Predicting Clicks: Estimating the Click-Through Rate for New Ads

Matthew Richardson
Microsoft Research
One Microsoft Way
Redmond, WA 98052
mattri@microsoft.com

Ewa Dominowska
Microsoft
One Microsoft Way
Redmond, WA 98052
ewad@microsoft.com

Robert Ragno
Microsoft Research
One Microsoft Way
Redmond, WA 98052
rragno@microsoft.com

ABSTRACT
Search engine advertising has become a significant element of the Web browsing experience. Choosing the right ads for the query and the order in which they are displayed greatly affects the probability that a user will see and click on each ad. This ranking has a strong impact on the revenue the search engine receives from the ads. Further, showing the user an ad that they prefer to click on improves user satisfaction. For these reasons, it is important to be able to accurately estimate the click-through rate of ads in the system. For ads that have been displayed repeatedly, this is empirically measurable, but for new ads, other means must be used. We show that we can use features of ads, terms, and advertisers to learn a model that accurately predicts the click-though rate for new ads. We also show that using our model improves the convergence and performance of an advertising system. As a result, our model increases both revenue and user satisfaction.

Categories and Subject Descriptors
I.2.6 [**Artificial Intelligence**]: Learning. H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval.

General Terms
Algorithms, Measurement, Performance, Economics, Experimentation.

Keywords: Click-through rate, sponsored search, paid search, Web advertising, CTR, CPC, ranking.

1. INTRODUCTION
Most major search engines today are funded through textual advertising placed next to their search results. The market for these search advertisements (sometimes referred to as "paid search") has exploded in the last decade to $5.75 billion, and is expected to double again by 2010 [17]. The most notable example is Google, which earned $1.63 billion in revenue for the third quarter of 2006 from search advertising alone [2] (a brief summary of the history of sponsored search can be found in [7]).

Though there are many forms of online advertising, in this paper we will restrict ourselves to the most common model: pay-per-performance with a cost-per-click (CPC) billing, which means the search engine is paid every time the ad is clicked by a user (other models include cost-per-impression, where advertisers are charged according to the number of times their ad was *shown*, and cost-per-action, where advertisers are charged only when the ad display leads to some desired action by the user, such as purchasing a product or signing up for a newsletter). Google, Yahoo, and Microsoft all primarily use this model.

To maximize revenue and user satisfaction, pay-per-performance systems must predict the expected user behavior for each displayed advertisement and must maximize the expectation that a user will act (click) on it. The search system can make expected user behavior predictions based on historical click-through performance of the ad. For example, if an ad has been displayed 100 times in the past, and has received 5 clicks, then the system could estimate its click-through rate (CTR) to be 0.05. This estimate, however, has very high variance, and may only reasonably be applied to ads that have been shown many times. This poses a particular problem when a new ad enters the system. A new ad has no historical information, so its expected click-through rate is completely unknown.

In this paper, we address the problem of estimating the probability that an ad will be clicked on, for newly created ads and advertising accounts. We show that we can use information about the ad itself (such as the length of the ad and the words it uses), the page the ad points to, and statistics of related ads, to build a model that reasonably predicts the future CTR of that ad.

2. MOTIVATION
The key task for a search engine advertising system is to determine what advertisements should be displayed, and in what order, for each query that the search engine receives. Typically, advertisers have already specified the circumstances under which their ads may be shown (e.g., only for certain queries, or when certain words appear in a query), so the search engine only needs to rank the reduced set of ads that are matches.

As with search results, the probability that a user clicks on an advertisement declines rapidly, as much as 90% [5], with display position (see Figure 1). Thus, it is most beneficial for the search engine to place best performing ads first. Note that, because the probability of clicking on an ad drops so significantly with ad position, the accuracy with which we estimate its CTR can have a significant effect on revenues.

The number of eligible advertisements matching a given query usually far exceeds the number of valuable slots. For example,

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

most users never go beyond the first page of search results, in which case the number of ads displayed is limited to the set shown on the first page (this number tends to range between 5 and 8 for the most common search engines). Even within the first page, the significant decrease in CTR by ad position means that ads in very low positions have less impact.

