Bid Generation for Advanced Match in Sponsored Search

Andrei Broder, Evgeniy Gabrilovich, Vanja Josifovski, George Mavromatis, Alex Smola
Yahoo! Research, 4301 Great America Parkway, Santa Clara, CA 95054, USA
{broder, gabr, vanjaj, gmavr, smola}@yahoo-inc.com

ABSTRACT

Sponsored search is a three-way interaction between advertisers, users, and the search engine. The basic ad selection in sponsored search, lets the advertiser choose the exact queries where the ad is to be shown. To increase advertising volume, many advertisers opt into *advanced match*, where the search engine can select additional queries that are deemed relevant for the advertiser's ad. In advanced match, the search engine is effectively bidding on the behalf of the advertisers. While advanced match has been extensively studied in the literature from the ad relevance perspective there is little work that discusses how to infer the appropriate bid value for a given advanced match. The bid value is crucial as it affects both the ad placement in revenue reordering and the amount advertisers are charged in case of a click.

We propose a statistical approach to solve the bid generation problem and examine two information sources: the bidding behavior of advertisers, and the conversion data. Our key finding suggests that sophisticated advertisers' bids are driven by many factors beyond clicks and immediate measurable conversions, likely capturing the *value chain* of an ad display ranging from views, clicks, profit margins, etc., representing the total ROI from the advertising.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Retrieval

General Terms

Algorithms, Economics, Experimentation

Keywords

Optimization, Sponsored Search, Machine Learning

1. INTRODUCTION

Displaying ads alongside Web search results, i.e. *sponsored search* is a key financial driver of the Internet economy. It provides traffic to millions of Web sites, and accounts for a large portion of the $30 billion online advertising spend.[1]

[1]eMarketer.com, estimates for 2009

WSDM'11, February 9–12, 2011, Hong Kong, China.

Historically, the ad selection process in sponsored search was delegated to the advertiser. For each ad, the advertiser explicitly listed the *bid phrases* — queries for which the ad is to be shown. This *exact match* mechanism allowed for the search marketplace to grow quickly without the need for search engines to address the complexities of ad selection. In addition it also gave the advertiser nearly complete control of the ad placement. However, limits of this mechanism quickly became apparent, making a pure exact match system undesirable. First, the query volume distribution follows a power law with a long tail [6], making it impossible for advertisers to *know* and to explicitly enumerate all queries which are commercially relevant for their ad. This results in lost revenue for the search engine and in lower volume of traffic for the advertisers due to their inability to target all relevant tail queries. Next, it is inconvenient for advertisers to specify *explicitly* all possible matches. For smaller advertisers the work to enumerate all matches may outweigh the benefit. Finally, many advertisers are *not* privy to advanced details of the bid and conversion landscape due to limited volume. Their bids for tail queries would be suboptimal. To address this limitation, *advanced match* was introduced, which allows search engines to match ads to reasonably related queries (with advertiser's permission). One challenge is that matching can no longer be solved by simple record lookup [2, 8] — information retrieval techniques are used instead. However, a major point remains unresolved — if the advertiser no longer explicitly bids on every query, the search engine needs to *automatically generate bids* which accurately reflect the *advertiser's known bidding behavior*. The bid is very important in the subsequent steps of the ad serving that determine the ordering of the ads, and the amount that the advertiser is charged using an auction, as reviewed later in the paper. Advanced match is a well established technique for sponsored search advertising used by millions of advertisers and responsible for billions of dollars of search engine revenue. The paper contributes the following:

1. We formalize the problem of bid generation for advanced match. While previous work [2, 8] addressed the issue of ad relevance, this is the *first reported study* that addresses generating a bid for advanced match as deployed in current search engines. The mechanism is crucial since the auctioneer (search engine) is effectively bidding on behalf of auction participants.

2. We propose machine learning methods for bid generation, and formulate a regression problem by predicting new bids from existing ones in a large real-life corpus.

3. Our experiments using real advertising data from a major search engine show that the proposed method can very accurately predict the bids of actual ads. Furthermore, we should that using the bid data of the sophisticated advertisers, in most cases results in better prediction than using the conversion data. This suggests that this bid data captures value from the other steps of the funnel, as ad views and clicks.

Outline. We begin with a discussion of textual advertising on the web in section 2. This is followed by a discussion of the estimation problem proper, where we describe *What* estimated in section 3. Basic machine learning methodology to implement a solution, i.e. the *How* the bids are estimated, is discussed in Section 4 and followed by an overview over sample weighting (Section 5). We explain the set of features used for estimation in Section 6 and experimental results are presented in Section 7. We conclude with a discussion.

2. TEXTUAL ADVERTISING ON THE WEB

2.1 Sponsored search

Sponsored search is an interplay of the following three entities: The **advertiser** supplies ads. His goal can be broadly defined as promotion of products or services. The **search engine** provides "real estate" for placing ads (i.e., it allocates space on search results pages), and selects ads that are relevant to the user's query. **Users** issue queries and examine the search result page composed of web search results and sponsored search ads.

The prevalent pricing model for textual ads is that advertisers pay per click (PPC) on the advertisement. The amount for each sponsored search click is usually determined by an auction process [4]. The advertisers place *bids* on a search query, and their position in the column of ads displayed on the search results page is determined by their bid via a generalized second price auction. Thus, each ad is annotated with one or more queries or *bid phrases*.

In the model currently used by all the major search engines, bid phrases serve a dual purpose: they explicitly specify queries that the ad should be displayed for, and simultaneously define the marketplace for the auction that determines the price of ad clicks. Figure 1 shows an overview of today's sponsored search engines. The user query is analyzed and two separate queries are submitted, one to the exact match selection, and another the advanced match ad. For example, for exact match, light stemming and reordering can be performed. The advanced match query can be expanded with optional terms or query rewrites. Both queries are evaluated by the respective layers. In the case of exact match, the bid is specified by the advertiser. In advanced match, the bid is determined by the bid generation process that is the focus of this paper. Once all ads have associated bid, the final slate is composed by revenue reordering and the cost for the advertiser is determined by the auction mechanism.

