New Stochastic Algorithms for Placing Ads in Sponsored Search*

Esteban Feuerstein
Depto. de Computación, Fac. de Cs. Exactas y Naturales, Univ. de Buenos Aires
efeuerst@dc.uba.ar

Pablo Ariel Heiber
Depto. de Computación, Fac. de Cs. Exactas y Naturales, Univ. de Buenos Aires
pheiber@dc.uba.ar

Javier Martínez-Viademonte
Depto. de Computación, Fac. de Cs. Exactas y Naturales, Univ. de Buenos Aires
javiermv@dc.uba.ar

Ricardo Baeza-Yates
Yahoo! Research
Barcelona, Spain & Santiago, Chile
ricardo@baeza.cl

ABSTRACT

We introduce a family of algorithms for the selection of ads in sponsored search that intends to increase ad diversity while not significantly reducing revenue and maintaining an incentive for advertisers to keep their bids as high as possible. Diversification of ads may be useful for many different reasons. Our algorithms try to distribute the available slots among *all* ads, using some kind of proportional mechanism based on the bids and the expected CTRs of the ads. Although in our experiments we used a simple first-price auction, our algorithms are compatible with strictly incentive-compatible auctions and pricing mechanisms. We have analyzed the performance of our algorithms both assuming a static intrinsic CTR associated to each ad and in the more general case in which ads' CTR varies dynamically with time. The main result is that the best among our algorithms perform well compared to the traditionally used (in some cases even better), while increasing notably the diversification of the published ads.

1. INTRODUCTION

The most important source of revenue for search engine companies in the Internet is the income they obtain by selling ads that are provided to the users that use their sites besides the so-called editorial results of the search. Another source of revenue is given by the sell of ads that are contextually associated to the contents of the browsed pages on third-party sites depending on the presence of certain words or terms. In both cases, advertisers pay only for each visit to their site coming from a click in their ad, in what is known as the Pay-per-click (or PPC) model. PPC has been one of the key factors of the success of companies such as Google or Yahoo!, since it allowed the inclusion in the market of a huge amount of advertisers which would have stayed outside of the game if the business model would have remained with the more traditional Pay-per-Impression model.

In the most typical setting, a certain number of slots is available for sponsored results, and these slots are auctioned among the interested advertisers and distributed according to some more or less public criteria. These criteria are naturally oriented, in the long run, to optimize the search engine's revenue, but this objective entails, as it has been studied by other authors, a good user experience and the conformity of the advertisers. Even if we think only in optimizing the revenue, we must note that the simplest mechanisms one can think of, like choosing the ads whose bids are maximum, are not optimal, as they do not take into account important factors like the probability of the ad being clicked (this is the clickthrough rate or CTR of the ad).

The real CTR of a particular ad is not known in practice, so that value must be estimated. Generally, the estimation is updated dynamically, as the ads are (or are not!) clicked by users. There are different ways of doing this, and the task is even more complicated when ads are displayed and clicked/not clicked in more than one position. When there is place for more than one ad, the order in which ads are presented plays a significant role. In all search engine sites, ads appear in an ordered list (though in some cases there may be more than one list). The expected clickthrough of an ad depends strongly on the position in which it appears in that list, and the most widely assumed model is the "exponential decay model", that considers that there is some constant $\alpha > 1$ such that the expected clickthrough at a certain position is $1/\alpha$ times the expected clickthrough at the previous position of the listing (although a power law model similar to the distribution of clicks in search results is also possible).

This problem and its variations have been widely studied in the last years (e.g. [4, 7]). In all known algorithms, ads are selected and ordered according to the advertisers' bids, the ads' expected CTRs and eventually also their combination with other variables. But all these algorithms share the same pattern of behavior, that we could call "winner takes

*This research was funded by a Yahoo! Research Alliance Grant. We are submitting this paper for confidential review to be considered for publication in the Third Workshop of Sponsored Search Auctions on May, 2007."

WWW2007, May 8–12, 2007, Banff, Canada.
.

all" philosophy. The so called "fat tail of the distribution" is not completely taken into account, as many advertisers remain out of the game if their offers are not among the most convenient regarding the function used to select them. Moreover, this pattern of behavior may leave out also ads with high potential revenue due to errors in the initial estimation of the CTR. This has motivated in the past the necessity of introducing some mechanisms that try to obtain an explore/exploit trade-off, by alternating, in a deterministic or stochastic way, ads with smaller or unknown CTR with those of higher revenue expectation.

