Personalized Ad Delivery when Ads Fatigue: An Approximation Algorithm

Zoë Abrams[1] and Erik Vee[2]

[1] Yahoo!, Inc., 2821 Mission College Blvd., Santa Clara, CA, USA
[2] Yahoo! Research, 2821 Mission College Blvd., Santa Clara, CA, USA

Abstract. We consider a crucial aspect of displaying advertisements on the internet: the individual user. In particular, we consider *ad fatigue*, where a user tires of an advertisement as it is seen more often. We would like to show advertisements such that, given the impact of ad fatigue, the overall efficiency of the system is optimized. We design an approximation algorithm, for the case that we study, that approaches the optimum as the number of unique ads shown, if there is only one available position, increases.

1 Introduction

Internet advertising is a booming business, and already provides a large portion of search engine revenue. Placing ads strategically is key to optimizing the efficiency of these advertising systems. The current measures used in most academic literature to determine the match between advertisement and user is the estimated clickthrough rate, based on the advertisement, the keyword, and the position on the web page [1, 11, 5, 10, 2, 6, 7]. In practice, landing page information [12] and demographic targeting [9] have been used to better match advertisements to users.

In this paper, we explore another crucial aspect in determining when and where to place ads: the individual user. There are many ways that an individual user's experience may influence how an advertisement is received, including previous positive or negative experiences with advertisements and previous exposure to a company name or logo. We concentrate in this study on how previous experiences viewing an advertisement influence a user's likelihood to click on that advertisement. The setting used is one where ads are embedded in webpages, such as AdSense at Google or Content Match at Yahoo!. An individual user may view the particular site several times in a single day, if the site is a user's homepage or a frequently visited resource, and ads are less likely to be clicked as they are shown more often. This phenomena is referred to as *ad fatigue* [4], since the user tires of the ad after viewing it several times. We study the problem of determining which ads to display where and when in order to maximize efficiency over several displays of the same page. Efficiency is considered to be the expected number of clicks times the value of that click to the advertiser. We design an algorithm that achieves close to optimum efficiency.

2 Model

Our model follows that of [8]. We assume the input to the algorithm is:

- The max number of times, T, a particular web page will be viewed by a single user during the course of some time period.
- A vector f of fatigue rates that contains T values. We use f_t to denote the value corresponding to the t^{th} element of the vector f, and $\forall t, 0 \leq f_t \leq 1$.
- A vector CTR representing the decay in expected clicks due to position on the page, sorted from most to least likely position to be clicked. We use CTR_p to denote the value corresponding to position p. There are $p \in \{1, ..., P\}$ positions total.
- A set of ads, each with a value $v_i, i \in \{1, ..., I\}$. For an ad i, v_i represents the value to the advertiser showing the ad for receiving a click, times the ad dependent component of the click through rate. Although in practice, values are private and known only to the advertiser, there are options for estimating this value based on information observable to the search engine [11].

We use the word *slot* to signify an opening for an advertisement at a particular time and in a particular position. Our results, algorithm and model make the following assumptions:

- Ad values are all a power of some base b.[3]
- Values in f are decreasing, and all of the form $f_t = \frac{1}{b^t}$.
- The expected efficiency of ad i when shown for the k^{th} time in position p is $CTR_p \cdot f_k \cdot v_i$.
- A slot can be left empty.

For every time slot t, position p, and bidder i, let x_{ipt} be an indicator variable for whether bidder i is shown at time t in position p. We require a solution such that, $\forall it, \sum_p x_{ipt} \leq 1$ (in words, an ad can only be shown in one position for a given time slot). Let $\forall it$, $y_{it} = \sum_{k<t,p} x_{ipt}$. In words, y_{it} is the number of times bidder i has already been shown, at time t, in the solution so far. The problem is to decide which advertisement to show in which position and at which time such that we maximize

$$\text{Total Efficiency} = \sum_{ipt} x_{ipt} \cdot CTR_p \cdot f_{y_{it}} \cdot v_i.$$

