INTERFACES
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0092-2102 | EISSN 1526-551X | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

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, TX 75080, radham@utdallas.edu

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

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

We study the problem of an internet advertising firm that wishes to maximize ad revenue subject to a click-through-rate constraint imposed by the publisher (i.e., the website where the ads are displayed). The problem is directly motivated by the internet advertising firm Chitika (`www.chitika.com`) operating in the Boston area. The internet advertising firm (ad firm, for short) contracts with a publisher to place relevant ads over a specified time period (usually one month) on the publisher's web site. We develop a predictive model of a visitor's click event for 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 a given visitor. We vary the threshold depending on: (1) the cumulative number of ads shown (number of impressions) so far, and (2) the cumulative number of clicks so far. The decision model has the objective of maximizing the ad firm's revenue subject to a *ctr* constraint. We have implemented the above models that operate in real-time in Chitika's ad network. We discuss implementation challenges and the business impact of these models.

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

Introduction

In recent times, the use of the internet as a medium to deliver promotional material to prospective customers has grown tremendously (Fenton 2011, Hof 2011a). Despite the current economic slowdown, internet advertising revenues in the United States increased by 15 percent over 2009 to reach \$26 billion in 2010 (Interactive Advertising Bureau 2011a). This trend is expected to continue: eMarketer (2011) estimates internet ad revenue to grow by over 20 percent in 2011 to pass the \$30 billion mark, and exceed \$40 billion by 2013 and \$50 billion by 2015. Another estimate shows that internet advertising will reach \$77 billion in 2016, and will comprise 35 percent of all advertising spending, overtaking television advertising (Hof 2011b). Finally, internet advertising is not purely a U.S. 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 the advertiser to accurately quantify the success of an internet ad campaign has been one of the most important factors for this growth (Monica 2005). When an ad is shown to a web visitor on a particular web site (or the ad *publisher's* site), it is possible to track the visitor's click behavior. This is important for the advertiser since clicking on an ad is usually a good signal of a visitor's interest in the ad material. It is therefore not surprising 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* (the event where a visitor is shown an ad). In the first half of 2011, 64 percent of the total online advertising spending in the United States was based on performance based pricing models (such as the cost-per-click model), whereas the impression based model accounted for only 31 percent (Interactive Advertising Bureau 2011b). Furthermore, the percentage share of performance based pricing models is increasing.

In addition to the advertiser and the publisher, there is usually an intermediary, namely an internet advertising firm that matches an advertiser's ads with a publisher's site and with a web user visiting the site. Our problem in this study is from the perspective of the internet advertising firm (or ad firm), `www.chitika.com`. At first glance, it may appear that the incentives of Chitika and the publisher should be completely aligned; both would like to show as many ads to as many visitors so as to maximize the number of clicks. However, this is often not the case. From the perspective of the publisher, it is also important to consider the number of impressions that generated a given number of clicks. For a publisher, an ad that is shown but not clicked on, is a lost opportunity; another ad could use the space where this ad was shown. Additional content (or even the same content displayed in a more attractive manner) could also use this wasted space. It is therefore typical for the publisher to hold the ad firm responsible for using its real-estate efficiently. The publisher conveys this requirement to the ad firm in the form of a click-through-rate constraint or a yield constraint. The click-through-rate is the number of clicks generated divided by the number of impressions (ads shown). The yield constraint is the ad revenue to the publisher per impression. If the revenue per click to the publisher is constant, then it is easy to map the yield constraint to a click-through-rate constraint. For this study, we will assume that the publisher's ad revenue per click is constant; hence we will only focus on a click-through-rate constraint for the remainder of this paper.

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

The rest of the paper is organized as follows. We first present an overview of Chitika's problem and the solution approach. Next, we provide some analytical details of the solution. We formulate the problem of managing an ad campaign as an optimization problem and discuss four solution approaches for the same. We next present a comparison of the four approaches using real-world data collected from Chitika. The following section (Implementation Challenges and Impact) provides a discussion of the implementation challenges and the impact of the solution on Chitika. The last section summarizes and concludes the paper.

Overview of Problem and Solution

Founded in 2003, Chitika (`www.chitika.com`) is an internet ad company. Since its inception, it has focused on building a large publisher network to enable advertisers to display ads on a diverse selection of websites. Currently, Chitika has more than 100,000 publishers spanning many different countries. For its ads, Chitika partners with large ad aggregators that provide Chitika 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 cost-per-click of 1 dollar, then the merchant pays this amount to the ad partner when a click occurs on its ad. The ad partner retains a fraction of this dollar (usually, about 15 cents). Chitika and the publisher share the remaining amount (i.e, 85 cents) depending on the revenue sharing contract between Chitika and the publisher. In addition to ad partners, there is a small, but growing number of merchants who directly deal with Chitika for managing their ad campaigns. In this case, the entire cost-per-click amount flows to Chitika that shares it in some contractual manner between 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. Wall Street Journal), Chitika launched Chitika Premium, a new service offering publishers an innovative value proposition. In Chitika Premium, a publisher can control the average click-through-rate of Chitika's ads. Typically, this program offers a monthly contract. At the end of the month, Chitika pays a publisher (that chooses Chitika Premium) a fraction of revenue accrued from clicks that were generated on the ads. In addition, the publisher enforces a click-through-rate constraint on Chitika. Such a constraint ensures that the publisher's space for ads on its website is used efficiently. If too many ads placed do not result in clicks, this is an indication of a wasted opportunity for the publisher. Either the publisher could place more relevant ads, or use the space for additional content. The click-through-rate constraint essentially balances two opposing goals of the publisher: (1) generate as much revenue as possible from ads, and (2) keep the website content interesting so that visitors continue to patronize the website. If a publisher becomes too greedy and shows an excessive number of ads, this could sacrifice content that is the main driver of traffic to the website. In the long run, the publisher would likely suffer. On the other hand, if a visitor clicks on an ad, it indicates that she likes the ad. In such cases, the ad is almost indistinguishable from content.

This paper presents analytic models to optimize Chitika's Premium program. There are two aspects to the problem. We first need to find a way to respect a publisher's click-through-rate constraint while collecting as much revenue as possible for Chitika and the publisher. Thus in the first part, we need to optimize the ad campaign for a given click-through-rate constraint. For the second part, we need to find a convenient way to help the publisher choose an appropriate click-through-rate constraint. Essentially, the publisher would like to reduce ad clutter but not lose too much revenue. For example, a publisher may want to impose the following rule via a click-through-rate constraint on Chitika: reduce the number of impressions by 25%, but not lose more than 5% of the revenue. Thus the publisher needs a simple way to balance the loss of revenue with the reduction in ad clutter. Therefore, as part of the Chitika Premium program, we need to predict the publisher's revenue for a given click-through-rate constraint.

Overview of Solution

The solution consists of two parts: (1) a step involving data analytics, and (2) a follow-up step involving decision analytics. The solution combines the above steps and operates in real-time, i.e., the delivery of ads on the website is managed in real-time. We use data analytics to predict how likely it is for a visitor to click on a given ad. If this probability is low (i.e., below a threshold, controlled by the decision analytic model), then the ad-unit collapses and no ad is displayed. Otherwise, the ad is displayed. The challenge in the decision analytic model is to continuously vary the threshold so that the display criterion (the threshold) can be relaxed or tightened, depending on how much time is left in the planning horizon (typically, this horizon is one month) and whether the current click-through-rate is above or below the target level that has to be achieved at the end of the month. If we are ahead (i.e., the current click-through-rate is above the target), the threshold can be lowered (more ads can be shown) so that more ad revenue can be earned. On the other hand, if we are behind (i.e., the current click-through-rate is below the threshold), then the threshold should be increased (so that ads are shown to more interested visitors). Increasing the threshold sacrifices ad revenue, but we need to take this action to achieve the target click-through-rate at the end of the planning horizon.