Textual ads are composed of visible and invisible components (to the user). The visible components are the *title* usually displayed in bold font; the *creative* which is the few lines of text shown to the user; and the display URL that is shown to the user under the ad. Besides a bid phrase, the invisible components are the full URL to the advertised web page and the web page itself, also called the *landing page*. While textual ads appear as individual units to the user,

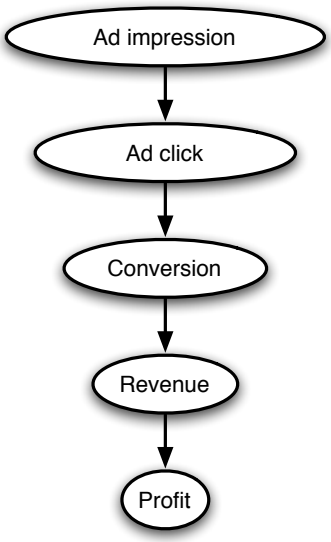

Figure 3: Advertiser utility funnel from ad views to profit. (Indirect) benefits are obtained at every step of the funnel rather than just at the end.

in practice ads are hierarchically defined in a nested structure of several entities, as shown in Figure 2. At the highest level, each advertiser has one or more *accounts*. Within an account, usually the activity of the advertisers is organized around one or more *campaigns*, which are defined by a set of ads with a particular temporal or thematic goal (e.g. the "New Year" and "Black Friday" campaigns in Figure 2). Campaigns consist of *ad groups*, which can have multiple *creatives* and multiple bid phrases. In Figure 2 an ad group promotes appliances within the Black Friday campaign.

An ad, as seen by the user, is a particular combination of a creative and a bid phrase. Any creative can be paired with any bid phrase in the same ad group. In some cases the creatives are templated and can be filled in with the chosen bid phrase at runtime. This type of ad schema has been designed with the advertisers' needs in mind, as it allows the advertisers to easily define a large number of ads for a variety of products and marketing messages. Each bid phrase can be a different product or service offered by the advertiser. Different creatives represent different ways to advertise those products. Usually the number of creatives is limited to a few dozens, while each ad group can have hundreds or even thousands different bid phrases.

2.2 A motivating example

We begin our discussion with a small motivating example, considering a contractor advertising his services. He may be willing to pay a small amount when his ads are clicked from general queries such as "home remodeling", and higher amounts if the ads are clicked from more focused queries such as "hardwood floors" or "laminate flooring". Since he may not be privy to information about *all* possible related queries that would provide him with business opportunities, he may choose to opt into advanced match to benefit from queries he may have overlooked, such as "marble flooring", or tail queries such as "distressed hickory wood flooring".

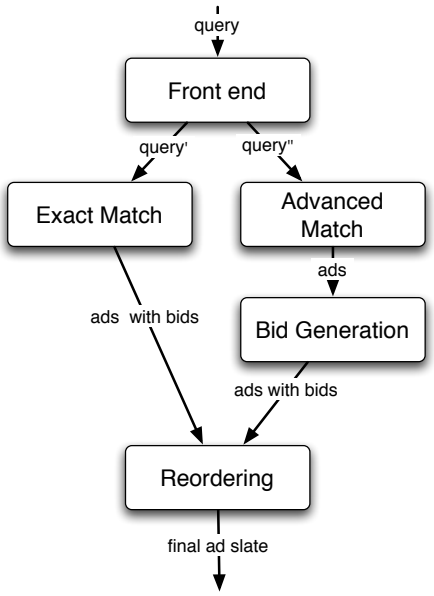

Figure 1: Sponsored search workflow for advanced match of textual ads.

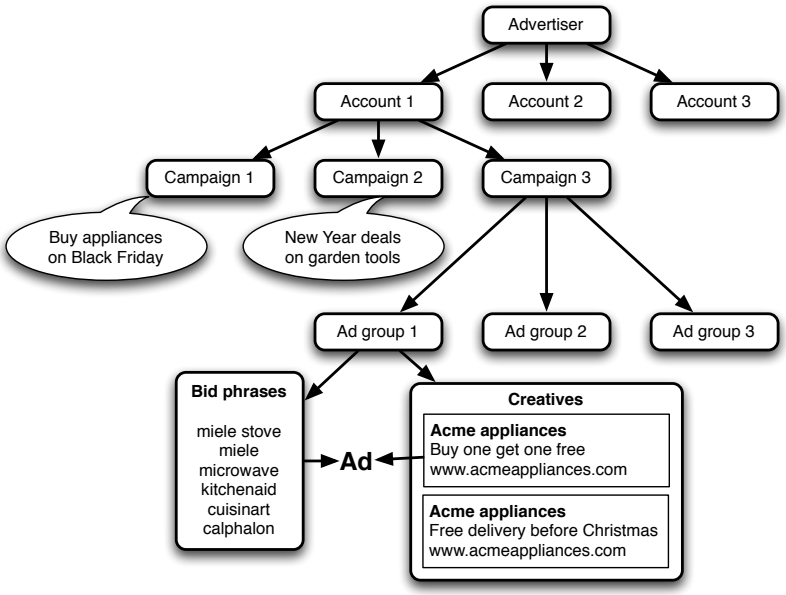

Figure 2: Ad database schema — an advertiser has several accounts and campaigns within each account. Typically ad groups are topically coherent.

Even if he knew of these opportunities he might not know how much to bid for them due to lack of skill or data.

His benefit in advertising may vary widely from building his brand (impressions) and disseminating information about his trade (clicks) to users commissioning him to install new flooring (conversions) to users recommending his services and becoming loyal customers (recurring business).

Figure 3 depicts this advertising utility funnel. Note that the difficulty for the advertiser is that he needs to specify his benefit not in terms of 5 events (impression, click, conversion, revenue, profit) but in terms of a single event (typically either impression, click or conversion).

