we replace the value of \( K \) by \( K - j \).

**Solution to Problem P.** To solve problem P, we show that the objective function is decreasing in the threshold \( \alpha \). To see why this is so, note that in the expression for \( \mathbb{E}[\hat{r}] \), the term \( Pr[\hat{m} = i] \) is independent of \( j \). Therefore, the inner summation equals the expectation of a binomial distribution with \( i \) trails and a success probability of \( \delta(\alpha) \), or \( \delta^i(\alpha) \). Therefore,
\[
\mathbb{E}[\hat{r}] = \delta(\alpha)^i \sum_{i=0}^{K} \binom{K}{i} (\gamma(\alpha))^i (1 - \gamma(\alpha))^{K-i} = K\lambda\delta(\alpha)\gamma(\alpha).
\]

Thus, in problem P, maximizing \( \int_0^1 pf(p) dp \), which is decreasing in \( \alpha \), is sufficient. Therefore, the optimal solution to problem P is the smallest value of \( \alpha \) that meets the CTR constraint.

**Four Solution Approaches.** We discuss four solution approaches that are based on whether the optimization is static or dynamic and whether the calculation of the expected CTR is exact or approximate. The four solution approaches are static-exact, static-approximate, dynamic-exact and dynamic-approximate. In static optimization, we use the same value of the threshold (\( \alpha \)) for the entire planning horizon. At the beginning of the planning horizon, the values of the number of impressions and clicks are set to 0, and the problem is solved once for the entire horizon to yield an optimal value of the threshold. In dynamic optimization, we solve the problem in each period for the rest of the planning horizon. Here, values of the number of impressions and the number of clicks are set to actual (i.e., observed) values. Thus, the threshold can be different for each period in the planning horizon. In a solution approach, we also must look at whether the calculation of the expected CTR is exact or approximate. In an approximate approach, we calculate the expected CTR by replacing the random variables for the number of clicks and the number of impressions by their expected quantities:
\[
\mathbb{E}\left[\frac{r_0 + \hat{r}}{m_0 + \hat{m}}\right]_{\text{approx}} = \frac{r_0 + K\lambda\delta(\alpha)\gamma(\alpha)}{m_0 + K\lambda\gamma(\alpha)}.
\]

In the above approximation, \( k \) is the number of periods left in the planning horizon, \( k \in \{1, 2, 3, \ldots, K\} \). The approximation technique rapidly yields results and can be useful when the value of \( K\alpha \) is high (e.g., in the hundreds of thousands) and the threshold must be calculated quickly to manage an ongoing campaign. The approximation is also useful because a closed-form expression is not available for the optimal value of \( \alpha \). Instead, the optimal must be found using a numerical search procedure that begins with a very low value of \( \alpha \) and increments this value until the CTR constraint is first satisfied.

**References**