Details of Solution

Data Analytics

Here we describe how the probability of a click and the distribution of these probabilities in the visitor population is estimated.

Estimation of Click Probability Using Logit This model uses a vector of observations collected from the visitor's cookie as well as meta data available from the `http` header. These observations include variables such as the visitor's search string, internet browser, operating system, previous click data, and so on. The Logit model combines information associated with the visitor with other information about the publisher's website. Note that one of the major strengths of Chitika is its large publisher network. Having a large network enables the firm to have repeated interactions with visitors across different web sites, thus enabling the firm to develop a *profile* of each visitor. The predictive model uses this profile information to estimate the chances that a visitor will click on a given ad. We use Logistic regression (Logit) for this predictive task. The Logit model can be expressed as a function $\hat{p} = L(\mathbf{X})$, where $\hat{p}$ is the estimated click probability, and $\mathbf{X}$ is the vector of variables used for the prediction. The model uses over 50 different variables; some of the important ones are listed in Table 1. The Logit function, $L(\mathbf{X})$, is a polynomial function that provides coefficients for each of the significant variables that predict a user's likelihood of clicking on an ad.

Estimating the Click-probability Distribution Using the above Logit model, the value of p for any given visitor to the publisher's site is estimated. Figure 1 shows a frequency plot of this data. The data indicates that the probability increases rapidly from zero to a maximum frequency. After the maximum frequency, the frequency sharply reduces. 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 different publishers, we were able to estimate that the click-probability follows a *Beta* distribution. We calibrate the shape and scale parameters for this distribution to optimize the ad campaign for a particular publisher. For example, the

Variable	**Description**	**Possible Values**
operating_system	Which operating system is being used by the visitor?	Linux, Android, Mac OS, Microsoft Windows, etc.
browser	Which browser is being used by the visitor?	Internet Explorer, Firefox, Chrome, Safari, Opera, etc.
search_engine	Which search engine is used by the visitor?	Google, Yahoo, Bing, etc.
btnSearch	Did the user click on the search engine button?	Yes, No
bad_speller	Is the visitor a bad speller?	Yes, No
search-string_type	Does the search string have local intent?	Yes, No
token_interest	Total number of clicks by this visitor in the past for ads on similar search strings	Integer
domain_token_interest	Total number of clicks by all the visitors in the past for ads on similar search strings	Integer
day	Day of the visit	Monday, Tuesday, . . . , Sunday
time	Time of the visit	Morning, Afternoon, Evening, Night
height	Height of the ad unit	Numeric
width	Width of the ad unit	Numeric
loc_x	x-location of the ad	Numeric
loc_y	y-location of the ad	Numeric
user_clicks	Total number of clicks by this visitor in the past	Integer
user_imps	Total number of impressions for this visitor in the past	Integer
CLICK	**Dependent Variable** Did the visitor click on the ad shown?	Yes, No

Table 1 There are more than 50 variables used in the Logit model; some of the important ones are shown here.

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 employ the following rule: Show an ad to a visitor only if the click-probability (p) meets or exceeds a given threshold value (say, α). That is, show the ad only if

$$p \geq \alpha.$$

For any given threshold, the click-probability distribution allows us to estimate the probability

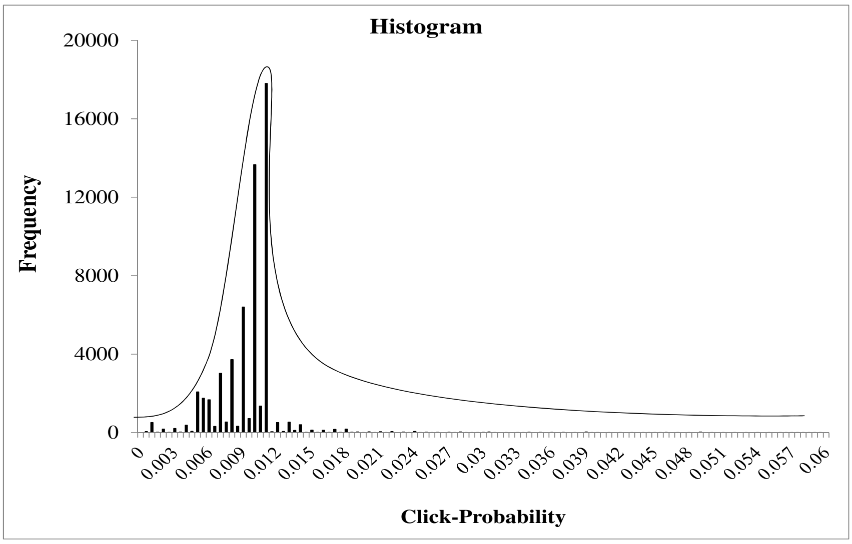

Figure 1 **We can approximate the click-probability distribution as a Beta distribution.**

that an ad will be shown and the probability that a visitor who is shown ads will generate a click. The mathematical expressions of these quantities are provided in the Appendix. 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 α) and the click-probability that a visitor will click on an ad is the conditional expectation of the upper tail of the distribution.

Decision Analytics

Model Formulation The Objective Function is to maximize the number of clicks over the planning horizon (usually 1 month). The value of the objective function depends on the decision variable α . This variable represents the threshold used to decide whether or not to show an ad to a given visitor. The objective function is clearly decreasing in the value of α. That is, we can set α to zero to maximize clicks. This is because decreasing the value of α leads to more impressions (i.e., we show ads to more visitors). Hence, a lower value of α cannot reduce the number of clicks; typically the clicks will increase. Hence, in an unconstrained world, we should set α to zero, implying that all visitors are shown ads. The reason that α is not set to zero is the presence of the publisher's click-through-rate constraint. We calculate this constraint as follows. At the end of the planning horizon, we calculate the click-through-rate as the ratio of the total number of clicks to the total number of impressions. This click-through-rate must be greater than or equal to a specified fraction (between 0 and 1, around 0.01 for a typical publisher). Since both the total clicks and the total impressions are functions of α, the click-through-rate at the end of the horizon is also a function