2.3 Pricing

It is well known [4] that the generalized second price auction is not entirely incentive compatible, i.e., the optimal strategy of an advertiser is not quite exactly to bid using his true value. However, the deviations from this effect are commonly assumed to be negligible and in the remainder of the paper we assume that the mechanism is at least monotonically incentive compatible, i.e., that the bids scale monotonically with the value for the advertiser. A simplified model of the value of an ad is given by

$$\begin{aligned} v(q,a) = v_{\text{disp}}(q,a) &+ p(\text{click}|q,a) \cdot v_{\text{click}}(q,a) \\ &+ p(\text{conv}|q,a) \cdot v_{\text{conv}}(q,a) \end{aligned}$$

where $v_{\text{disp}}(q,a)$, $v_{\text{click}}(q,a)$, and $v_{\text{conv}}(q,a)$ are the advertiser values for display, click, and conversion, respectively, and where q denotes the query and a denotes the associated ad. This means that an advertiser would be able to compute the value $v(q',a)$ for a new query q' simply by evaluating the corresponding click and conversion probabilities, $p(\text{click}|q',a)$ and $p(\text{conv}|q',a)$. Unfortunately, the search engine only receives

$$b(q,a) = \frac{v(q,a)}{p(\text{click}|q,a)} \tag{1}$$

which specifies the bid, i.e., how much an advertiser is willing to pay per click, such that the expected cost per view matches the desired value $v(q,a)$. However, this is insufficient to infer the values for the respective stages of the conversion funnel, hence we formulate our approach in terms of the probability of conversion rather than clicks.

We make a quite crude assumption that advertisers only care about conversions, i.e., $v_{\text{disp}}(q,a) = v_{\text{click}}(q,a) = 0$, and furthermore the value of a conversion is independent of the query, i.e., $v_{\text{conv}}(q,a) = v_{\text{conv}}(a)$ (the latter assumption is also used in Section 3.2). Then we obtain the following relationship:

$$b(q',a) = b(q,a)\frac{p(conv|q,a)}{p(conv|q',a)} \tag{2}$$

This relationship will be the basis for one of our approaches to compute the missing bids for advanced matches: the conversion data model. The advantage of the conversion data model is that Eq. (2) does *not* depend on explicit knowledge of $v_{\text{conv}}(q,a)$ any more, but rather just on externally observable quantities such as the ratio between the conversion probabilities for different queries. One of the key disadvantages of assumptions in Eq. (2) is that it does not reflect the value of an ad very well, owing to a large number of (practically necessary) simplifications. We evaluate this model in detail in Section 7.

3. ADVANCED MATCH TARGETS

3.1 Bid based targets

A second source of data that can be used to predict the bids are the existing bids of the same and other advertisers.

In other words, we can try to estimating $b(q, a|Q)$, that is the amount an advertiser should be bidding for a click on ad a when query q was issued, where Q represents the information contained in the existing bids in the system. This would include all pairs of (q', b') bid phrases and bids and the information about how they fit in the ad schema described above. In the following we will simplify the formulas and omit the dependency on Q. A number of issues make it quite difficult to tackle the problem directly.

One of the key issue is lack or representative bids for the advanced match queries. When ad advanced match occurs, not only that there is no bid from the chosen advertiser, but in many cases advanced match triggers on tail queries without any bids at all. A proxy is to use the existing bids from the same ad group. That is, we may pretend having received bids for all queries but one and try to estimate their value based on the remaining bids. In the example of the flooring contractor assume that he issues bids on queries for 'acorn', 'beech' and 'cedar' to show a given ad a. In this case we could try to estimate how much he would have bid for 'cedar' given that we know his bids on 'acorn' and 'beech' and that we have information of how similar 'cedar' is to the previous two queries. This is obviously contingent on the preprocessing stage identifying 'acorn' and 'beech' as similar to 'cedar'. There are several potential issues with this approach:

Subsampling bias: A significant problem arises from the fact that by leaving out only one ad at a time we might be able to glean an unrealistic amount of information from the advertiser's bidding behavior on related queries. We can alleviate the issue by evaluating the performance on queries, ad groups, campaigns or accounts which were not used in building the model. While this approach does *not solve* the problem (by definition advanced match data is not available), it allows us to assess the bias rather conservatively.

Advanced match bias: A considerable number of advertisers which provide exact match bids do so by providing bids for a *combination* of exact and advanced matches. In other words, our flooring contractor may bid $1 for queries of 'acorn' with the implicit expectation that an adjusted version of this bid is to be used for an advanced match. This bid is unbiased *provided* that the advanced matching estimates are perfect. In case the advanced matches are 'underpriced' this will drive up the bids for the associated keywords and vice versa. If the sets of advanced matches were disjoint this would result in an overall fair distribution of bids (and it might arguably be the reason for the model discussed in [3]). However, since in reality these sets are partially overlapping no such efficiency guarantees are available. The degree of this bias can be assessed (partially) by evaluating the advanced match bids for advertisers which issue exact match bids (but in large numbers) only.

Relevant instance bias: Advertisers provide us only with bids for keywords they are actually interested in. For instance, the flooring contractor is unlikely to bid $b = 0$ for queries such as "bottle opener", since users issuing such queries are unlikely to be interested in his services. This problem is addressed by pre-filtering algorithms such that the set of queries for which we will attempt to estimate an advanced match bid is restricted to relevant ones (see Figure 1).

A second issue is that advertisers are equally unlikely to bid on items that are consistently too expensive for them. For instance, while the query "home loan" might very well be commercially valuable for the flooring contractor, he is likely to be outbid by banks and credit unions, hence he may not bid on "home loan" at all or his bid may not be very accurate. The later is less of an issue since it is likely to occur only for bids that are considerably *higher* than usual (which are unlikely to occur with a regression estimator), hence even if our bid estimator is somewhat inaccurate, it would not have a significant effect since the only outcome is that the advertiser would lose an auction that he would not have won anyway.

Nonetheless we will use this data in one set of experiments to assess the accuracy.

3.2 Conversion based targets

As noted earlier, an alternative to the use of leave-one-out estimates is to employ Eq. (2) despite the rather crude approximation used in obtaining it and to assume rationality of advertisers. In other words, whenever advertisers provide conversion data for their ads we can use this information to adjust bids accordingly. Unfortunately such data is sparse. That is, while we might have a sizable number of conversion events per advertiser, it is quite common to have many keywords for which only a single conversion has been recorded, leading to unreliable estimates of the associated conversion probability.

We use a simple technique from natural language processing — backoff smoothing of counts [14], to address the problem. The basic idea is that aggregate conversion probabilities at a given level will be a good prior for conversion probabilities at the next lowest level (e.g. a good prior of conversion probabilities for bid phrases are conversion probabilities for the associated ad group).