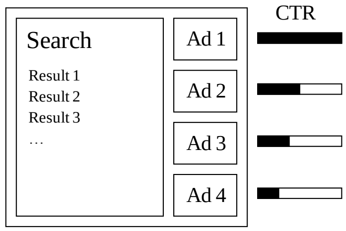

Figure 1. The CTR of an ad typically decreases with lower-positioned ads, due to reduced visual attention.

In order to maximize ad quality (as measured by user clicks) and total revenue, most search engines today order their ads primarily based on expected revenue:

$$E_{ad}[revenue] = p_{ad}(click) \cdot CPC_{ad}$$

(The most notable exception to this is Yahoo, which orders ads based on advertiser bid alone, but plans to switch to using expected revenue soon). The CPC for an ad is its bid (in a first price auction) or the bid of the next-highest bidder (in a second-price auction), optionally normalized by ad performance. The details of the relation between CPC and bid are not important to this paper, but are the study of many works on search engine auction models [8][12].

Thus, to ideally order a set of ads, it is important to be able to accurately estimate the *p(click)* (CTR) for a given ad. For ads that have been shown to users many times (ads that have many *impressions*), this estimate is simply the binomial MLE (maximum likelihood estimation), *#clicks / #impressions*. (In this paper, we assume that over time each ad converges to an underlying true click-through rate. We ignore ads that exhibit periodic or inconsistent behavior for the purposes of this paper, although the work could be extended to such cases.) However, because the CTR for advertisements is relatively low, the variance in this estimate is quite high, even for a moderate number of impressions. For example, an ad with a true CTR of 5% must be shown 1000 times before we are even 85% confident that our estimate is within 1% of the true CTR. In general search advertising, the average click-through rate for an ad is estimated to be as low as 2.6% [4].

The time over which the system converges reflects a large amount of search monetization. For example, an ad with a cost per click of $1.60 (an average rate on Google [4]) would require $80 of click-through behavior to experience 50 clicks. Any error in the click-through rate estimation during that time will result in suboptimal ranking and thus lost revenue for the search engine and lower traffic for the higher performing ads.

The search advertising market has grown significantly in recent years; there are many new advertisers that enter the market each day. Simultaneously, existing advertisers frequently launch new advertising campaigns. Many advertisers create new campaigns each month, some even every day; others create side-by-side orders for testing purposes in order to optimize their ad performance. All of these practices result in an increasing number of ads to be ranked for each query.

Additionally, existing ads are sometimes targeted to new queries. Some advertisers attempt to increase their return on investment by targeting thousands of infrequently searched terms. There has been a significant increase in keyword volume for PPC campaigns: In one study, the number of keywords per campaign per month increased from 9,100 in September 2004 to 14,700 by March of 2005, and was expected to grow to as many as 17,300 by September 2005 [4].

As a result, there is a large inventory of ads for which the search engine has no prior information. These ads need to be ranked with other, already established ads. An incorrect ranking has strong effects on user and advertiser satisfaction as well as on the revenue for the search engine. Thus, for ads that are new, or have not been shown enough times, we must find a way to estimate the CTR through means other than historical observations. This is the goal of the system described in this paper: to predict, for new ads and new advertisers, the probability that an ad will be clicked. (from here on, an *ad* will refer to a combination of a particular *ad presentation* from a particular *advertiser*, for a particular *bid term*).

Previous research by Regelson and Fain [19] estimates the CTR of new ads by using the CTRs of existing ads with the same bid terms or topic clusters. Our experience shows that even within the same term there can be a large variation in ad performance (in some cases, the CTR of the best ad can be ten times that of the average ad). To account for these within-keyword variations, it is important to incorporate features that depend on more than just the terms the ad was bid on; our model naturally incorporates such features, as we demonstrate in later sections.