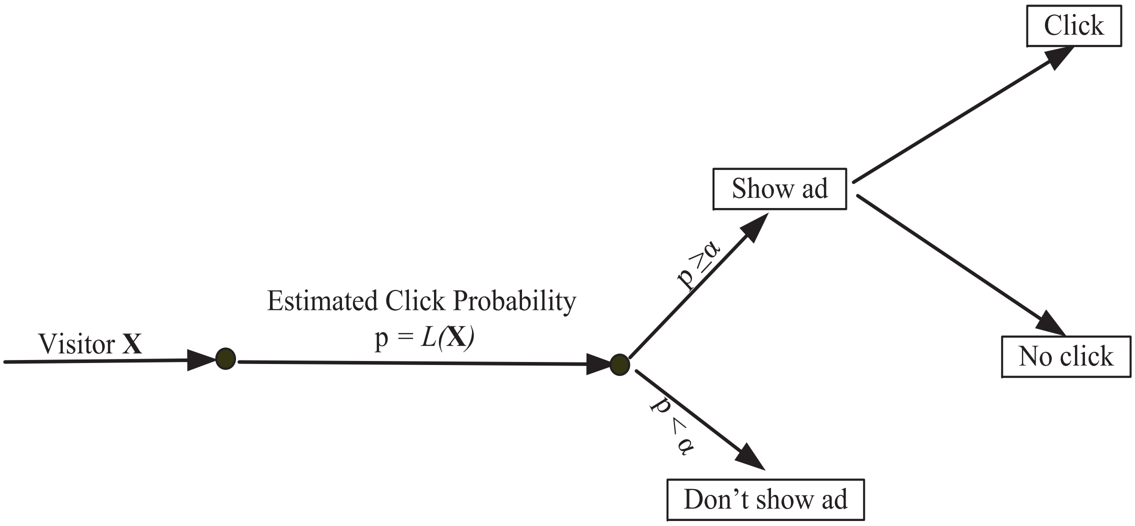

Figure 2 **We show the ad only if the visitor's likelihood of clicking on it is sufficiently high.**

of α. Figure 2 illustrates the overall solution process.

Solutions Considered The optimal value of α is simply the lowest value that achieves, on an expected basis, the click-through-rate constraint. There are two ways of approaching the problem of optimizing α over the planning horizon. One way is to use a static approach. In such an approach, we calculate the value of α at the beginning of the horizon and hold it constant over the entire horizon. Such a static approach has the benefit of less computational overhead but it suffers from the drawback that it does not use real-world feedback on the actual number of impressions shown and clicks generated.

In the company, we have implemented a solution where we change the threshold every period (later, we will say more about how we choose the length of a period). Such a (dynamic) approach copes well with the variability of incoming traffic and the clicking behavior of visitors that come to the website. In a dynamic approach, we change the value of α over the planning horizon. At each step where the value of α is changed, we account for the actual number of clicks and impressions that have occurred so far.

We explain the basic idea behind varying the thresholds as follows. Imagine that we are at the beginning of the planning horizon, i.e., for the first period of the month. We set the threshold at a smallest level such that the target click-through-rate level should just be achieved at the end of the month. We keep this threshold value constant for the current period. At the end of the period, we collect data on the number of ads displayed (specifically, the number of impressions) and the number of clicks. The current click-through-rate value is the number of clicks divided by the number of impressions. If this value is better than the required click-through-rate at the end of the month, we can afford to lower the threshold and show more ads. Conversely, if the current click-through-rate value 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.

Let us take a simple example to see how the dynamic approach works. To keep the example simple, let us imagine that there are 3 days in the planning horizon and that the value of α is updated on a daily basis. Let the publisher require us to achieve a click-through-rate constraint of 0.01 (1% click-through-rate). At the beginning of the first day, we find the lowest value of α that (if held constant) would just achieve the required click-through-rate of 0.01. Suppose this value is α_1. We set the threshold to α_1 for the first day. At the end of the first day, we make actual observations of the number of clicks (say $r_1 = 10$) and the number of impressions (say $m_1 = 2000$). We next use these values (r_1 and m_1) to find the lowest value of α that if held constant for the remaining 2 days, would just achieve the required click-through-rate of 0.01. Suppose this value is α_2. Intuitively, this value should be higher than α_1 since the click-through-rate in the first day (0.005) was below the required target (0.01). After considering the fact that we are behind on the constraint, we set the value of α to α_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 α needed to achieve the click-through-rate constraint (say, α_3). The third day would use a threshold of α_3. The difference between static and dynamic approaches is illustrated in Figure 3.

The dynamic approach has the advantage that it increases or reduces the threshold to account for actual feedback from the real-world. Thus, if we are ahead on the constraint, we can afford to use a lower value of α . Conversely, if real-world feedback is that we are behind on 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 click-through-rate that will be achieved if a certain value of α is held constant for the planning horizon (or the remainder of this horizon). As shown in the Appendix, exactly evaluating the expected click-through-rate can be extremely time consuming. At the same time, we need to evaluate this frequently if we plan to use the dynamic approach. Because the dynamic approach outperforms the static approach, it becomes imperative to find a fast and accurate approximation that will

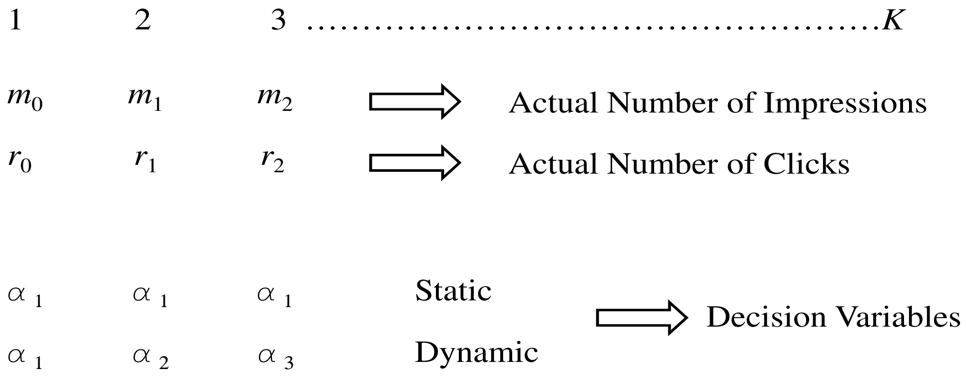

Figure 3 **The main difference between the static and dynamic approaches is that the threshold is varied in the dynamic approach.**

enable the calculation of α on a real-time basis. As we will see later in the next section, we often employ an hourly updating scheme for α . Thus, approximating the expected click-through-rate becomes a crucial step in solving the problem in the real-world. In summary, we explore four solution approaches as shown in Figure 4.

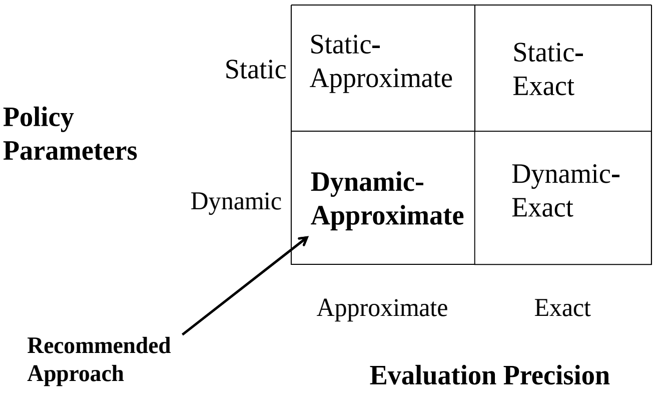

Figure 4 **The four solution approaches use different combinations of evaluation precision and policy parameters.**

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. This accuracy owes to the fact that when the sample sizes (number of impressions and the number of clicks) are large enough (for most publishers, it takes only a few hours to generate sufficient data), the values converge to their respective means. Thus we can replace all random quantities in the expression for the expected click-through-rate by their means. This allows us to calculate the expected click-through-rate very quickly. To provide an exact evaluation of the expected ctr, we would need several hours, even on a fast computer. However, this is a luxury we cannot afford - the value of α needs to be updated frequently (if possible, every hour) so the approximation is a huge benefit in solving the real problem in the company.