We use hierarchical Laplace smoothing to address this issue: it is reasonable to assume that the general conversion probability $p(\text{conv}|\text{click})$ is a good prior for the conversion probability per advertiser $p(\text{conv}|\text{click}, A)$. Moreover, it is reasonable to assume that the latter is a good prior when conditioning on advertiser and campaign (A, C), and so on for ad groups S and queries Q, as described in the hierarchy of Figure 2. This yields the following estimates:

$$\hat{p} = \frac{n_{\text{conv}}}{n_{\text{click}}}$$

$$\hat{p}(A) = \frac{n_{\text{conv}}(A) + n_0\hat{p}}{n_{\text{click}}(A) + n_0}$$

$$\hat{p}(A, C) = \frac{n_{\text{conv}}(A, C) + n_0\hat{p}(A)}{n_{\text{click}}(A, C) + n_0}$$

$$\hat{p}(A, C, S) = \frac{n_{\text{conv}}(A, C, S) + n_0\hat{p}(A, C)}{n_{\text{click}}(A, C, S) + n_0}$$

$$\hat{p}(A, C, S, Q) = \frac{n_{\text{conv}}(A, C, S, Q) + n_0\hat{p}(A, C, S)}{n_{\text{click}}(A, C, S, Q) + n_0}$$

In practice we choose $n_0 = 10$. The rationale is that unless we have approximately 10 instances per parameter, estimates of probabilities are going to be rather noisy. Note

that as the sample sizes for n_{conv} and n_{click} increase, this estimator will converge to the true probability estimate due to consistency of conjugate priors. Just as the bid based targets the conversion based targets have a number of drawbacks:

Sample selection bias: Not all advertisers are equally sophisticated when it comes to monitoring their campaigns. Nor do all advertisers have the same notion of conversion. While the latter is less relevant since (2) only uses ratios of counts, the former matters considerably since only a biased subset of advertisers actually uses conversion tracking. Hence, using their data to build a general advanced match bid generator is fraught with difficulty.

Model bias: An equally big issue is the fact that the approximations leading to (2) are rather limited in the first place. That is, they entirely ignore variations of the value of a conversion dependent on a query, any value of a click or any value of an ad display.

As our experiments show, conversion data is often only of limited use while bid data is much more reliable, both due to the economic incentive to bid accurately and due to the larger amount of data.

4. METHODOLOGY

Having established the means for computing regression targets for bids we want to estimate b. Using information retrieval as a filter ensures that the distribution of possible (q, a) pairs is not too dissimilar from the actual set of bids, we estimate the random variable $b|a, q$ using either the advertisers' existing bids or his conversion probabilities to generate training data. Both stages are necessary: the first stage limits the set of potential ads whereas the second one fine-tunes the bids such that they most closely match what an advertiser *would* have offered had he chosen to display an ad for a query.

4.1 Loss

Assuming we have the true bid b for a given (q, a) tuple we need to determine by how much a deviation between the true bid b and the estimate $\hat{b}$ should be penalized. Overall, we posit that the class of functions

$$L(b, \hat{b}) := l(\psi(b) - \psi(\hat{b})) \tag{3}$$

is suitable to measure the discrepancy between the "true" bid and its estimate. Here $\psi : \mathbb{R} \to \mathbb{R}$ is a *strictly increasing* function and $l : \mathbb{R} \to \mathbb{R}$ is a convex non-negative function which satisfies without loss of generality that $l(0) = 0$.

Picking the identity $\psi(x) = x$ is not necessarily in the advertiser's best interest: while this strives to minimize the average prediction error, it means that an error of \$0.05 for a bid of \$10.00 has equal value as that error for a bid of \$0.10. In other words, advertisers for cheap keywords are at a significant disadvantage in terms of estimation accuracy. This is undesirable since advertisers care about performance *relative* to their expense rather than in absolute terms. Choosing $\psi(x) = \log x$ addresses the issue.

Secondly, we choose squared loss $l(x) = \frac{1}{2}x^2$ to penalize deviations on the log-scale. Log-normality of errors is a common assumption in financial mathematics (e.g. the Black-Scholes model of option pricing). Note that a large number of alternatives are possible, for instance Huber's robust loss [9] which limits the influence of outliers. We use

$$L(b, \hat{b}) = \tfrac{1}{2}\big(\log b(q, a) - \log \hat{b}(q, a)\big)^2 \tag{4}$$

to compute $\beta := \log \hat{b}$ directly, yielding bids via $\hat{b} = e^{\beta}$.

4.2 Risk

Doing well on a single bid per se is not very meaningful. Instead, we want to have a measure of performance which quantifies progress on the entire range of combinations (a, s, q). We define the expected risk via

$$R := \sum_{q,a} L(b(q, a), \hat{b}(q, a))w(q, a) \tag{5}$$

Here $w(q, a)$ is a weighting function which ensures that we emphasize goodness of fit in relevant regions. Moreover, we will need to fashion a corresponding empirical risk term

$$\hat{R} := \sum_{(q,a)\in Z} L(b(q, a), \hat{b}(q, a))\hat{w}(q, a) \tag{6}$$

which tries to approximate R as well as possible. Here Z contains all *available* data and $\hat{w}(q, a)$ denotes a weighting term associated with the available data.

4.3 Generalized Linear Model

We use a generalized linear model to capture the dependency between queries, bid phrases, and advertisers. To train this model we use the existing bids in the system (or alternatively the conversion based targets described in Section 3.2). For each bid phase, bid pair (q, b), we assume that there is no bid specified (take the pair out). And then train the model to guess the bid for this bid phrase. This way we can train the model to guess the ground truth of existing bids specified by the advertiser. The rationale is that the estimator should be capable of recovering the advertisers' true bids for exact match data. After all, this is the only data where we have proper information about what the advertiser actually intended to bid, capturing all potential value of the bid for the advertiser. More formally, we extract features $\phi(q, a, Q')$ in order to obtain

$$\log b(q, a) = \langle \phi(q, a, Q'), w \rangle \tag{7}$$

for a suitably chosen parameter vector w. Note that Q' here represents all the other bids in the system, excluding the one that has been considered. In summary, we have the following minimization problem:

$$\hat{w} = \operatorname*{argmin}_{w} \sum_{(q,a)} w(q, a)\frac{1}{2}\left[\log b - \langle \phi(q, a, q', b'), w \rangle\right]^2$$