Following that direction, in this paper we will introduce a family of algorithms that intend to select a more varied set of ads (actually, in this algorithms, *all* ads will be eventually shown a fraction of the times), while not significantly reducing the revenue and maintaining an incentive for advertisers to keep their bids as high as possible. This could imply many benefits:

- More and happier advertisers *(in particular is an incentive to get a wider base of advertisers, bringing in the "fat tail", but still giving high visibility to strong bidders and decreasing ad starvation)*

- Happier editor *(increased aggregate click rate of the selected ads, higher total expected click-through in the short term and in the long run, a wider base of advertisers represents a wider set of bidding agents and therefore higher prices and less empty slots in non peak-hours, more opportunities to show and convert potential click-through into real clicks, lower risk regarding the choices that are taken, more reasons for the advertisers to offer more money when bidding, etc.)*

- An improved user experience *(a richer set of results will have a better coverage of the possible intent of the user, obtaining a higher chance of finding an ad that satisfies the user needs[1])*

- Better CTR estimation for more ads *(more uniform data).*

We introduce a family of algorithms that may apply both for the cases in which just one ad is shown as a sponsored result (single-slot) and when there is a list of $k > 1$ sponsored results (multislot). In very simple terms, our algorithms tries to distribute the available slots among *all* ads, using some kind of proportional mechanism based on the bids and the expected CTRs of the ads. We have analyzed the performance of our algorithms in the case of a static intrinsic CTR associated to each ad and the more general case in which the CTRs of the ads vary dynamically with time. There are many scenarios in which it seams reasonable to consider the latter kind of behavior[2].

Our algorithms implement themselves the explore/exploit trade-off, in a way that can be fine-tuned to regulate the desired weight of each of the stages, but even if there is no will or possibility of changing policies, our algorithms can be seen as a controlled way to implement the "explore" part of other mechanisms, with some kind of performance guarantee. Although we have seen in our experiments that the average revenue obtained by our algorithms is not much lower than that obtained by more standard policies, it is important to note that the new algorithms could be applied *together* with other approaches, interleaving them to form a blend of algorithms according to the required needs of the particular case under study.

In all cases, we compare the performance of the new algorithms with that of the traditional deterministic ones. Our algorithms admit deterministic or randomized implementations, and this flexibility may be an additional advantage, as the best of the approaches could be selected depending on the particular needs. For simplicity reasons in this stage of our research we assumed a first-price auction mechanism [6], though at present we are investigating other pricing schemes with better properties (it's easy to see that our algorithms are compatible with the use of the pricing mechanism defined in [10], leading to a strictly incentive-compatible auction).

The remainder of this paper is organized as follows: We first give an overview of traditional deterministic algorithms and estimators. Then we introduce the first and simplest proportional algorithms, and show that they are subject to some kind of fraud or misbehavior, which motivates the introduction of a second and more complex family of algorithms. In section 5 we show how our algorithms and the whole model can be extended to a multislot environment. Section 6 is devoted to the results of our experiments in the static (fixed CTR) environment. In Section 7 we show how to extend our algorithms to the case in which the CTRs of the ads vary along time, and we describe the design and results of the experiment conducted for that case. Finally, in Section 9 we summarize the results, present conclusions and further work.

2. STATE OF THE ART

Recently there has been an interesting amount of papers studying sponsored search related problems. We cite just a few of them here. An early example is [4], where the trade-off between the advertiser- and used-based revenues for the search engine site is studied. In [7] different allocation mechanisms are proposed and theoretically and experimentally analyzed the impact of the various parameters (revenue model, allocation mechanisms, etc.) on the revenue. In [8], an approach intending to overcome the problem of click fraud [2] is presented, with an idea that, initially, could resemble ours, as the proposed solution consists in selling percentages of impressions instead of clicks. An important difference with our work is precisely that we stay bound to the successful pay-per-click model. In [9] some ways of estimating ads' CTRs that are resistant to click-fraud are presented. In our experiments we used some of the methods presented there. In [11] the problem is explored under a

[1]For example, if a user types "mustang" and we show three diverse ads, one about the Ford Mustang, one about horses, and one about hotels in Mustang, Oklahoma, we may maximize the chance the user will click on at least one of them. Similarly, in a query like "Ford" would be better to show ads for different cars than for the same car.

[2]For example, the search for "china" at certain times of the day may be strongly oriented toward Chinese restaurants or delivery shops, while at other time it may refer to some recent event or tourism in China.

different vision: according to it, the aim consists in optimizing the total revenue for a set of queries, in an on-line manner, trying to consume the maximum amount of each bidder's budget, along a sequence of queries. The result is a best competitive algorithm under this framework. Other articles, like [1] and [6], focus on another fundamental aspect, as are the properties of different auctioning mechanisms that may be applied to sponsored search. In [10], some stochastic allocation models are analyzed and, most importantly, a truthful/strictly incentive-compatible pricing model for them is presented, showing that these kind of mechanisms (in particular ours) are compatible with rational bidder behavior.

3. CLASSICAL ALGORITHMS

Now we present the basic techniques used by search engines up to now. As we mentioned earlier, these algorithms are of a "winner takes all style both for the single and the multislot frameworks, as they simply consist in ordering the ads according to their respective criteria and then selecting the first ads in that order.

In the following we present simplified versions of the typically used algorithms, but before we will introduce the following assumption: Each ad has its own "intrinsic" CTR, that is, the probability that the ad would be clicked if presented as a result of a search. We assume that probability holds when the ad is presented as the only result in a single-slot framework or as the first result of the list in a multislot framework. In Section 5 we will explain how the intrinsic CTR defines the probability of the ad receiving a click when presented in a different position of the list of results. The assumption of intrinsic and static CTRs will be removed in Section 7, when we extend and analyze our algorithms in a dynamic framework, in which the CTRs of the ads vary along time.