Providing Decision Support to Publishers: The Revenue Slider

A publisher may not know how to set the click-through-rate constraint. If the constraint is set too high, we will only show ads to interested visitors and while the click-through-rate will likely be high, the revenue from ads will suffer. To help publishers balance the trade-off between ad revenue and ad clutter, Chitika provides publishers with a "revenue slider." The revenue slider allows the publisher to slide a button and observe the revenue impact of a candidate value of the click-through-rate constraint. At each point along the slider (representing different values of the click-through-rate constraint), the slider application shows the expected revenue that the publisher will get at the end of a month, if the current value of the click-through-rate constraint is enforced. This revenue calculation uses the specific details of the publisher's traffic and the revenue sharing contract between Chitika and the publisher. Figure 5 shows an illustration of the revenue output and the expected click-through-rate from the slider for a specific publisher for different values of the click-through-rate constraint. Of course, as should be the case, the revenue is highest when the slider is at the extreme left, implying that the value of the click-through-rate constraint is zero and we show ads to all visitors. This setting will generate the most number of clicks, but the expected click-through-rate at the end of the month could be lower than what the publisher desires. The revenue slider application also shows the expected click-through-rate that results from a particular choice of the click-through-rate constraint. Beyond a certain point, it is not possible to meet an arbitrary click-through-rate constraint. Thus we restrict the slider's movement between zero and some high value that is specific for that publisher.

Experiments

In this section, we describe a series of experiments aimed at comparing the four solution approaches described earlier.

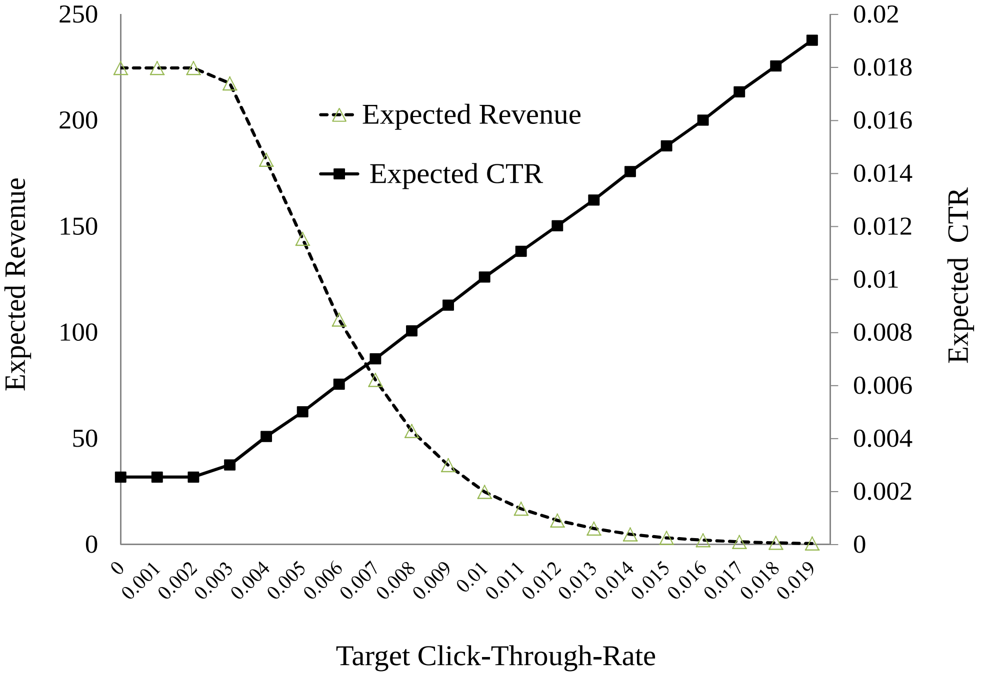

Figure 5 **The slider provides a prediction of the revenue and the click-through-rate for any given ctr constraint.**

Solution Selected

The test data used for comparing the four approaches consisted of 100,000 visits to a publisher's web site. To provide a valid comparison of the four approaches, we set the threshold for testing purposes to zero. Thus we show an ad for each of these 100,000 visits. For each visit, we recorded the predicted probability of a click (using the Logit model described earlier) and whether or not a click was generated. We then simulated the ad campaign for these 100,000 visits using each of the four solution approaches.

Table 2 shows a summary of results. We vary the number of periods (K), the value of the *ctr* constraint (η), and the approach used to manage the ad campaign (Dynamic-Exact, Static-Exact, Dynamic-Approximate, and Static-Approximate). For each solution approach, we report the number of impressions shown (m), the number of clicks generated (r), and the associated click-through-rate (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 extremely well; for both the dynamic and static approaches, there is hardly any difference between an exact and an approximate approach. However, the time taken to find the optimal solution for the Dynamic-Exact approach grows quickly and is quite significant even for a modest number of periods (9 periods). Focusing on the approximate approaches, the results also show that the

		Dynamic-Exact				Static-Exact			Dynamic-Approximate				Static-Approximate		
K	η	*m*	*r*	CTR	CPU Secs	*m*	*r*	CTR	*m*	*r*	CTR	CPU Secs	*m*	*r*	CTR
3	0.005	17377	166	0.009553	294	16438	165	0.010038	17377	166	0.009553	0	16438	165	0.010038
3	0.0075	14207	158	0.011121	621	7142	132	0.018482	14207	158	0.011121	0.01	7142	132	0.018482
3	0.01	10973	137	0.012485	889	2392	79	0.033027	10973	137	0.012485	0.016	2392	79	0.033027
3	0.0125	5927	74	0.012485	1276	1493	53	0.035499	5927	74	0.012485	0.026	1493	53	0.035499
3	0.015	4681	65	0.013886	1455	867	41	0.04729	4681	65	0.013886	0.032	867	41	0.04729
3	0.02	3204	54	0.016854	4934	513	26	0.050682	3204	54	0.016854	0.098	513	26	0.050682
6	0.005	35135	325	0.00925	1730	32905	321	0.009755	35135	325	0.00925	0	32905	321	0.009755
6	0.0075	31174	316	0.010137	3113	14210	252	0.017734	31174	316	0.010137	0	14210	252	0.017734
6	0.01	26258	288	0.010968	4967	4783	149	0.031152	26258	288	0.010968	0	4783	149	0.031152
6	0.0125	20542	260	0.012657	6035	3009	107	0.03556	20542	260	0.012657	0.01	3009	107	0.03556
6	0.015	14181	226	0.015937	7817	1754	78	0.04447	14181	226	0.015937	0.015	1754	78	0.04447
6	0.02	8819	183	0.020751	14035	1039	47	0.045236	8819	183	0.020751	0.04	1039	47	0.045236
9	0.005	52854	486	0.009195	5156	49335	483	0.00979	52854	486	0.009195	0	49335	483	0.00979
9	0.0075	48299	477	0.009876	9574	21509	384	0.017853	48299	477	0.009876	0	21509	384	0.017853
9	0.01	40790	445	0.01091	14170	7256	227	0.031284	40790	445	0.01091	0.019	7256	227	0.031284
9	0.0125	28016	358	0.012778	18650	4607	170	0.0369	28016	358	0.012778	0.031	4607	170	0.0369
9	0.015	21097	326	0.015452	22166	2716	125	0.046024	21097	326	0.015452	0.047	2716	125	0.046024
9	0.02	12736	275	0.021592	29033	1595	74	0.046395	12736	275	0.021592	0.016	1595	74	0.046395

Table 2 **A comparison of the four solution approaches shows that it is best to use the dynamic-approximate approach.**

dynamic approach typically generates more clicks than the static approach, while also meeting the click-through-rate constraint. The static approach often overshoots the target click-through-rate, resulting in a lower value for the number of generated clicks.