Finding a near-optimal solution of the above optimization problem is straightforward via stochastic gradient descent. For numerical and statistical stability we add a small quadratic penalty $\frac{\lambda}{2}\|w\|^2$ to the objective. This proved essential to ensure good convergence. We have the following optimization algorithm:

Initialize $w = 0$ and $n = n_0$
repeat
  Get new (q, a)
  Increment counter $n \leftarrow n + 1$
  Set learning rate $\eta = c/\sqrt{n}$
  Compute error $\delta = \langle \phi(q, a, Q'), w \rangle - \log b$

Update $w \leftarrow (1 - \eta\lambda)w - \eta \cdot \delta \cdot \phi(q, a, Q')$
until no more data

It can be shown [12] that this algorithm converges at rate $O(T^{-\frac{1}{2}})$ to the risk minimizer. A very small number of passes through the data (in the order of 10) suffices.

4.4 Missing Variables

Missing variables are pervasive in sponsored search: for instance, some queries are sufficiently rare that not all of their features are available, systems might fail to record and process data, and certain features may not be well-defined (e.g., the bid variance for advertisers with only one bid).

In the following we denote by $x = (x_o, x_u)$ a random variable where x_o represents the observed part and where x_u corresponds to the unobserved (hence missing) part of an observation. It is tempting to approach the regression problem of computing $\langle w, x\rangle$ by estimating the unobserved random variables $x_u|x_o$ first and to simply plug the conditional estimate into the linear function $\langle w, x\rangle$. This approach is not desirable since it ignores a number of aspects:

1. There may be significant estimation error associated with trying to find the missing variables *conditioned* on the fact that they are missing.

2. The variables may *not* be missing completely at random: the fact that we have partial information might be indicative of a particular type of data (e.g., the case of missing variance for advertisers with only one bid).

3. At runtime, the estimation process is slowed down since we first need to estimate the value of the missing variables and only then compute $\langle w, x\rangle$.

These problems can all be addressed by defining the following feature representation: instead of x we use

$$x_i \rightarrow (x_i, 0) \text{ if } x_i \text{ is observed; } x_i \rightarrow (0, 1) \text{ if } x_i \text{ is missing}$$

(the second element serves as an indicator variable). Consequently, instead of estimating the expected value of x_i directly, we estimate the value of the product $x_i \cdot w_i$ as a single term. We never need to compute the value of the missing variables at all, and moreover we simply perform the linear-optimal correction provided that x_i is missing. The only drawback of this approach is that it does not take the actual value of the remaining observed features into account (this could easily be addressed by higher order features).

5. SAMPLE WEIGHTING

Our methodology predicts bid values based on existing bids of the same advertiser as well as bidding behavior of other advertisers. When taking into account others' bids, we should obviously only consider bids of live ads that are being displayed and disregard those of dormant or discontinued campaigns. But should a bid of an ad showing once a month be trusted to the same extent as the one showing thousands times a day? At the very least, frequently displayed ads are likely to be much better tuned, and hence their bids are likely to be more realistic in the given market. We capture this intuition by weighting bids by the amount of money spent by the advertiser.

$$w(q, a) \propto \{\text{Spend on } q \text{ by advertiser}\} \quad (8)$$

Scale Neutrality: An immediate consequence is that estimates which have the same relative error in terms of bid estimation will have the same amount of overall error contribution *regardless* of the level of the actual bid. More concretely, an advertiser spending \$100 on bids of a price of \$1 each and an advertiser spending the same amount on bids of a price of \$10 each, both of which attract a relative error of, say, 5%, will generate the same error contribution.

Robustness: A desirable side-effect of weighting by budget is that the bid estimator becomes robust against manipulation by advertisers: if an advertiser were to increase his bid in the hope to increase the bid estimate for advanced match of a competitor, his data would only weighted by his actual spend on the keyword. Consequently significant manipulation would require resources proportional to the degree of manipulation. Likewise, if the advertiser were to try and lower his bid, the advertiser will fail to win auctions and as a result his spend on the keyword will decrease, thus decreasing his statistical weight. This prevents an oscillating strategy where an advertiser alternates between high and low bids to benefit from pricing his competitors out of the market due to incorrect advanced match bids.

Effective Sample Size: While weighting data may reduce bias (if the weighted loss is what we strive to minimize), it may significantly increase variance and thereby lead to inferior estimates. One means of quantifying this effect is to compute the effective sample size m_{eff}. For a dataset with weights w_i where $\sum_i w_i = 1$ one may show [7] that $m_{\text{eff}} = \|w\|_2^{-2}$. For instance, if we have 100 observations out of which 10 have 90% of the weight with the remainder evenly distributed among the rest, we have $m_{\text{eff}} = 12.3$. This is intuitively clear — changing any of the over-weighted observations affects the estimate almost as if the remaining 90 observations did not exist.

This leads to two opposing effects for sample weighting: while minimizing the weighted dataset eliminates potential bias it also induces higher variance due to the reduced data set size. Hence, if the unweighted data is not too biased, it may be advantageous to refrain from reweighting since variance increase dominates the bias reduction. Empirical evidence (Section 7) suggests this for advertisers' bids.

6. FEATURES

Bid generation is a complex problem as it essentially seeks to match human reasoning and sales information about the business value of the bid phrases. We believe that merely using the bids of other phrases is insufficient. Since we are predicting the bid of a given ad for a given query, we formulated three kinds of features, namely, those characterizing the query, the ad (and the advertiser), and their interaction (i.e., the query-ad pair). In what follows, we experimentally validate the utility of these features.

All textual features are computed using stop word removal and stemming. For phrase extraction we used a variant of AltaVista's Prisma tool [1]. Whenever dealing with text, we use a TFIDF representation of the text as a bag of words vector.

6.1 Query Features

The idea behind query-side features is that similar queries should get similar bids. For instance, bids for the query 'red roses' should tell us more about suitable bids for 'white roses'

rather than for 'car insurance'. Given a query q, we define the following features:

1. A TFIDF vector representing the query as a bag of words and phrases (this leads to a potentially unlimited number of features);

2. The number of words and the number of phrases in q (the length of a query is indicative of its prevalence).

3. Following [2], we expand the query with Web search results, and take the most salient $N_w = 50$ unigrams and $N_{\text{ph}} = 50$ phrases from these results as additional features of the query.