From now on, we will have the ads relevant to a query numbered consecutively, and denote by bid_i the bid that is placed for ad i, by ctr_i the CTR of the ad and by $ectr_i$ its estimated CTR.

The typical algorithms are:

- **MaxBid**: Choose the ad i that maximizes bid_i.
- **MaxEstRev (MER)**: This is a family of algorithms in which for each ad i at each time there is an estimation $ectr_i$ of its CTR. Each algorithm of this family chooses the ad i that maximizes the product $bid_i \times ectr_i$ (i.e., the expected revenue), and algorithms differ from each other in how the estimation is done.

Some of the algorithms for CTR estimation present in the literature, are (descriptions extracted from [9]):

- Total average: Let x be the total number of clicks received by the ad, and y be the total number of impressions, then the estimated CTR is x/y.
- Average over fixed time window: For a parameter t let x be the number of clicks received by the ad and y be the number of impressions during the last t time units, then the estimated CTR is x/y.
- Average over fixed impression window: For a parameter y let x be the number of clicks received by the ad during the last y impressions, then the estimated CTR is x/y.
- Average over fixed click window: For a parameter x let y be the minimum number of last impressions of the ad that contain x clicks, then the estimated CTR is x/y.
- Exponential discounting: For a parameter α, let $e^{-\alpha i}$ be a discounting factor used to weigh the i^{th} most recent impression. Take a weighed average over all impressions, that is, $\sum_i x_i e^{-\alpha i}/\sum_i e^{-\alpha i}$ where x_i is an indicator variable that the i^{th} impression resulted in a click.

Besides the previous algorithms, we will consider the value **MaxRev** that would result from choosing the ad i that maximizes the product $bid_i \times ctr_i$, where ctr_i is the real (not estimated) CTR of ad i. This is not really a feasible algorithm, as that value is not known a priori. However, it serves as a benchmark to compare the behavior of other algorithms in different scenarios. On the other hand, the better an estimation method for the CTR is, the more alike MaxRev will behave like an instance of MaxEstRev.

4. PROPORTIONAL ALGORITHMS

This family contains algorithms that, instead of assigning the next impression of a keyword to the ad that maximizes a certain function (like all the algorithms of the previous section), selects the ad to be shown following a probability distribution, which in turn will depend on some parameters like the bids, the CTR (or expected CTR) and so on. Eventually, this probability distribution could be transformed in a frequency (# of impressions / # of searches for the keyword), so the model has its deterministic version too. For simplicity reasons, we just work with the probabilistic version of the algorithm.

We first present two simple algorithms that capture the spirit we are trying to introduce. We will then see some problems that arise from these definitions, and will show two further algorithms that overcome these problems.

- **SimpleBid Proportional (SBP)**: In this algorithm, the probability of selecting an ad i (or the frequency of selection of ad i) is computed as $bid_i/\sum_j bid_j$, where the sum $\sum_j bid_j$ is taken over the set of all ads j for the same term.
- **SimpleBidCTR Proportional(SCP)**: In this family of algorithms, the probability of selecting ad i depends not only on bid_i but on ctr_i. As the real CTR is not known, there will be a different algorithm for each way of estimating the CTR (in particular, there will be one variant that will be computed using the "real" CTR as a benchmark). Given the estimator of the CTR, algorithms in this family will compute the probability of selecting ad i as $(bid_i \times ectr_i)/(\sum_j bid_j \times ectr_j)$.

4.1 Frauds

The previous two algorithms suffer from an important drawback, that is, they could be subject to two particular types of fraud:

1. Split fraud: Many smaller bids may account for more frequency than one bigger bid.

2. Multiple bid fraud: A malicious bidder could replicate its bids to "buy" a higher frequency of appearance, without paying more for it.

SBP algorithm is subject to both types of fraud, while SCP suffers only of Multiple Bid Fraud. Actually, to show this we need to make some assumptions on the effect that multiple appearances of the same ad would have on its CTR[3], as well as on the capacity of the provider to detect this kind of frauds. We will now propose different algorithms that behave much better against this frauds. Moreover, we will soon argue that also classical algorithms are subject to multiple bid fraud.

Our result regarding split fraud is applicable to a more general class of algorithms: we can show that any probability function that is constructed by assigning a score to each ad independently of the other ads (with a function that is monotonically increasing on the bid, which is the only reasonable thing to do, as it is assumed on [10]), and then taking the probability proportional to that score, is generally subject to both types of fraud. Both SBP and SCP are particular cases of this general property[4]. To prove this, let f be any monotonically increasing scoring function used to define the assignment of probabilities $P :: bid \rightarrow [0, 1]$ $(P(x) = f(x)/\sum_y f(y))$. Let x be a current bid, z the minimum possible bid and $k = \lceil f(x)/f(z) \rceil$. The score achieved by doing k bids of value z is $k \cdot f(z)$ which is by definition of k greater than or equal to $f(x)$ and therefore, the frequency achieved by the bidder did not decrease while the payment per click is reduced to its minimum value z. When the algorithm (the function f) is also dependent on CTR estimations, things get more complicated because it is not clear how the repetition of ads affects the CTRs or its estimations. However, we can conjecture that a similar advertiser behavior would still hold. The proof would be similar to the previous case, because the CTR estimation would fit in the previous equations (to some extent) as a constant.