The remainder of the paper is as follows. First, we discuss the search advertising framework. The next two sections describe our data and model. Sections 6-9 introduce term, ad, order, and external features to the model. In Section 10, we discuss the results and make observations about the model performance and properties. We conclude with a summary of contributions and future work.

3. SEARCH ADVERTISING FRAMEWORK

Whenever an ad is displayed on the search results page, it has some chance of being viewed by the user. The farther down the page an ad is displayed, the less likely it is to be viewed. As a simplification, we consider the probability that an ad is clicked on to be dependent on two factors: a) the probability that it is viewed, and b) the probability that it is clicked on, given that it is viewed:

$$p(click \mid ad, pos) = p(click \mid ad, pos, seen)\, p(seen \mid ad, pos)$$

(Note that we are assuming that the probability that it is clicked on but not viewed is zero). We also make the simplifying assumptions that the probability an ad is clicked is independent of its position, given that it was viewed, and that the probability an ad is viewed is independent of the ad, given the position, and independent of the other ads shown:

$$p(click \mid ad, pos) = p(click \mid ad, seen)\, p(seen \mid pos)$$

Let the CTR of an ad be defined as the probability it would be clicked if it was seen, or *p(click | ad, seen)*. From the CTR of an ad, and the discounting curve *p(seen | pos)*, we can then estimate the probability an ad would be clicked at any position. This is the value we want to estimate, since it provides a simple basis for comparison of competing ads.

For any ad that has been displayed a significant number of times, we can easily estimate its CTR. Whenever the ad was clicked, it was seen. Whenever the ad was not clicked, it may have been seen with some probability (Figure 2 shows a heat map of search page viewership intensity for different ad position). Thus, the number of *views* of an ad is the number of times it was clicked, plus the

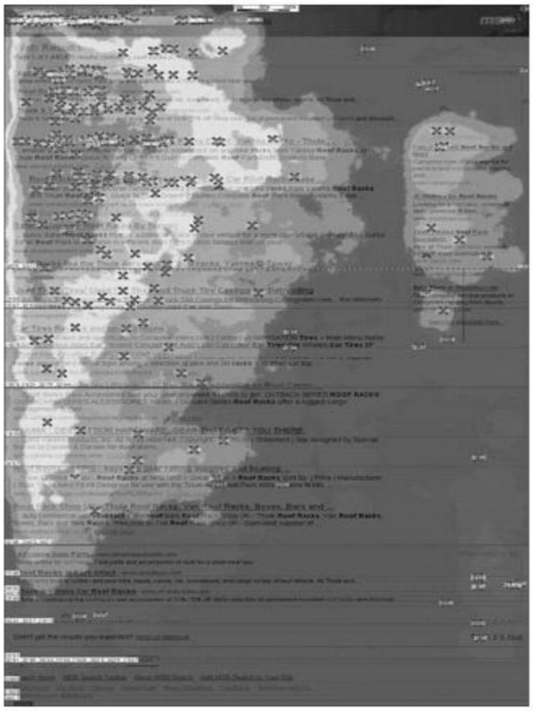

Figure 2. Eye scan activity on search results page [5]

number of times it was estimated to have been seen but not clicked. The relative probability of an ad being seen at different positions can be experimentally measured by presenting users with the same ad at various positions on the page. The CTR of the ad is simply the number of clicks divided by the total number of views.

Our goal is to create a model which can predict this CTR for new ads. In the next sections, we first present the data we used to train and test the model, followed by details on the model itself.

4. DATA SET

We collected information on a set of active ads in the Microsoft Web search engine. Each ad contains the following information:

- **Landing page**: The URL that a user is redirected to upon clicking the ad.
- **Bid term ("keywords")**: The query for which this ad should be displayed (this may be multiple words, e.g., "machine learning books").
- **Title**: The ad title, shown to the user.
- **Body**: The text description of the ad.
- **Display URL**: The URL shown to the user at the bottom of the ad.
- **Clicks**: The number of times the ad has been clicked since it was entered into the system.
- **Views**: The number of times the ad has been seen since it was entered into the system, as described in section 3.