Motivating Example We show that the greedy approach of placing the best ad in the best slot currently available is not optimal. Say advertisers A and B have value 1 for placing their ads, C has value $\frac{1}{2}$, the fatigue vector $f = [1, \frac{1}{2}]$, position clickthrough rates are $CTR = [1, \frac{1}{2}]$, and there are two time

[3] This assumption can be relaxed by rounding ad values down to the nearest number with base b and multiplying the result in Theorem 1 by a factor of $\frac{1}{b}$.

periods $T = 2$. The greedy placement of ads is shown in Figure 1 (the top left triangle is the advertiser and the bottom right the efficiency they contribute). The total efficiency is 2.25. However, the optimal solution (seen in Figure 2), has a total efficiency of 2.5. In the (extreme and impractical) worst case, the greedy approach can lead to as much as T times less than the optimal efficiency.

	$t = 1$	$t = 2$
Position 1	Ad: A Bid: 1	Ad: A Bid: $\frac{1}{2}$
Position 2	Ad: B Bid: $\frac{1}{2}$	Ad: B Bid: $\frac{1}{4}$

Fig. 1. Greedy Solution

	$t = 1$	$t = 2$
Position 1	Ad: A Bid: 1	Ad: B Bid: 1
Position 2	Ad: C Bid: $\frac{1}{4}$	Ad: A Bid: $\frac{1}{4}$

Fig. 2. Optimal Solution

3 An Approximation Algorithm

We now describe our algorithm and give an approximation guarantee for its performance. The algorithm works in two stages. In the first stage, we generate a "flattened tableau": rather than showing a set of P different ads (placed in positions 1,2,..., P) during each of the T time steps, imagine that we show just one ad during each of $P \cdot T$ time steps. For the first T time steps, we value the ad as though it were shown in position 1. For the next T time steps, we value the ad as though it were shown in position 2, and so on. Ad fatigue is applied normally. We can produce an optimal flattened tableau in a greedy fashion.

Consider "unflattening" the tableau. Take the first T ads and place them all in position 1 (in their respective positions). Take the next T ads and place them all in position 2, and so on. This causes two problems: the first is that ads may appear more than once for a given time step. The second is that the value that we ascribed to an ad in Stage I may no longer be accurate; an ad that appears for the first time in, say, the 2nd position in the flattened tableau, may appear for the first time in the 3rd position in the unflattened tableau. In Stage II, we provide a method for rearranging (and removing) ads so that these problems are resolved. Our rearrangement is provably close to the efficiency for the unflattened tableau, which is more efficient than the optimal solution.

Stage I: Finding an optimal flattened tableau
As we outlined above, Stage I of our algorithm finds the optimal placement of ads in the flattened tableau. Let $\bar{y}_{ipt}$ be the number of times that ad i has appears in slots (p', t') such that either $p' < p$ or $p' = p, t' \leq t$, assuming i is placed in slot (p, t). In symbols, $\bar{y}_{ipt} = \sum_{p'<p} \sum_t x_{ip't} + \sum_{t'<t} x_{ipt'}$. We seek to maximize $\sum_{ipt} x_{ipt} \cdot CTR_p \cdot f_{\bar{y}_{ipt}} \cdot v_i$. Note that this is the total efficiency, with

y_{ipt} replaced with $\bar{y}_{ipt}$. We define $f_{\bar{y}_{ipt}} \cdot v_i$ to be an advertiser's "flattened value". The pseudocode is given below.

STAGE I of ALGORITHM $\mathcal{A}$: FLATTENED TABLEAU
Initialize the "flattened value" of each ad i to be v_i.
For $p = 1$ to P
  For $t = 1$ to T:

1. Place the ad with the largest "flattened value" in slot (p, t). Ties are broken lexicographically.
2. Update the "flattened value" of this ad by dividing by b.

We now show that the efficiency of the flattened tableau (using "flattened values") is better than the efficiency of the optimal solution of the original problem (using the actual values). Let x_{ipt}^{A} and x_{ipt}^{OPT} be the assignment variables produced by the algorithm of Stage I, and the optimum solution, respectively.