Even when split is controlled, many functions will have the problem of multiple bidding. With the previous algorithms an advertiser is likely to try to increase his frequency by repeating his bid many times. This behavior does not impact *directly* in the revenue, so it seems not to be as bad as split. Actually, in the limit case (all bidders repeating their respective bids) the model would tend to be equal to the algorithm that always chooses the maximum bidder, but that would be against our goal of increasing the overall diversity of the results. Furthermore, it's very important to note that this problem seems to appear in all current models also if there is no kind of editorial control of the ads: in the multislot framework, the same bidder could take, by repeating his bid, all slots. As users are not likely to click twice on the same advertisement[5], he would get all the attention without paying for it, decreasing the overall click-trough rate and henceforth the revenue (besides the "second-order" negative effects on users and other advertisers).

4.2 Beyond Split: Accumulated Proportional Algorithms

The previous analysis leads us to the formulation of a more sophisticated mechanism to achieve the idea of proportionality, defining the probabilities in such a way that higher bids consistently determine higher frequency of appearance, and therefore there is no incentive for the bidders to split their bids. Concretely, we will define the probabilities with a procedure that warranties that the probability of each ad is, on one side, greater than that of all ads with smaller bid together, and on the other side at least proportionally greater than that of each smaller bid.

The first of these algorithms works as follows:

- AccumBid Proportional (ABP): In this algorithm, the probability of selecting ad i is computed by the following procedure:
  1. Let $b_i = bid_i$
  2. Sort the b_i in increasing order.
     Let $v_1, v_2 \ldots v_k$ be all the different values of the b_i
  3. Take all the ads with minimum b_i. Assign to each one of these ads a weight $w(i) = 1$
  4. For $i = 2$ to k
     Let S_i be the sum of the weights of the ads with value smaller than v_i. Let $v = S_i \times v_i / v_{i-1}$. Assign to each one of the ads j of value v_i a weight $w(j) = v$.
  5. Assign to each ad i a probability $p(i) = w(i)/\sum_j w(j)$.

In the second algorithm we follow the same procedure, but instead of using the bids, we use the product of the bid and the estimated CTR (i.e., the expected revenue). Therefore the algorithms (a different one for each of the ways of estimating the CTR) are as follows:

- AccumBidCTR Proportional (ABCP): In this algorithm, the probability of selecting ad i is computed by the same procedure as before, just changing the first line:
  1. Let $b_i = bid_i \times ectr_i$

We note that both previous algorithms can be computed in linear time with a simple dynamic programming algorithm, provided that the b_i are sorted.

Table 1 shows an example of the probabilities assigned by algorithm ABP.

This algorithms are immune to split because each bid is assigned a score (value) that is greater than the sum of all smaller bids (because that sum is multiplied by a number greater than 1), so any number of smaller bids would lead to a lower frequency of impressions. The probability is strictly increasing with the bids, so there's always a reason to bid more[6].

[3] In more general terms, algorithms that take the CTR into account are less subject to Multiple bid fraud due to the natural assumption that an ad appearing repeatedly should negatively affect the estimated CTR of each individual instance of the ad.

[4] If we assume that simultaneous impressions of the ad will affect the CTRs, by using them in the construction we preclude the score function of being independent of the other ads.

[5] Unless the ad is disguised as something different, which is not good for the advertiser.

[6] Under some pricing schemes, we can show that the mechanism is truthful (incentive-compatible), so that bidders have reasons to bid precisely their true value for the item.

b_i	2	2	3	4	4	6
w_i	1	1	3	$\frac{20}{3}$	$\frac{20}{3}$	$\frac{55}{2}$
$p(i)$	$\frac{6}{275}$	$\frac{6}{275}$	$\frac{18}{275}$	$\frac{40}{275}$	$\frac{40}{275}$	$\frac{165}{275}$
$\sim$ % of impr.	2.2%	2.2%	6.5%	14.5%	14.5%	60%

Table 1: Example of probabilities assigned by AccumBid Algorithm. $\sum_j w(j)$ is $\frac{275}{6}$

The model is not totally robust against multiple bids, but has some good properties, as we will see.