Table 3 provides more details on how a dynamic approach works. Here, we only consider the Dynamic-Approximate approach since the approximation works extremely well for the size of problems that one is likely to encounter for a typical Chitika publisher. In Table 3, we illustrate the Dynamic approach using a 30 day planning horizon. We update the value of α daily. It is instructive to see how a dynamic approach adjusts the threshold value to accommodate actual events (impressions, clicks). In the first period, the threshold chosen (α) is always the same for dynamic and static approach. However, if in the first period, there is a favorable draw of impressions and clicks (i.e., good progress has been made towards meeting the *ctr* constraint), a dynamic approach can risk lowering the threshold with the goal of generating more clicks. For example, on day 2, we see that the threshold is lowered from 0.0082 (that was used during the first day) to 0.006 for the second day. The static approach, of course, uses the same threshold (0.0082) for all 30 periods. In

	DYNAMIC				**STATIC**			
Day	**Alpha**	**Impressions**	**Clicks**	**CTR**	**Alpha**	**Impressions**	**Clicks**	**CTR**
1	0.0082	1036	38	0.03668	0.0082	1036	38	0.03668
2	0.006	3850	68	0.017662	0.0082	2226	58	0.026056
3	0.0059	5874	119	0.020259	0.0082	3090	91	0.02945
4	0.005	9845	167	0.016963	0.0082	4236	120	0.028329
5	0.0048	13996	220	0.015719	0.0082	5312	143	0.02692
6	0.0046	17655	269	0.015236	0.0082	6129	168	0.027411
7	0.0043	22551	320	0.01419	0.0082	7361	197	0.026763
8	0.0042	27164	355	0.013069	0.0082	8309	216	0.025996
9	0.0043	31468	424	0.013474	0.0082	9298	255	0.027425
10	0.0039	36681	496	0.013522	0.0082	10609	298	0.028089
11	0.0036	41710	543	0.013018	0.0082	11461	322	0.028095
12	0.0035	46865	606	0.012931	0.0082	12495	354	0.028331
13	0.0033	52338	661	0.012629	0.0082	13747	393	0.028588
14	0.0032	57598	703	0.012205	0.0082	14522	409	0.028164
15	0.0032	63006	765	0.012142	0.0082	15642	445	0.028449
16	0.003	68626	810	0.011803	0.0082	16762	465	0.027741
17	0.0031	73944	863	0.011671	0.0082	17581	498	0.028326
18	0.003	79527	918	0.011543	0.0082	18763	532	0.028354
19	0.0028	85139	953	0.011193	0.0082	19775	550	0.027813
20	0.003	90515	1021	0.01128	0.0082	20731	585	0.028219
21	0.0026	96285	1070	0.011113	0.0082	21938	608	0.027714
22	0.0026	101945	1132	0.011104	0.0082	22837	636	0.02785
23	0.0023	107689	1178	0.010939	0.0082	23798	657	0.027607
24	0.0023	113529	1228	0.010817	0.0082	25054	687	0.027421
25	0.0022	119316	1271	0.010652	0.0082	25891	709	0.027384
26	0.0022	125127	1351	0.010797	0.0082	26994	752	0.027858
27	0.0013	131053	1421	0.010843	0.0082	28237	789	0.027942
28	0.0006	137011	1477	0.01078	0.0082	29061	817	0.028113
29	0.0001	143008	1534	0.010727	0.0082	30230	853	0.028217
30	0.0001	149005	1589	0.010664	0.0082	31299	882	0.02818

Table 3 **While both approaches meet the ctr constraint, the dynamic approach produces more clicks than the static approach.**

this problem, we set the value of the *ctr* constraint to 0.01. As we can see, while both approaches meet this constraint, the dynamic approach (by varying the threshold across periods) generates more clicks than the static approach (1589 for dynamic versus 882 for static). Also, we observe that the static approach can overshoot the click-through-rate constraint and hence miss some revenue opportunities.

Implementation Challenges and Impact

Before we discuss implementation challenges, it is necessary to provide some background on Chitika's existing Ad data flow and network architecture.

The Ad Data Flow and Architecture

Figure 6 shows the ad data flow and architecture. Currently, Chitika delivers ads using ad servers in 5 data centers across the country. Each data center handles the ad traffic from a specific geographical region, e.g., Midwest, South, East, and so on. A geo-balancer placed at the head of the network ensures that ads servers in the correct geographical are contacted for ad delivery. Once the ad request goes to a data center in a particular geographical area, a load balancer within each data center ensures that no particular ad server in the data center gets over-loaded. There are several hundred ad-servers in each data center and each ad-server in a data center typically gets about 30-40 ad-requests per second. This request originates at a script that executes on the publisher's web page at the time the page is being rendered for the incoming visitor.

Not all ad-requests result in an ad display. This is because an ad-server must first decide whether or not to call the ad-partner for an ad. In order to make this decision, the ad-server calculates the probability of a click for that specific visitor and compares this value with the threshold for ad display for that publisher. If the click probability is higher than the threshold, the ad-sever makes a call to the ad partner to supply an ad. The partner returns the ad as an ad-unit (in about 200 milliseconds) that is ready for rendering on the publisher's page. Sometimes, it is necessary to get an image associated with the ad. In such cases, for fast processing, the ad-server obtains the image by calling an Akamai (`www.Akamai.com`) edge-server that stores the image. In other cases, the image may be locally available at the data center and hence it is not necessary to call an Akamai server. After displaying the ad, a log of this impression is created. This log records detailed data on the specifics of the ad shown, time, and other information concerning the visitor. If the visitor clicks on the ad, the ad-server logs this event. It is important to note that the ad-server that records the impression may be different from the one that records the click (although both are likely to be in the same data center). Once every hour, all the ad servers at a data center purge their logs to a local log processor at the data center. These local log processors (one for each data center), in turn, broadcast their 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 last hour. Thus at the end of each hour, we must process the raw log data 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