Lemma 1. $\sum_{ipt} x_{ipt}^{OPT} \cdot CTR_p \cdot f_{y_{ipt}} \cdot v_i \leq \sum_{ipt} x_{ipt}^{OPT} \cdot CTR_p \cdot f_{\bar{y}_{ipt}} \cdot v_i \leq \sum_{ipt} x_{ipt}^{A} \cdot CTR_p \cdot f_{\bar{y}_{ipt}} \cdot v_i$.

Proof The first two equations differ only in the index for the ad fatigue factor. Consider a single ad i. All variables multiplying the ad fatigue factor remain unchanged. The ad fatigue indices are 1 through the number of occurrences of the ad in both equations. The only thing that changes is the pairing of fatigue values to the other variables. But, the way to pair fatigue values with the other multipliers to maximize efficiency is exactly the pairing created by the $\bar{y}$ variables[4]. It is not hard to verify that algorithm $\mathcal{A}$ produces the flattened tableau with optimal efficiency. □

Stage II: Reconstructing the flattened tableau
We now use the flattened tableau to construct the final (real) tableau. Stage I produced an assignment of ads to slots. In Stage II, we would like to guarantee that, for each i, all occurrences of ad i in position p appear before any occurrences of ad i in position $p+1$. We accomplish this by arranging the ads within each position, then shifting ads to the right until our guarantee is met. The algorithm is given below. We show later how to bound the loss in efficiency due to shifting ads to the right.

[4] For any two vectors of positive real numbers, the maximum dot product is achieved when both are sorted in ascending (or descending) order.

STAGE II of ALGORITHM $\mathcal{A}$: FEASIBLE TABLEAU
Given the assignment from Stage I, let S_p be the multiset of ads appearing in position p.

1. For each $p \in \{1, ..., P\}$, sort the ads in S_p in ascending order of their v values (breaking ties lexicographically). Place the first ad in slot $(p, 1)$, the second ad in slot $(p, 2)$, and so on. (Note that S_p is a multiset, so the same ad may appear multiple times, always in a contiguous sequence.)
2. For $p = 2$ to P, shift ads to the right until the last time ad i appears in position $p-1$ is before the first time ad i appears in position p, for all i.

We now prove the approximation ratio for algorithm $\mathcal{A}$. Let $last(j, p)$ and $first(j, p)$ be the last and first, respectively, time at which bidder j is shown in position p. Let Q_{jp} be the set of unique ads shown after ad j (including j) and in position p. Let Q_p be the set of unique ads shown in position p (Q_p is the set version of the multiset S_p). Observe that there could be ads with flattened values in position p that are equal to flattened values in position $p-1$. We ignore this complication and assume flattened values in position p are strictly less than flattened values in position $p-1$. Eliminating this assumption requires Step 2 of Stage II shift ads to the right at most an additional 2 slots. See the full version of this paper for more details.

Lemma 2. *Step 2 of Stage II shifts any row p over by at most an additional $\frac{T}{|Q_{p-1}|}$ time periods after aligning with the row above it.*

Proof The average number of times an ad is shown in position $p-1$ is $\frac{T}{|Q_{p-1}|}$. For any j, $last(j, p-1)$ is at most $T - |Q_{j,p-1} - 1| \frac{T}{|Q_{p-1}|}$ since every ad shown afterwards is shown at least as often as the average (sorted in increasing order, by number of times shown). $first(j, p)$ is at least $T - |Q_{j,p}| \frac{T}{|Q_{p-1}|} + 1$ since each ad from the previous position is shown (in position p) at most the average number of showings from the previous position $(p-1)$. By definition and Step I of Stage II, $|Q_{j,p-1}| = |Q_{j,p}|$. Clearly, shifting row p over by $\frac{T}{|Q_{p-1}|}$ time units is sufficient to guarantee $last(j, p-1) \leq first(j, p)$. $\square$

Define Avg to be the average efficiency of a bidder in position 1, divided by CTR_1. Precisely, $Avg = \frac{\sum_{i1t} x_{i1t} \cdot f_{y_{it}} \cdot v_i}{|Q_1|}$.