The way the scoring function is constructed places an upper bound in how much frequency of impression an ad can obtain by staying at position i in the ranking of bidders for a particular query. Let us suppose that a malicious bidder bids m times with a bid of x or less, and that a bid of x results in position i in the ranking of bids. Now, let $G_x = \{x' \in b_1, \ldots, b_k \mid x' > x\}$ be the set of bids greater than x. By hypothesis, $|G_x| = i - 1$. Let S be the sum of the weights of the m malicious bids, and T the sum of the weights of all the bids. Then, the weight that each member of G_x will receive when the algorithm is applied is greater than S, and the aggregated probability for the m bids together will then be S/T. Since T includes the value for the m bids and the score for each bid in G_x, we have that $T > S + (i-1) \cdot S = i \cdot S$, so the probability is at most $\frac{S}{T} < \frac{S}{i \cdot S} = \frac{1}{i}$ and therefore, staying at position i of the ranking a malicious bidder with any number of bids of any value can have at most a frequency of $1/i$.

This means that even if the use of multiple bids is still possible, there is an incentive for bidders that are not on top of the ranking to increase the bid to get a higher rank and therefore have a higher probability of impression.

On the other hand, we can also give a lower bound to the probability of the winner. Assuming that there are no ties in the values, the probability assigned to the ad with the highest value will be at least $\frac{R}{1+R}$, where $R = v_k/v_{k-1}$ is the ratio between the first and second highest values respectively. This means that the first ranked ad's probability will be at least $1/2$, and will approach 1 as the ratio among the values increases. Similarly, each following ad will have a probability of at least one half of the remaining probabilities.

5. PROPORTIONAL ALGORITHMS IN THE MULTIPLE SLOT MODEL

The algorithms presented in the previous section apply directly to the single slot model, in which just one ad is shown as a result of each query. In this section, we will introduce an extension of the previous algorithms to the case in which more than one ad is shown, that is the more typical situation. We first show the model that we will consider, and then the extension of the algorithms. The latter will include, as a fundamental component, the extension of the CTR estimation mechanisms that each algorithm will use.

The idea behind this generalization is that there are many "slots" that are to be filled with ads. The previous model in which one ad was displayed for each query can then be seen as a particular case with just one slot.

We propose the following model, assuming there are k available slots, numbered from 0 to $k-1$, from top to bottom:

- For each query, we will show k different ads. If there are less ads than available slots, then some slots (the bottommost ones in the page, the highest in number) will remain empty.
- Each shown ad will receive (or not) a click independently of the other ads shown.
- Let ctr_i be the intrinsic CTR for the ad i. Then we assume that if the ad i is displayed in position j, it will be clicked with a probability $ctr_i \times \alpha^{-j}$, for some $\alpha > 1$. This is usually called "exponential decay"[5, 7]. We will use $\alpha = 2$ for simplicity.

As it can be easily seen with this model, the expected revenue for n slots is measurably greater than the expected revenue for one slot, a desirable property, and is also upper-bounded by $1+1/(\alpha-1)$ times that value, which is also reasonable since too many slots should not lead to unbounded revenue.

5.1 Generalization of the Algorithms

An algorithm for a multiple slot model is a procedure that decides which ad will be displayed in each one of the k slots as a result of a query to the term.

- Maximum Expected Revenue (MER): The extension of this algorithm to the multiple slot model is pretty straightforward. Simply choose the k ads that maximize the expected revenue, i. e. put the ad with j^{th} highest expected revenue in the $(j-1)^{th}$ slot, for $j = 1..k$. This clearly maximizes the overall expected revenue. Recall that expected revenue is computed as the product $bid_i \times ectr_i$ for each CTR estimator.
- Proportional Algorithms: We will extend all proportional algorithms in a similar way. All our proportional algorithms are based in procedures to compute a probability of impression in a different way. There are two simple ways of extending these procedures to the multislot framework. Both ways imply selecting an ad for the first slot, removing it from the list of possible ads, selecting another one for the second slot, removing it, and so on.

The two variants for computing the probabilities of each ad are:

- Recalculate values after each removal

- Do not recalculate values

This two variants only differ for the Accumulated Proportional algorithms. The Simple proportional algorithm or any algorithm in which the value for each ad is independent from the other ads coincide in both variants. We implemented the version without the recalculation, because it is more similar to the single-slot version, but it is expected that both alternatives should work in a pretty similar fashion in practice.

CTR Estimators in Multiple Slot Models

To continue with the definition of the multislot model, we need to extend the behavior of the CTR estimators in order to correctly estimate the intrinsic CTR. Due to the exponential decay property, the probability of being clicked in a position that is not the first one is different from the ad's CTR. Therefore, using the same approach of the single slot model would fail because not all experiments (where an experiment is defined as the impression of an ad and the result of it, clicked or not clicked) are equal.

Suppose first that we estimate the CTR of a given ad a at a given position i, meaning that we take each position individually. If the estimation works properly and we have an adequate number of experiments for each position, the estimated CTR for a at position 0 should be α times the estimated CTR of a at position 1, and similarly for all positions (the estimated CTR based on experiments on position i should be $1/\alpha^i$ times the expected CTR on position 0). So, we can use this information to put all experiments together and then have more accurate estimations (assuming that more experiments lead to a better approximation) of the intrinsic CTR. The point is how to weight the information obtained at each position.