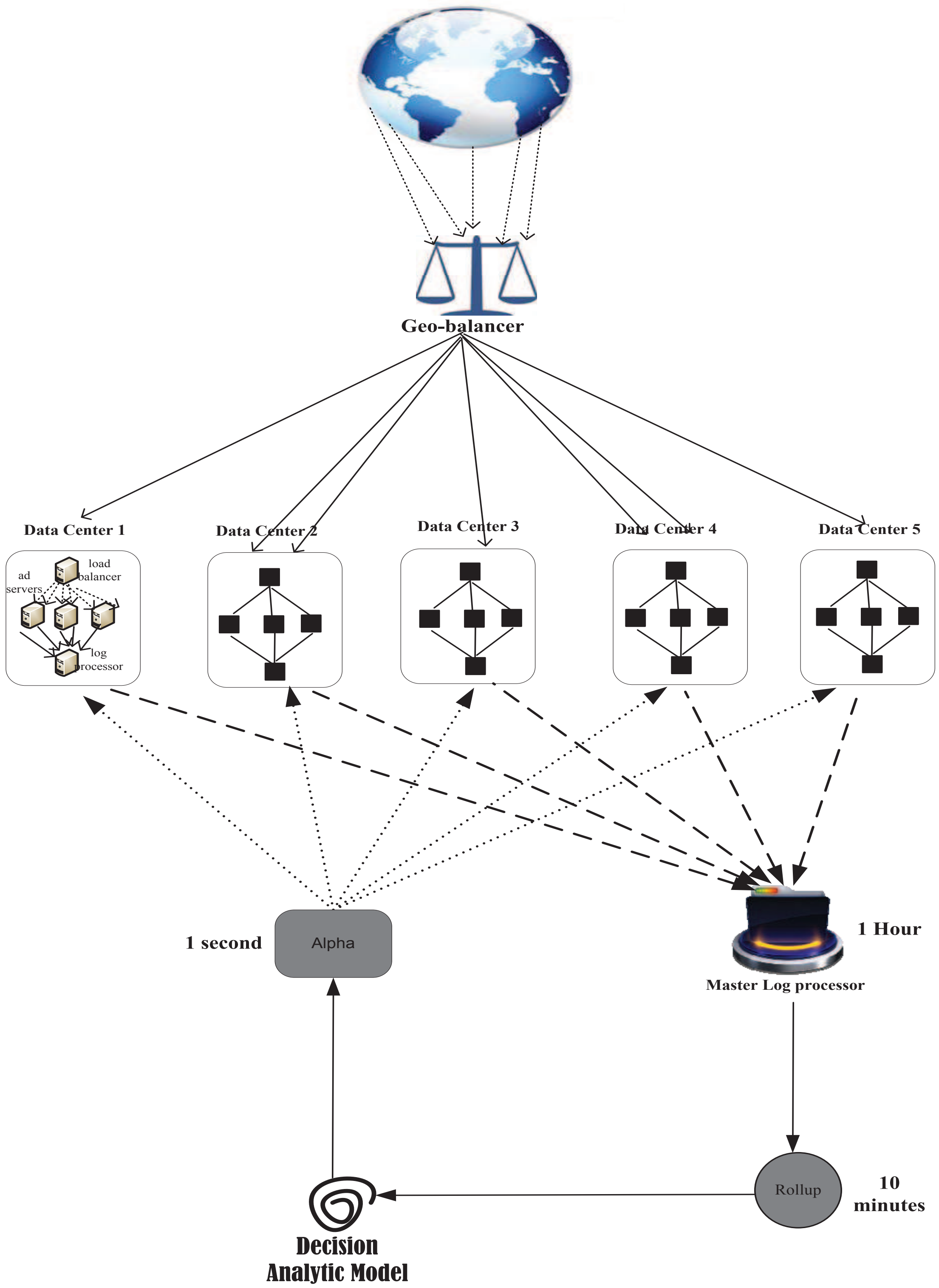

Figure 6 The ad-network architecture at Chitika creates several implementation challenges for optimally managing an ad campaign.

the page typically takes less than 0.5 second. 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 above 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 so that we can compare it against the threshold α. While the value of α stays constant for a period (say, for a few hours), we need to calculate the value of the click-probability (p) in real-time for every visitor. Recall that we need to calculate the click-probability using $p = L(\mathbf{X})$. The function is a polynomial that we can quickly evaluate using the knowledge of $\mathbf{X}$ for every visitor. A database lookup (for $\mathbf{X}$) is infeasible, given the time constraints. 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 we wish to update the value of α. Note that to find a new value of α, we need to know the actual events (impressions and clicks) that occurred in the last period. As mentioned above, Chitika processes the logs of these events only once every hour. Thus it is not feasible to update the value of α more than once every hour. In reality, even an hourly updating scheme may not be possible to implement for certain, low-traffic publishers. As we noted earlier, the dynamic approach (of frequently updating α) is only possible if we approximate the expected click-through-rate that results for any given choice of α. This approximation is extremely fast and accurate, but there is a proviso – the approximation requires that the sample sizes of the random quantities (specifically the number of impressions shown) are in thousands (≥ 5000). This requires that the traffic to the publisher must be sufficiently large. Thus the updating frequency must match the traffic of the publisher; an hourly updating scheme is possible for large publishers, but for relatively low-traffic publishers, we need to wait until a sufficient number of impressions have been shown before an update is done.

Benefits and Impact

The implementation began in March 2010 and has resulted in an increase in revenue for Chitika at the rate of about \$3,000 per day. Based on the data collected between March 2010 and September 2010, we estimated the total increase in revenue to be in the order of \$1.2 million per year. This revenue increase comes from Chitika being able to sign up more publishers under the Chitika

Premium program. Over the past year, Chitika was able to use its Premium program to partner with a very large ad aggregator to show ads in the United Kingdom. As part of the trial process, Chitika was asked to demonstrate a click-through-rate of 0.015, or 1.5%. Our methodology was able to provide a click-through-rate of 0.0151 or 1.51%. This accuracy won Chitika the contract and has contributed to a huge revenue increase for the company.

In December 2010, Chitika offered another service called Chitika *Select*. Most of Chitika's publishers came 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 the site from other sources (e.g., by directly typing in the url), Chitika chose not to show ads to such visitors as it could dilute the click-through-rate.

With Select, Chitika offered publishers the chance to expand the usage of ads (and hence drive more revenue) with the assurance that the expanded coverage would not dilute the Premium click-through-rate by more than 25%. Without a way to control the dilution in click-through-rate, the coverage expansion could seriously hurt the click-through-rate and hence risk losing some publishers completely. But with the solution, Chitika was able to guarantee a certain level of click-through-rate, and hence able to give the publishers this option with assurance. The Select offerring expanded out usage of Chitika's service across a large percentage of the network traffic. Whereas with Chitika Premium, Chitika only took search traffic and collapsed the ads for non search traffic, the Select service could show ads to a much larger traffic base. The Chitika Select offering gave the firm an additional 25% boost in revenue.

Impact on Other Areas of the Company

Inclusion of Advertiser Constraints This problem is similar to the one described in the paper except that the solution must respect the performance constraints of both the publisher and the advertiser. In addition to the publisher's constraint of exceeding a given click-through-rate, the advertiser often poses an additional constraint, a cost-per-conversion constraint. The advertiser's constraint requires that the cost-per-conversion is below a certain specified constant. The cost-per-conversion is the ratio of the total advertising cost (i.e., the cost-per-click times the number of clicks) divided by the number of conversions that are generated from the clicks. Once again, for a constant cost-per-click, 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 these probabilities. In other words,