The data set used is a significant sample of search ads. It includes 10,000 advertisers, with over 1 million ads for over a half million keywords (with over 100,000 unique ad texts).

Note that the same ad content may appear for different bid terms. In fact, the user interface for the ad system encourages this: Account holders create an "order", which is the ad information, and an associated collection of terms for which the ad should be displayed. We consider each pairing of the ad text with a term to be a unique ad, as the CTR for different terms varies significantly. Also, advertisers may specify whether an ad is displayed under the rules of *exact match*, or *broad match*. In the exact match case, the user query must exactly match the bid terms. In the broad match case, the bid terms can be related more loosely, such as being a subset of the query words. In this paper, we consider all clicks and views regardless of match type, and thus are attempting to predict the ad's CTR across all match types.

From the data, we have a number of ads, with the observed CTR for each. For each ad, our goal is to predict its CTR as if we did not know about it or any of the other ads entered by the same advertiser.[1] In order to prevent train-test contamination, we thus split our data on an advertiser-level (that is, all ads by the same advertiser went into the same split). We randomly placed 70% of the advertisers in the training set, 10% in the validation set, and 20% in the test set.

We also eliminated "premium" advertisers, which are advertisers with accounts that are professionally managed. This was done for two reasons. First, their advertisements often exhibit different trends from the general population (i.e., they have a different mean CTR and generally lower variance between ads), indicating they should be modeled separately. Second, in this paper we wanted to focus on advertisers for whom we have little or no experience or data, which constitute the majority of advertisers in self-serve systems, such as Microsoft's adCenter or Google's AdWords, where individuals can create accounts and post advertisements. Additionally, we limited the data to 1000 randomly selected ads per advertiser, to ensure significant variation.

The purpose is to estimate the *true* CTR of an ad, but all we actually know is the observed number of clicks and views an ad received, which leads to an *empirical* CTR. For ads with too few views, the empirical CTR may be wildly different from the true CTR, leading to much noise in the training and testing process. We thus filtered out any ads that had less than 100 views. (The choice to filter at a threshold of 100 is a balance between wanting less noise in the training and testing process – which argues for requiring more views, and reducing the bias that occurs when only considering ads that have been shown many times – which argues for requiring fewer views).

5. MODEL

Since our goal is to predict a real-value (the CTR of an ad), we cast it as a regression problem – that is, to predict the CTR given a set of features. We chose to use logistic regression, which is ideally suited for probabilities as it always predicts a value between 0 and 1:

$$CTR = \frac{1}{1+e^{-Z}} \qquad Z = \sum_i w_i f_i(ad)$$

where $f_i(ad)$ is the value of the i^{th} feature for the ad, and w_i is the learned weight for that feature. Features may be anything, such as the number of words in the title, the existence of a word, etc. (They will be described in more detail in the next sections.)

[1] This restriction is because our goal is to consider each ad and account as completely novel to the system. In future work, we would like to look at estimating the CTR for ads in already-established accounts

The logistic regression was trained using the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method [16]. We used a cross-entropy loss function, with zero-mean Gaussian weight priors with a standard-deviation of σ. The best σ was chosen on the validation set from the values [0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100]. In all experiments, σ=0.1 was the best. As is commonly done, we also added a bias feature that is always set to 1.

For each feature f_i, we added derived features of $log(f_i+1)$, and f_i^2 (the purpose of adding one before taking the log is so as to naturally handle features whose minimum value is 0, such as counts). We also normalized the features to have zero mean and unit standard deviation (the means and standard deviations were measured on the training set and then applied to both the training set and test set). Some features had significant outliers, so any feature value that was more than five standard deviations from the mean was truncated to five. These modifications to standard logistic regression were found to improve performance on the held-out validation set.