The first possibility would be to obtain the estimation of the intrinsic CTR by simply taking the average of all the partial estimations, each one multiplied by the corresponding α^i. Therefore, if there are k slots and the estimation for slot i is p_i, the overall estimated CTR would be $(\sum_{i=0}^{k-1} p_i \times \alpha^i)/k$.

While this seems to be reasonable at first glance, one problem remains. Since each p_i is estimated individually, there could be big differences in the quality of the estimations for the different positions (for example, imagine a given ad is printed only once in the fifth slot and 1000 times in first place. In this scenario, taking the plain average between those two estimations would make the estimated CTR much less reliable than it could be considering the 1001 impressions). Then, we propose to take a weighted average, where the weight is simply the number of impressions in which the estimation is based for each slot. With this improvement, the estimated CTR will be $(\sum_{i=0}^{k-1} I_i \times p_i \times \alpha^i)/I$ where I is the total number of impressions of the ad (considering all slots), and I_i is the number of impressions for the ad at position i (so $\sum_{i=0}^{k-1} I_i = I$). Then, since $p_i = C_i/I_i$ with C_i the number of clicks at position i, the estimated CTR becomes $(\sum_{i=0}^{k-1} C_i \times \alpha^i)/I$, meaning that the estimation remains the same as in the single slot model, but clicks add an exponentially scaled amount to the "number of clicks".

Note that this estimation is no longer a probability function, for its sum may exceptionally exceed 1, but rather a score function that is suitable whenever the exponential decay assumption holds.

In addition, since each p_i can be obtained using any of the methods described earlier, this idea allows us to extend all of them to the multislot environment.

6. EXPERIMENTS AND RESULTS FOR STATIC CTRS

In this section we summarize the experiments done to test the new proposed algorithms in a static framework. The following are the main points of the setting for the single-slot case:

- Each experiment consisted of a series of 5.000 simulated queries on a single keyphrase, each one yielding a single element as a result from a set of 10-15 ads. Complementary tests were run on a base of 20-30 ads to choose from.
- Simulations were run on 10 distinct keyphrases, using per keyphrase 10 random CTR assignments, for a total of 100 simulations per batch.
- Each ad had its own fixed CTR, immutable over time. Random CTR assignment followed uniform, normal and power-law distributions with arbitrary but reasonable parameters for different data sets.
- Bids for each ad followed static data provided by Yahoo!.
- Different CTR estimation methods were employed.

The first half of Table 2 summarizes average behaviors of the algorithms that take the CTRs into account relative to the optimum hypothetical revenue, represented by the use of algorithm MER with known CTRs. The first block of results corresponds to the algorithms MER, ABCP and SCP using the "real" CTRs, as if they were given by an "oracle" (in which case MER would consist in choosing all the time the ad that maximizes the product bid times CTR, and therefore obtain a 100% of effectiveness). We can see that the loss of revenue due to showing periodically all the ads is of ~10% of the optimum on average. The second block of table 2 shows that when CTRs must be estimated (which is almost always the case in practice), the performances of ABCP and MER are similar w.r.t. the revenue. But while known CTR forces MER to show only one ad, both SCP and ABCP show almost always almost all the ads.

The fact that ABCP obtains such a good performance and with a much greater variety is the main result of our experiment.

CTR estimations can lead MER to choose an ad not maximizing revenue, and showing it almost all the time, while proportional methods do not seem to suffer from this inconvenient. Moreover, when CTR is estimated, MER is less stable than ABCP (in the statistical sense, i.e., the statistical variance of the results of MER is greater than that of ABCP). Estimating the CTR, MER's mean was never more than 10% than ABCP's, but in some cases it was a 25% worse. This holds both for the single and multislot frameworks.

Additional conclusions that were obtained from the results of our experiments are:

Case	Single slot						Multiple slot					
CTR	known			estimated			known			estimated		
Alg.	MER	ABCP	SCP	MER	ABCP	SCP	MER	ABCP	SCP	MER	ABCP	SCP
min	100%	79.0%	50.3%	1.1%	13.4%	16.8%	100%	87.1%	64.5%	9.6%	24.3%	40.8%
mean	100%	89.6%	69.5%	81.7%	80.7%	72.9%	100%	93.1%	75.8%	82.8%	83.8%	77.9%
max	100%	99.1%	91.5%	100%	98.4%	97.0%	100%	98.5%	91.7%	100%	98.0%	96.6%

Table 2: Single and multiple slot performance compared to optimum.

- Algorithms without CTR estimation perform poorly: their average revenue was about ∼60% of optimum.
- Results vary greatly among CTR estimators. The selection of the best estimators deserves further work, that we are currently developing.
- The variance of the performance of different algorithms is not uniform. SCP shows the minimum dispersion, but most importantly, ABCP has a smoother behavior than MER.
- ABCP consistently outperformed SCP.
- When considering a larger amount of ads (between 20 and 30 ads per keyphrase) it was found that the revenue dispersion slightly diminishes for all algorithms, that the mean performance of ABCP is improved by ∼5% of optimum under both known and estimated CTR conditions, and the average revenue gap between MER and ABCP narrows for all CTR estimators.
- Regarding the variety of ads, ABCP always assigns more than 50% of the impressions to the ad maximizing revenue but never gave it more than 66%; ∼45% of the ads were shown less than 1% of the time and ∼30% were shown near or above 10% of the time.
- SCP shows over ∼80% of the ads more than 1% of the time, near 50% at least 10% of the time but none above 50%, both with known or estimated CTR.