Theorem 1. *Algorithm $\mathcal{A}$ has efficiency at least $OPT - Avg \cdot \sum_{p=2}^{P} (p-1) \cdot CTR_p$.*

Proof Algorithm $\mathcal{A}$ before step 3 of Stage II finds x_{ipt} that maximizes $E^A = \sum_{ipt} x_{ipt} \cdot CTR_p \cdot f_{\bar{y}_{ipt}} \cdot v_i$. By Lemma 1, optimum efficiency $\leq E^{OPT} \leq E^{\mathcal{A}}$. The efficiency of A after Step 3 is the $E^{\mathcal{A}}$ minus the efficiency in $E^{\mathcal{A}}$ from ads that are removed by the process of shifting over the rows. Since $v_i f_{\bar{y}_{ipt}}$ only decreases as the position increases, by Lemma 2, the efficiency of units shifted off row p is at most $\sum_{k=1}^{p-1} \frac{T}{|Q_k|} \cdot \frac{CTR_p \sum_{i1t} x_{i1t} \cdot f_{y_{it}} \cdot v_i}{T} \leq (p-1) \cdot CTR_p \cdot Avg$. Summing over all positions, at most $Avg \cdot \sum_{p=2}^{P} (p-1) \cdot CTR_p$ is lost. □

Corollary 1. *Algorithm $\mathcal{A}$ has efficiency at least $OPT \cdot (1 - \frac{P-1}{|Q_1|})$.*

4 Future Work

There are several possible avenues for future work on this problem. First, it would be useful to generalize the result to handle fatigue rates that are not geometrically decreasing. We observe that our algorithm places the most valuable ads latest, so T must be a lower bound on the number of times the page will be shown, to avoid eliminating the most valuable advertisements. There may be other ways to model uncertainty about the number of page showings in the future. There is also the possibility that the ideas presented here will work for personalization in other contexts, such as creating a coordinated advertising campaign targeted at an individual, across multiple domains (i.e. sites for mail, search, etc.) and mediums (i.e. banner ads, studied in [3]). Finally, we do not know the hardness of the problem we study.

References

1. Zoe Abrams, Ofer Mendelevitch, and John Tomlin. Optimal delivery of sponsored search advertisements subject to budget constraints. *ACM Conference on Electronic Commerce*, 2007.
2. Gagan Aggarwal, Ashish Goel, and Rajeev Motwani. Truthful auctions for pricing search keywords. *ACM Conference on Electronic Commerce*, 2006.
3. Patrali Chatterjee, Donna L. Hoffman, and Thomas P. Novak. Modeling the clickstream: Implications for web-based advertising efforts. *Marketing Science*, 22:4:520–542, 2003.
4. Harold Davis. *Google Advertising Tools*. 2006.
5. Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, 2007.
6. Daniel Fain and Jan Pedersen. Sponsored search: a brief history. *Proceedings of the Second Workshop on Sponsored Search*, 2006.
7. Juan Feng, Hemant K. Bhargava, and David M. Pennock. Implementing sponsored search in web search engines: Computational evaluation of alternative mechanisms. *Knowledge and Data Management*, 2005.
8. Prachi Gupta and Helmut Geschwollen. Automatic ad rotation via modeling effects of presentation count on clickthrough rate. Technical report, 2006.
9. Przemyslaw Kazienko and Michal Adamski. Personalized web advertising method. *The Third International Conference on Adaptive Hypermedia*, 2004.
10. Sebastien Lahaie and David M. Pennock. Revenue analysis of a family of ranking rules for keyword auctions. *ACM Conference on Electronic Commerce*, 2007.
11. Hal R. Varian. Position auctions. *International Journal of Industrial Organization*, 2005.
12. Robin Zeff and Brad Aronson. *Advertising on the Internet*. 1999.