4. Query frequency in search logs of the previous month.

5. The minimum and maximum document frequency (DF) of query words and phrases in the Web corpus

6. The number of advertiser accounts bidding on the query (this quantifies how competitive each bid phrase is).

7. The average, minimum, and maximum bid on the query (if any) across all advertiser accounts.

6.2 Ad Features

1. Simple statistics of the ad group, as well as of its enclosing campaign and account: the number of bid phrases and creatives; the average, minimum and maximum bid; the average, minimum and maximum frequency of bid phrases as queries in Web search.

2. The centroid of all the bid phrase vectors in the ad group.

3. The centroid of the expansion vectors for the bid phrases (using Web search results), similar to the expansion of queries (cf. item 3 in Section 6.1).

4. The centroid of the text of all creatives in the ad group.

5. The topical cohesiveness of the ad group, as well as of its campaign and account, computed as an average distance of bid phrases and creatives from the corresponding centroids (see items 2–4 above).

6.3 Query-Ad Features

The obvious combination of per-query and per-ad features by taking outer products may become computationally prohibitive. Therefore, we explicitly define features of the query-ad pairs and compute cosine similarity between the query vector (item 3 in Section 6.1) and the centroids pertaining to the ad features (items 2–4 in Section 6.2)

As explained in Section 4.3, we employ the "leave one out" approach both for training and for evaluating our methodology. That is, we use it for predicting existing bids of actual ads in our corpus. For a fair experiment, we obviously exclude the bid phrase and its bid value from any feature computation used for predicting that bid value.

7. EXPERIMENTAL EVALUATION

Our proposed strategy for dealing with advanced match is to use penalized weighted least mean square regression on the log-bids as a means for predicting advanced match bids, namely by using the bids obtained in section 3 via a synthetic leave-out estimate or a conversion based estimate. Weighting of individual instances is achieved (whenever needed) via budget-dependent rescaling as discussed in section 5. Finally, we deal with missing variables via the variable duplication trick of section 4.4.

7.1 Data description

We evaluated our methodology on a fraction of Yahoo's ad database obtained as a snapshot for a given day in June 2009. The raw data included 70k advertiser accounts, 200k campaigns and 2m ad groups.

We adopted the "leave one out" approach described in Section 4.3 which allowed us to test the ability of our system to predict actual bids that the advertisers explicitly specified for existing ads. This way, advertiser-specified bids served as the "gold standard" — the rationale was that after all advertisers should know best what they would like to bid for an keyword. We excluded all ad campaigns that had constant or near-constant bids, including single bid phrase campaigns.

In fact, one might argue that these advertisers would do better if they adjusted their bids with a larger degree of variation (which may not be possible for them due to lack of sufficient data, though). Constant and near constant campaigns provide no information to discriminate between the possible bid values, and our method will be effectively forced to predict that constant value. One possible reason for such indiscriminate bidding might be advertiser's lack of knowledge of the true value of the various bid phrases. Our definition of near-constant bids was a logarithmic bid variance of less than 0.1. This is significantly less than the error of the bid predictor that we computed, hence including such advertisers would only improve the error rate. The resulting data set is only 20% smaller. We created three different test sets to simulate the following scenarios:

Cold start (ACCT): When an advertiser establishes a new account, the system should be able to generate bid values for the account immediately. To evaluate the ability of our system to support this scenario, we formed the first test set by randomly selecting 10% of advertiser accounts. In each account, we used the leave one out approach to predict each bid given all the other ones, but *none of these accounts' data was included in the training set.* This is the most difficult scenario.

New campaign (CAMP): The second test set was similarly designed to evaluate our system's ability to predict bids for newly defined ad campaigns. We randomly selected 10% of all campaigns and used all their bids as test set (CAMP). Other campaigns belonging to the same account could be included in training.

New bid phrase (PHRASE): Finally, we emulate the direct case of advanced match bid generation by pretending that the advertiser did not add a particular bid phrase and by checking whether our estimates in this case accurately reflect what the advertiser would have bid. This scenario is closest to the real bid generation

problem since we only need to perform estimation for new phrases on otherwise well known advertiser data. In analogy to before we defined the third test set by randomly selecting 10% of individual bid phrases.

To summarize, we applied the {90%,10%} split at different levels of the ad hierarchy (see Section 2 and Figure 2) to test the prediction abilities of our system at different resolutions.

7.2 Sample weighting

To evaluate the soundness of the budget calibration of Section 5 we use data concerning funds spent in the previous week to weigh the accuracy of bids. As with click data, we used backoff smoothing (see Section 3.2) to smooth between spent budget at different levels. This addresses issues such as spurious reweighting within rare keywords.

Moreover, to address questions regarding the validity of the weighting approach we compare the performance of estimates obtained by uniform weighting (ignoring money spent per bid phrase) and by our proposed weighting scheme. This leads to the following experiments:

1. Both training and test examples are weighted uniformly.
2. Only test examples are weighted.
3. Only training examples are weighted.
4. Both training and test examples are weighted according to actual spend.

Table 3 provides the experimental results for the case of subsampling on a per-phrase basis (PHRASE). As evaluation metric we use the least mean squares error defined in Section 4.1. That is we penalize by the squared deviation between the logarithm of the bid and the estimate using (4).

7.3 Baseline

Our methodology uses a multitude of features to predict the bid value for a given bid phrase. In order to test whether this complexity is warranted, we compared the estimator to a simple baseline algorithm that uses only the average of the remaining bid values for other phrases in the same ad group in order to predict a bid value for a new phrase.

To justify our choice of the baseline, let us first revisit the ad retrieval method, which selects candidate ads to be shown on the page. Given a query q', it retrieves a number of relevant ads, each of which is composed of a creative s and a bid phrase q (we assume that $q \neq q'$, that is we assume that q was not explicitly bid on by the advertiser, and hence this bid needs to be predicted at runtime).