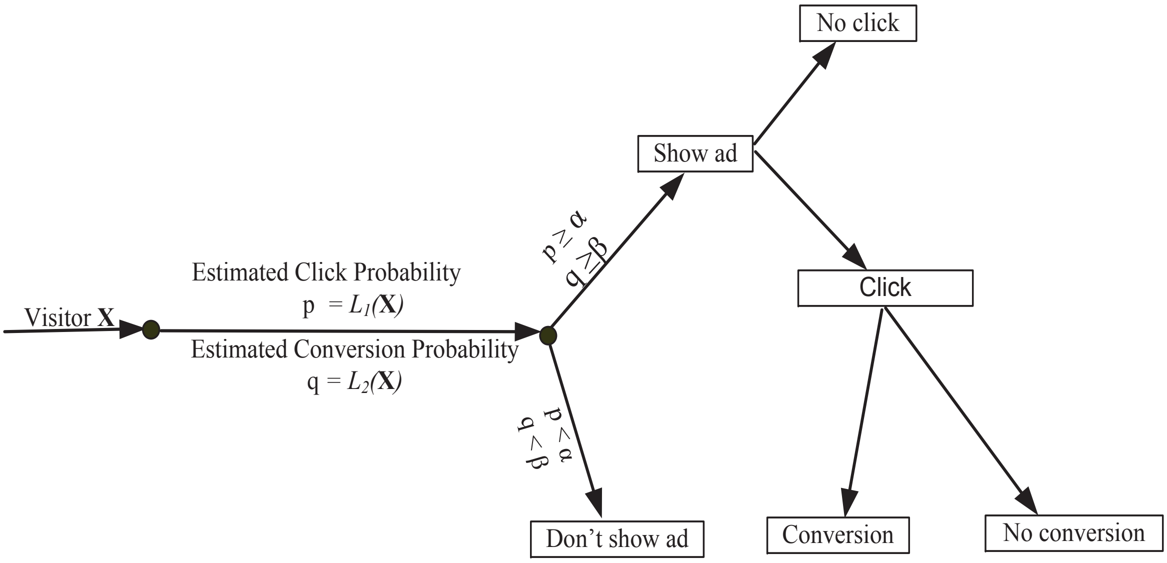

Figure 7 **Conversion constraints add another layer of logic in the decision-making process for ad display.**

we use two thresholds (one for the click probability and the other for the conversion probability) for ad display. Figure 7 depicts the overall solution process. Once again, we vary these thresholds over the planning horizon depending on current click-through-rate and the current conversion rate. The inclusion of advertiser constraints often applies to problems where Chitika directly contracts with the merchant for ad display, rather than obtaining these ads from an ad partner. Using the extended model, Chitika was able to win a contract to show ads for an ad partner that specializes in conversion-based payment schemes. This ad partner charges merchants on a cost-per-conversion basis, but pays Chitika on a per-click basis. However, to retain the business, Chitika must deliver a sufficient number of conversions to the merchant and at the same time, keep the publisher satisfied by using the website space efficiently.

Real-time Media Buying To extend its access to publishers, Chitika is actively considering buying impressions at ad-exchanges. These ad-exchanges allow ad-networks such as Chitika, to place a bid to buy the rights to show ads to a particular visitor on a particular website that is outside Chitika's current network of publishers. However, because of its large publisher network, Chitika has often encountered these visitors before within its network. Thus these visitors have a profile that can be leveraged to target them with relevant ads. The main difference between the problem described in this paper and real-time media buying is that in real-time bidding, Chitika pays for the impression regardless of whether there is a click or not. Thus, instead of a click-through-rate 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. This is a problem where the click-through-rate constraint disappears, but a cost-per-impression 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 α. Depending on whether the advertiser is a direct client of Chitika, a conversion constraint may still be applicable in the real-time media buying problem. There is a huge market opportunity that awaits the firm if it is able to fine-tune and adapt the ideas in Chitika Premium and Chitika Select to media buying.

Fading Ads Here the idea is to show the ad, but fade away after a specific amount of time. This is a relaxation of the model discussed earlier, where either the time for which the ad is shown is zero (for the case where $p < \alpha$) or the ad is shown for a fixed amount of time. A fading ad (or *fad*, for short) will fade away after a certain time (say, τ), depending on the visitor's previous click history. This click history not only contains whether or not the visitor has clicked on ads, but also the time it took the visitor to generate these clicks. Such an innovation, if correctly implemented, will provide Chitika with a very distinctive product in the internet advertising space.

Lessons Learned

In this study, we develop several approaches to manage an on-going internet ad campaign for Chitika using visitor profiling. The basic idea is to not show ads to every visitor, but show ads to only those visitors that have a reasonable chance of generating a click. Most publishers would like to maximize the revenue earned from ads (via clicks), but not clutter the web site with too many ad impressions that do not generate a click. Thus, for Chitika, which does not directly suffer from un-clicked ad impressions, it is necessary to optimize the revenue earned from the ad campaign while respecting a click-through-rate constraint that is supplied by the publisher. We formulate this optimization problem and find the optimal value of the decision variable (a probability threshold that governs the display of ads) using simple numerical search. Since an ad campaign usually lasts several weeks (or even months), it raises the potential to dynamically manage the campaign (i.e., change the probability threshold across periods during the planning horizon). The dynamic approach is attractive, but because we need to solve the problem again in every period of the planning horizon, its use could raise concerns about the time taken to solve the optimization problem. Motivated by the need to quickly solve the problem in each period, we created an approximate version of the dynamic approach. In this approach, we approximated the expected click-through-rate 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 (often in the thousands, for the number of visitors per period), the approximation works extremely well and the optimization problem can be solved in each period very fast, typically in a few milliseconds. Thus we are optimistic about the use of a dynamic approach to optimize an on-going ad campaign. As mentioned earlier, the benefit of a dynamic approach is that it can make the probability threshold more or less strict, depending on the progress made towards the final goal of achieving the click-thorough-rate 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 the cumulative number of clicks.

Appendix

Optimization Model

We have summarized the details and definitions of model parameters and variables in Table 4. The question of interest in this study is to use the click-probability p to decide whether or not to show an ad to a visitor. Let α 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 α. For a given value of the threshold α and for a given time period, we denote the random variable for the total number of clicks (resp. total number of impressions) as $\tilde{r}$ (resp. $\tilde{m}$).

Consider a planning horizon of K periods, with a uniform arrival rate (of visitors) of λ per period. As mentioned, the objective for the firm is to maximize the number of clicks during a given planning horizon. With appropriate scaling, this objective amounts to maximizing the revenue accrued to the ad firm. The expected number of clicks ($\mathrm{E}[\tilde{r}]$) can be evaluated as

$$\mathrm{E}[\tilde{r}] = \sum_{i=0}^{K\lambda} \sum_{j=0}^{i} j \, \Pr[\tilde{r} = j | \tilde{m} = i] \, \Pr[\tilde{m} = i]$$

SYMBOL	DEFINITION
K	Number of periods in the planning horizon
λ	Number of visitors per period
m_0	Initial number of impressions
r_0	Initial number of clicks
$\tilde{m}$	Random variable for the number of impressions in the planning horizon
$\tilde{r}$	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
α	Click probability threshold **(Decision Variable)**
$\gamma(\alpha)$	Probability of impression for threshold α
$\delta(\alpha)$	Probability of click for threshold α
η	Publisher's click-through-rate constraint

Table 4 Model Parameters and Variables

The probability that there are i impressions shown in the planning horizon is binomially distributed, given by

$$\Pr[\tilde{m}=i]=\binom{K\lambda}{i}(\gamma(\alpha))^{i}(1-\gamma(\alpha))^{K\lambda-i}$$