Multiple Slot Results

For the Multislot case, the experiment was based on a similar setting, with the additional definitions:

- Each simulated query returns a 5-element list from a set of 10-15 ads.
- Click probability uses *exponential decay* model with factor $\alpha = 2$.

The results for the multislot case are even better for ABCP than with one slot, and they are shown in the second half of Table 2. In short:

- With estimated CTR usually ABCP has an advantage of ∼1% over MER.
- With known CTR, the results provided by MER are around 7% better than ABCP.
- The performance of algorithms ignoring CTR is poor.
- As for algorithms that consider CTR, ABCP outperforms SCP and MER is less stable than ABCP.

7. DYNAMIC CTRS

Now we will deal with the more general case in which the CTRs of the ads vary dynamically with time. As we explained in the Introduction, this variation may occur in practice due to many different reasons, from different user needs at different times of the day to the appearance of a banner ad in the same or other web page, some news published in the newspaper, etc.

The most complex issue in this part of the work was to define a pattern of variation of the CTRs. We chose to use the following model: at the beginning each ad is assigned an initial CTR following one particular distribution, and the CTR of ad i at time t is computed as a function of the CTR of that same ad at time $t-1$ plus a random perturbation that follows a Normal distribution. This is a stochastic process generally known as a *Wiener Process*. Namely, if we denote by $CTR_i(t)$ the probability of ad i being clicked if presented at time t, we have that

$$CTR_i(t) = CTR_i(t-1) + X$$

where X is a random variable that follows a distribution $N(0, \sigma^2)$. The value of σ^2 determines the speed of the variation[7]. We decided to test three different variation speeds, with the spirit of reflecting three different kinds of environments, namely slow, medium and rapidly changing environments. The values of σ^2 where respectively 0.001, 0.002 and 0.004. The initial CTRs were defined using a power-law distribution, namely we set $CTR_i(0) = 0.2 * U^2$, where U is uniformly distributed in [0..1].

We decided to extend to the dynamic framework and compare the behaviors just of the two better performing algorithms, namely MER and ABCP. The extension of both algorithms to handle dynamically changing CTRs is straightforward, as both are based on computing a certain function on the bids (that where kept unchanged w.r.t. the previous experiment) and the expected CTR (the computation of which deals "only" with clicks and no-clicks and hence remains unchanged). Obviously, the estimation itself becomes more interesting and difficult in the dynamic framework.

[7]Actually, there are some technical details that deserve to be cleared: as we have defined it, $CTR_i(t)$ could be greater than 1 (though with the actual values that situation would be very unlikely). We decided not to "truncate" that number until needed, that is, until the ad is "chosen" and we have to compute the probability of click. In that eventuality, an ad with $CTR > 1$ would be clicked at with probability 1. Even then, it's "good" for an ad to have its $CTR > 1$, because an eventual later decrease could still leave him with a $CTR > 1$. This is more important in the multislot framework, as the computation of the estimated CTR includes the division by the decay factor. Similar arguments apply to possible cases in which $CTR < 0$.

Finally, we kept on experimenting with single- and multislot frameworks separately, despite the fact that for the static case the results obtained for both frameworks where pretty similar.

8. EXPERIMENTS AND RESULTS WITH DYNAMIC CTRS

The basic setting of the experiment is summarized here:

- Each experiment consists of a series of 5.000 simulated queries on a single keyphrase, each one yielding a single element as a result from a set of 10-15 ads.
- Simulations were run on 10 distinct keyphrases, using 10 initial random CTR assignments per keyphrase, for a total of 100 simulations per batch.
- The CTR of the ads vary over time as described above following three different patterns in each case (slow, medium and rapid variations).
- Bids for each ad follow data provided by Yahoo!.
- Different CTR estimation methods are used.
- For the multislot environment, each simulated query returns a 5-element list from a set of 10-15 ads, and click probability uses an exponential decay model with factor $\alpha = 2$.

The results of the experiments for both, single and multiple slot cases, and the medium variation speed, are given in Table 3. They show that the results for the dynamic environment with small variation speed resemble those obtained in the static case. As variation speed increases, the performances of both algorithms decay but, interestingly, ABCP outperforms MER on average for estimated CTR. However, there are certain important particularities that deserve to be mentioned:

- When using the "exact" CTRs instead of estimations (hypothetical optimum that we use as a benchmark) ABCP maintained the same performance as in the static single slot case, namely ∼86% of optimum for all speeds of variation.
- With the *Total average* estimator, ABCP's performance is slightly worse than in the static case while the negative effect of changes in the CTRs seems to be much stronger for MER (falling from around ∼100% in the static case to ∼59% of optimal in the rapid variation framework). This may be due to the fact that the quality of this estimator is naturally better when there are less changes in the CTR values, and this affects more MER than ABCP (because of the tendency of the latter to give more opportunities to all ads).
- For the *time window* estimator, performance of both algorithms decays, but the effect on MER is more negative, and we have that ABCP's average is better than MER's for medium and rapid speed variations.
- Using the *click window* estimator, we found that the results are rather similar to the static case, though both algorithms performances against the optimum decrease slightly as the speed of variance of the CTRs increases, reducing the difference between both algorithms to less than ∼5% of optimum.
- The estimator that produces the best results is *exponential discount*. The revenue for both ABCP and MER is over ∼80% of optimum with slight variations depending on the parameters of the estimator. For the rapid variance, ABCP performs better than MER.