While the implementation details of the retrieval module are outside of the scope of this paper, it identifies relevant creatives and pairs them with the most relevant bid phrase (as described in Section 2). Note that each creative s may be paired with multiple bid phrases q. We average the bid values b of the ad group containing s and q, and we use this average value as our baseline. Since there may be significant variance within bids of an ad group, averaging the values in an ad group is more appropriate than taking any single one.

7.4 Bid generation for exact match data

We purposely decided to use exact match data to learn the bids since advertisers that sign for advanced match might already discount their bids. In fact, exact match bids are the clearest signal of the true value of the bid to the advertiser. By its very definition, there is no data for advanced match since the latter is defined by the absence of explicit bids.

	ACCT	CAMP	PHRASE
Sample size	31,089,439	29,492,142	30,031,135
Effective size	83,654	83,737	69,944

Table 1: Real and effective training sample sizes for bid data with at least 0.8 log-variance. Observe that the effective sample size is 2.5 orders of magnitude smaller, leading to a significant increase of variance in the estimator.

Data split	Uniform	Weighted
ACCT	15.1	4.3
CAMP	13.2	4.6
PHRASE	20.3	10.5

Table 2: Improvement (error reduction in percent) of the estimates relative to the baseline performance for a variance threshold on log-bids of 0.8 for both unweighted and weighted data.

Data split	Variance	Uniform	Weighted
PHRASE	0.05	10.8	4.1
PHRASE	0.10	4.2	6.6
PHRASE	0.20	13.6	5.0
PHRASE	0.40	20.6	6.7
PHRASE	0.80	20.3	10.5

Table 3: Improvement relative to the baseline performance at different variance thresholds for both unweighted and weighted data.

To test the accuracy of our approach we selected data with a sufficiently high level of variance relative to a constant bid. For the default set of experiments we chose cases where the variance in the log-bid exceeded a threshold of 0.8. This yielded a dataset of approximately 30m training samples (and 3m test samples) with an effective sample size of approximately 80m for weighted data (see Table 1).

Given the large sample sizes we report results on a 10% test set rather than a full 10-fold cross validation. This is statistically safe — Chernoff bounds for a test set of $3 \cdot 10^6$ suggest a relative confidence interval of 0.1%. This is much smaller than the gains we are reporting, hence our results are statistically highly significant.

Our experiments are carried out on commercially sensitive data. Hence we are unable to report absolute performance figures. Instead, we report improvements relative to the baseline performance in Table 2. Not surprisingly, our estimator performs best when applied to bid estimation at phrase level and worst when applied at the account level. This is the case since there we are able to use much more side information about keywords for a particular campaign and advertiser when leaving out PHRASE data rather than leaving out an entire account in the ACCT dataset. This is nonetheless encouraging since the PHRASE scenario is much closer to reality (we can always update our estimates once we see new campaigns of an advertiser).

Note that the improvement relative to the baseline is considerably worse when using weighted data (see Table 2 and 3). A large part of this is likely due to the dramatic reduction in effective sample size by 2.5 orders of magnitude (see Table 1), thereby increasing the variance more than what

can be counteracted by a reduction in bias. This hypothesis is confirmed in Table 4 which shows that while training on unweighted data improves performance on the weighted dataset, the converse is not true. Indeed, the estimate is significantly worse than the baseline (this finding also holds for other variance thresholds than 0.8 reported in the table).

7.5 Using conversion data

For a fraction of advertisers (in the range of 20%–30%), we have access to conversion data, which reflects the fraction of users who actually purchase the product or service being advertised *after* clicking on the ad. Intuitively, this information is highly valuable for bid generation, since knowing how different bid phrases "convert" can lead to a better estimation of their true value to the advertiser. Conversion has always been assumed to be the key factor in determining the value that the advertiser gets from the ads. In fact, today some of the major sponsored search providers offer experimental pricing models based on cost per conversion (or per action) as opposed to the traditional cost-per-click model. In our experiments we examined the conversion data (from a fraction of the advertisers) and compared its utility in predicting the bids with that of the existing bid values.

We conducted 3 experiments, coupling training and testing on the conversion data with training and testing data on the advertiser's bids. If conversions are the major factor in advertisers' determination of the bids, one would expect that models trained on one of these data sources would be able to predict the test values obtained from the other. Table 5 shows the results of these experiments (relative to the baseline of training and testing on bid data). The experiments indicate that while there is good signal in both bids and conversion data, both datasets are largely incommensurate, as evident in the large prediction error when applying an estimate from the bids to conversion data.

This discrepancy is supporting evidence that bids involve many more factors rather than just plain conversions (see also Section 8 for a more detailed discussion). It is subject of future research to find a joint latent space which is capable of predicting both effects simultaneously.

7.6 Feature selection

Our method uses multiple features of different types. We performed a series of ablation studies to assess the informativeness of different features. Owing to the multitude of features used by our model, each time we eliminated an entire group of similar features rather than individual ones and for comparison purposes we used only those features while excluding all others. Of particular interest are bid-related features. We investigate the effect of ad group, (ad group, campaign), (ad group, campaign, account), and the entire subset of bid features. Table 6 contains the changes relative to the baseline of a full feature set.

We see that removal of the ad-related features described in Section 6.2 decreases the estimates considerably with the not very surprising effect that performance keeps on decreasing as we remove more features. What is relevant, though, is the extent to which performance is decreased when removing the feature group and the fact that it, on its own, is equally insufficient for bid generation. In other words, Table 6 establishes that it is the interaction between different feature sets that leads to good results. Qualitatively similar results can be obtained by removing query-related features.

	train unweighted	train weighted
test unweighted	20.3	-19.2
test weighted	3.7	10.5

Table 4: Performance relative to baseline in percent when training / testing on weighted and unweighted data. The experiments were carried out at a variance cutoff of 0.8.

	train bids	train conversions
test bids	1.0	n.a.
test conversions	26.6	1.49

Table 5: Performance relative to training and testing on unweighted bid data. As can be seen there is relatively reliable signal in both the advertisers' bids and the users' conversion behavior. However, both datasets differ significantly, as can be seen in the very large relative prediction error of 26.6 when training on bid data and testing on conversion data.

	removed	exclusively
ad group	-14.0	-17.0
ad group & campaign	-14.3	-16.1
ad group & campaign & account	-15.0	-16.0
all bid features	-16.2	-14.2

Table 6: Relative effect of ad features on estimation performance. The results (in percent) are in terms of error increases over the full set of features. We used a subset of the data with a minimum log-variance threshold of 0.4.