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, the probability that an ad is shown to a visitor can be calculated as

$$\gamma(\alpha)=\int_{\alpha}^{1} f(p)dp$$

The conditional probability that there are j clicks generated from i impressions is also binomially distributed, given by

$$\Pr[\tilde{r}=j|\tilde{m}=i]=\binom{i}{j}(\delta(\alpha))^{j}(1-\delta(\alpha))^{i-j}$$

In the above, the probability that a visitor will click on an ad for a given threshold α is given by

$$\delta(\alpha)=\frac{\int_{\alpha}^{1} pf(p)dp}{\gamma(\alpha)}$$

Properties of $\gamma(\alpha)$ and $\delta(\alpha)$

It is obvious from the expression for $\gamma(\alpha)$ that the probability of showing an impression decreases with the threshold, α. Further, it can be shown that the probability of a click for a given value of α, $\delta(\alpha)$ increases with α.

$$\frac{d\delta(\alpha)}{d\alpha}=\frac{-\left(\int_{\alpha}^{1} f(p)dp\right)\alpha f(\alpha)+f(\alpha)\left(\int_{\alpha}^{1} pf(p)dp\right)}{(\gamma(\alpha))^{2}}$$

The above expression is strictly positive since $\int_\alpha^1 \alpha f(p)dp < \int_\alpha^1 pf(p)dp$. Thus the probability of a click increases with α.

Problem P

Problem P can be described as one of maximizing the number of clicks in a given planning horizon, subject to an expected click-through-rate constraint. The decision variable is the probability threshold, α. Using the above definitions, the optimization problem can be formally expressed as below

$$\max_\alpha \mathrm{E}[\tilde{r}]$$

subject to

$$\mathrm{E}\left[\frac{r_0+\tilde{r}}{m_0+\tilde{m}}\right] \geq \eta$$

In the above problem, the initial number of clicks is denoted by r_0 and the initial number of impressions is denoted 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 problem P is being solved at an intermediate stage (say, period j), then the value of K is replaced by $K - j$.

Solution to Problem P

To solve problem P, we show that the objective function is decreasing in the threshold, α. To see why this is so, note that in the expression for $\mathrm{E}[\tilde{r}]$, the term $\Pr[\tilde{m} = i]$ is independent of j. The inner summation is therefore equal to the expectation of a binomial distribution with i trails and a success probability of $\delta(\alpha)$, or $i\delta(\alpha)$. We therefore have,

$$\mathrm{E}[\tilde{r}] = \delta(\alpha)\sum_{i=0}^{K\lambda} i\binom{K\lambda}{i}(\gamma(\alpha))^i(1-\gamma(\alpha))^{K\lambda-i} = K\lambda\delta(\alpha)\gamma(\alpha)$$

Thus, in Problem P, it is sufficient to maximize $\int_\alpha^1 pf(p)dp$, which is decreasing in α. Therefore the optimal solution to problem P is the smallest value of α 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 click-through-rate is exact or approximate. Thus the four solution approaches are: SE (Static-Exact), SA(Static-Approximate), DE (Dynamic-Exact) and DA (Dynamic-Approximate). In static optimization, we use the same value of the threshold (α) for the entire planning horizon. Thus, at the beginning of the planning horizon, the values of the number of impressions and clicks are set to zero 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 for the planning horizon. Here, values of the number of impressions and the number of clicks are set to actual (or observed) values. Thus the threshold can be different for each period in the planning horizon. The other feature of a solution approach is whether the calculation of the expected click-through-rate is exact or approximate. In an approximate approach, the expected click-through-rate is calculated by replacing the random variables for the number of clicks and the number of impressions by their expected quantities,

$$\mathrm{E}\left[\frac{r_0+\tilde{r}}{m_0+\tilde{m}}\right]_{\text{approx}} = \frac{r_0+k\lambda\delta(\alpha)\gamma(\alpha)}{m_0+k\lambda\gamma(\alpha)}$$

In the above approximation, k is the number of periods left in the planning horizon, $k \in [1,2,3..,K]$. The approximation technique yields very fast results and can be useful when the value of $K\lambda$ is high, say, in the hundreds of thousands, and the threshold has to calculated quickly to manage an on-going ad campaign. The approximation is also useful because a closed-form expression is not available for the optimal value of α. Rather, the optimal has to be found using a numerical search procedure that begins with a very low value of α and increments this value until the *ctr* constraint is first satisfied.

References

Dawande, M., Kumar, S., and Sriskandarajah, C. 2003. Performance bounds of algorithms for scheduling advertisements on a web page. *Journal of Scheduling* 6(4) 373-394.

eMarketer. 2011. Online advertising market poised to grow 20% in 2011. June 8. Retrieved October 6, 2011, `http://www.emarketer.com/PressRelease.aspx?R=1008432`.

Fenton, B. 2011. Online advertising retakes top slot. *Financial Times* October 5. Retrieved October 6, 2011, `http://www.ft.com/intl/cms/s/0/330f645e-edda-11e0-a491-00144feab49a.html`.

Hartl, R.F. 1982. Optima1 control of non-linear advertising models with replenishable budget. *Optimal Control Applications and Methods* 3(1) 53-65.

Hof, R. 2011. Online ad spending jumps, led By video and Facebook. *Forbes* July 5. Retrieved October 6, 2011, `http://www.forbes.com/sites/roberthof/2011/07/05/online-ad-spending-jumps-led-by-video-and-facebook/`.

Hof, R. 2011. Online ad spend to overtake TV by 2016. *Forbes* August 26. Retrieved October 6, 2011, `http://www.forbes.com/sites/roberthof/2011/08/26/online-ad-spend-to-overtake-tv/`.

Interactive Advertising Bureau. 2011a. IAB internet advertising revenue report: An industry survey conducted by PwC and sponsored by the Interactive Advertising Bureau (IAB), 2010 Full Year Results. April. Retrieved October 6, 2011, `http://www.iab.net/media/file/IAB_Full_year_2010_0413_Final.pdf`.

Interactive Advertising Bureau. 2011b. Internet ad revenues at nearly $15 billion in first-half 2011, up 23%, second quarter 2011 breaks record again. September 28. Retrieved October 6, 2011, `http://www.iab.net/about_the_iab/recent_press_releases/press_release_archive/press_release/pr-092811`.

Joshi, A., Bagherjeiran, A., and Ratnaparkhi, A. 2011. User demographic and behavioral targeting for content match advertising. *Proceedings of the Fifth International Workshop on Data Mining and Audience Intelligence for Advertising, ACM*, San Diego, 53-60.

Kumar, S., Jacob, V.S., and Sriskandarajah, C. 2006. Scheduling advertisements on a web page to maximize revenue. *European Journal of Operational Research* 137(3) 1067-1089.

Kumar, S., Dawande, M., and Mookerjee, V.S. 2007. Optimal scheduling and placement of internet banner advertisements. *IEEE Transactions on Knowledge and Data Engineering* 19(11) 1571-1584.

Mangalampalli, A., Ratnaparkhi, A., Hatch, A.O., Bagherjeiran, A., Parekh, R., and Pudi, V. 2011. A feature-pair-based associative classification approach to look-alike modeling for conversion-oriented user-targeting in tail campaigns. *Proceedings of the 20th International Conference Companion on World Wide Web, ACM*, New York, 85-86.

Monica, P.R. 2005. Consumers keep on clicking: Stronger than expected 2Q results from Amazon is another good sign for e-commerce companies. *CNN Money* July.