The results for the multislot case are, in general, rather similar to the single slot ones.

9. CONCLUSIONS AND FURTHER WORK

We have introduced and tested a family of algorithms for sponsored search that is designed with the goal of increasing the diversity of the selected ads without reducing the revenue. Although it is known that under certain assumptions the optimal placement mechanism is deterministic [12], randomization can be helpful if those assumptions do not hold (for example, the advertisers base is not fixed) or just to explore how much you loose by optimizing other features like diversity.

The results of our experiments, conducted under different environments regarding the distributions of the bids and the CTRs, show that the intended goal is achieved by at least one of our algorithms, namely ABCP. We think that the results are promising, as there is a wide margin for potential improvement on our algorithms by fine-tuning some of its parameters to obtain still better results under particular environments. In particular, ABCP could be easily extended to allow adjusting the terms of the explore/exploit trade-off according to the needs.

A nice property of our algorithms is that they do not need to be used exclusively, but they could provide a controlled and non-costly way of conducting the exploration phases as part of a more general algorithm. Moreover, the possibility of implementing the algorithms in deterministic or randomized fashion, for one or many slots of sponsored results, for fixed and varying quantity of slots, and for alternative models like scrolling ads, and with different CTR estimation methods, increases the value of the idea.

Our algorithms have an additional nice feature in that they are naturally extensible to a framework in which a certain number of ads (larger than k) scroll in the screen of results instead of being static and fixed. However, we have not studied this extension yet.

The next steps in our research is to take into account bidder behavior in front of the model. We will consider other more elaborated pricing schemes (second-price and stochastic auctions) that are compatible with our model and have attractive properties regarding this issue. We also want to revise the exponential decay hypothesis, as with a small number of slots, an exponential model is similar to a power law model. However, it is well known that the click distribution follows a power law model in the answers of a search engine [3].

Case	Single slot				Multiple slot			
CTR	known		estimated		known		estimated	
Alg.	MER	ABCP	MER	ABCP	MER	ABCP	MER	ABCP
min	100%	80.7%	10.2%	22.9%	100%	88.9%	22.1%	37.8%
mean	100%	85.6%	58.3%	71.4%	100%	91.4%	70.1%	72.3%
max	100%	92.7%	99.9%	98.8%	100%	95.1%	94.6%	91.1%

Table 3: Dynamic single and multiple slot performance compared to optimum.

Acknowledgements

We thank the helpful comments of David Pennock and Michael Schwartz.

10. REFERENCES

[1] Gagan Aggarwal, Ashish Goel, and Rajeev Motwani. Truthful auctions for pricing search keywords. In *ACM Conference on Electronic Commerce*, pages 1–7, 2006.

[2] Vinod Anupam, Alain J. Mayer, Kobbi Nissim, Benny Pinkas, and Michael K. Reiter. On the security of pay-per-click and other web advertising schemes. *Computer Networks*, 31(11-16):1091–1100, 1999.

[3] R. Baeza-Yates. Query usage mining in search engines. *Web Mining: Applications and Techniques, Anthony Scime, editor. Idea Group*, 2004.

[4] H.K. Bhargava and J. Feng. Paid placement strategies for internet search engines. In *WWW*, 2002.

[5] J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[6] B.G. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. Stanford Graduate School of Business Research Paper No. 1917 Available at SSRN: http://ssrn.com/abstract=861164, 2005.

[7] J. Feng, H.K. Bhargava, and D.M. Pennock. Implementing sponsored search in web search engines: Computational evaluation of alternative mechanisms. *Informs Journal on Computing*, 2006.

[8] J. Goodman. Pay-per-percentage of impressions: an advertising method that is highly robust to fraud. In *Proc. ACM E-Commerce Workshop on Sponsored Search Auctions*, 2005.

[9] N. Immorlica, K. Jain, M. Mahdian, and K. Talwar. Click fraud resistent methods for learning click-through rates. In *Workshop on Internet and Network Economics (WINE)*, 2005.

[10] C. Meek, D.M. Chickering, and D.B. Wilson. Stochastic and contingent-payment auctions. In *Workshop on Sponsored Search Auctions - ACM Conference on Electronic Commerce (EC'05)*, 2005.

[11] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized on-line matching. In *Proc. Symposium on Foundations of Computer Science*, 2005.

[12] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.