7.7 Related Work

Recently, several approaches have been proposed to find related queries and bid phrases for use in advanced match and bid phrase suggestion. [11] report a framework for generation and evaluation of query rewrites for sponsored search using query log and user session data. As in our approach, query similarity is based on lexical and semantic features of the queries in the a session (all queries issued by the same user within a certain time period). While we don't use the same set of features as in this work, features can be easily added to our framework. The trade-off between the revenue and relevance in query rewrite generation for sponsored search is explored in [13]. Query rewrites here are evaluated using lexical features of the query and the rewrite. The ad schema can be used as a source for query rewrite generation as well. [10] report an approach where the strength of the relationship between the bid phrases is determined by a random walk over the bipartite graph of bid phrases and advertisers. A similar approach has been proposed for bid phrase suggestion uses the bipartite graph of queries and web search URLs [5]. None of these approaches addresses the issue of determining the bid for the rewrites.

Dar et al. [3] study the problem of advanced match from the advertiser's perspective. Specifically, they study the amount an advertiser should bid for a keyword provided that the search engine exercises a *uniform cost* for all clicks associated with the given keyword.

This is clearly an extremely adversarial and rather unde-

sirable scenario in which the advertiser needs to determine how much to bid for the mixed basket and how several of these baskets might interact. While the setting leads to mathematical insights concerning max-flow/min-cut problems, we believe that it is somewhat less applicable to a real world situation where advertisers would expect a discount for less relevant queries in the context of advanced match. It is the latter that we study in this paper.

Our work is orthogonal to the results in [3] insofar as the advertiser is free to use the max-flow reasoning to fine-tune his pricing strategy once the search engine has carried out its attempt to price ads optimally for the advertiser. We are not aware of other related work addressing the problem of advanced match bid estimation beyond [3].

8. SUMMARY AND DISCUSSION

We reported the first study of bid generation for advanced match in sponsored search. Advanced match is used to select a significant portion of the ads shown today, and is responsible for billions of dollars of sponsored search revenue. In this paper, we explored the advertisers' bidding behavior using two data sources: the actual bids and conversion data.

The analysis of the bid data shows that many advertisers assign the same or very similar value to all (or most) of the bid phrases in their ad campaigns. This does not mean, of course, that advertisers ultimately *derive* the same value from different bid phrases. Instead, we speculate that the near-uniform bidding is likely due to lack of data and inability to make more informed decisions. For such advertisers, the only information available for generating bids for a given query is the relevance score of the ad retrieval module (usually, an IR-based model). If a given ad is scored for the query as perfectly relevant, then the bid would be equal to the advertiser's (uniform) bid in the given ad group. Otherwise, the bid can be adjusted based on the confidence in the advanced match.

Our results on the ad campaigns with higher bid variance suggest that these advertisers bid rationally, based on market data features such us bid landscape, search frequency, query similarity to the ad, etc.

The disparity in quality of bid generation using conversion data and using actual bid data may have several causes:

- Some advertisers may lack appropriate expertise and are unable to use the conversion data for the bid calibration. However, in the presence of companies offering bid optimization services, we find this hypothesis the least likely.
- An alternate explanation is that conversion data is not representative of the actual profit that the advertiser obtains. A supporting argument for this case is that the value varies significantly based on keywords.
- Moreover, this disparity can be indicate that the value the advertisers obtain from sponsored search goes beyond the immediate measurable conversions, and include user engagements caused by the other events in the funnel (ad views and ad clicks), as well as potential offline transactions.
- Finally, we observed significant covariate shift among the advertisers opting to use conversion tracking. This may indicate that the data obtained from conversions is simply not representative of *all* advertisers.

Overall it seems that the bid data is the strongest source of signal. We think this is a very important finding, which suggests that the advertiser bid is derived from more than just the conversion rate (e.g., advertisers derive value from ad display).

In our future work, we plan to conduct additional experiments to obtain more insight into these issues. Specifically, we plan to study the correlation between the amounts advertisers spend on their campaigns and the variance of their bids. We also plan to experiment with alternative spent-based weighting schemes (e.g., logarithmic weighting).

9. REFERENCES

[1] P. Anick. Using terminological feedback for web search refinement: a log-based study. In *SIGIR'03*, pages 88–95, 2003.

[2] A. Broder, P. Ciccolo, M. Fontoura, E. Gabrilovich, V. Josifovski, and L. Riedel. Search advertising using Web relevance feedback. In *CIKM'08*, 2008.

[3] E. Dar, V. Mirrokni, S. Muthukrishnan, Y. Mansour, and U. Nadav. Bid optimization for broad match ad auctions. In *WWW*, pages 231–240, 2009.

[4] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, 2007.

[5] A. Fuxman, P. Tsaparas, K. Achan, and R. Agrawal. Using the wisdom of the crowds for keyword generation. In *WWW*, pages 61–70, 2008.

[6] S. Goel, A. Broder, E. Gabrilovich, and B. Pang. Anatomy of the long tail: Ordinary people with extraordinary tastes. In *WSDM*, pages 201–210, 2010.

[7] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Dataset shift in machine learning. In J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, editors, *Covariate Shift and Local Learning by Distribution Matching*, pages 131–160, Cambridge, MA, 2008. MIT Press.

[8] D. Hillard, S. Schroedl, E. Manavoglu, H. Raghavan, and C. Leggetter. Improving ad relevance in sponsored search. In *WSDM'10*, pages 361–370, 2010.

[9] P. J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.

[10] A. I., M. H, and C. C. Simrank++: Query rewriting through link analysis of the click graph. In *PVLDB*, 2008.

[11] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *WWW*, 2006.

[12] Y. Nesterov and J.-P. Vial. Confidence level solutions for stochastic programming. Technical Report 2000/13, Université Catholique de Louvain - Center for Operations Research and Economics, 2000.

[13] F. Radlinski, A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, and L. Riedel. Optimizing relevance and revenue in ad search: A query substitution approach. In *Proceedings of SIGIR*, 2008.

[14] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.