Our measure of performance is the average KL-divergence [13] between the model's predicted CTR and the true CTR on the test set (lower is better). The KL-divergence is simply the log-likelihood of the model, minus the entropy of the test set. (This also represents the number of bits needed to encode the result of one view of an ad, using the model to predict whether the ad would have been clicked or not). A perfect model would score 0. Our baseline model is to simply predict the average CTR on the training set. In all tables, we additionally provide the mean squared error (MSE) as a metric. Since the model is trained to optimize the KL-divergence, we report the % reduction in error on it. All of the improvements in this paper were found to be statistically significant ($p < 0.01$).

In preliminary experiments, we also measured the performance using boosted regression trees (we used MART: multiple additive regression trees [9]). They were found to have no significant improvement over logistic regression. Thus, for ease of interpretation and simplicity, we continued with logistic regression for the remainder of the experiments (we present only the logistic regression results here).

In the next section, we will discuss the first set of features, intended to capture the CTR variance inherent in the terms.

6. ESTIMATING TERM CTR

As discussed earlier, there is significant variation in the average CTR for different bid terms. Thus, when predicting the CTR for an ad, we expect that the CTR for other ads with the same, or possibly related, terms would be useful.

6.1 Term CTR

The first of our features is the CTR of other ads (not including those of the current advertiser) that have the same bid term. In order to handle ads whose term has not been seen before, we smooth these probabilities to the mean ad CTR (measured on the training set):

$$f_0(ad) = \frac{\alpha\overline{CTR} + N(ad_{term})CTR(ad_{term})}{\alpha + N(ad_{term})}$$

where *N(term)* is the number of ads with the given bid term (ignoring word order), *CTR(term)* is the average CTR for those ads, and $\overline{CTR}$ is the mean CTR for all ads in the train set. α sets the strength of the prior, in terms of number of views, and was set to 1 in our experiments (the results were relatively insensitive to variations in α). We also provide the logistic regression with N_{term} as a feature. These two features will be called the *Term CTR* feature set.[2]

The results are given in the first two rows of Table 1. In the first row is the baseline model, which has only one feature: $\overline{CTR}$. The second row shows that with the Term CTR features, we achieve a 13% reduction in error.[3]

Table 1: *Term* and *Related Term* Results

Features	*MSE (x 1e-3)*	*KL Divrg. (x 1e-2)*	*% Imprv.*
Baseline ($\overline{CTR}$)	4.79	4.03	-
Term CTR	4.37	3.50	13.28%
Related term CTRs	4.12	3.24	19.67%

6.2 Related Term CTR

As with Regelson and Fain [19], we wanted a way to take advantage of other ads that have related terms. For example, if the ad in question has bid on "red shoes", and another ad has bid on "buy red shoes", one would suspect that the CTR for the latter would be useful in predicting the CTR of the former. Rather than clustering terms, as they did, we took the approach of considering ads with subsets and/or supersets of the bid term.

Let $\mathbf{R}_{mn}(t)$ be the set of ads whose terms are the same as t when one removes m words from t and n words from the ad term (ignoring word order), and have at least one term in common. That is:

$$\mathbf{R}_{mn}(t) = \left\{ ad: \begin{array}{c} |ad_{term} \cap t| > 0 \text{ and} \\ |t - ad_{term}| = m \text{ and} \\ |ad_{term} - t| = n \end{array} \right\}$$

For example, if t is "red shoes", then an ad for "buy red shoes" will appear in $\mathbf{R}_{01}$, an ad for "shoes" will be in $\mathbf{R}_{10}$, and an ad for "blue shoes" will be in $\mathbf{R}_{11}$. Note that $\mathbf{R}_{00}$ is the set of exact-match ads, $\mathbf{R}_{m0}$ is any ad whose terms are missing m words (vs t), and $\mathbf{R}_{0n}$ is any ad that has n extra terms. We also let m or n take the value ∗, which means "any value". Hence, $\mathbf{R}_{0*}(t)$ is any ad whose terms are a superset of t, regardless of how many extra words it has.

Given $\mathbf{R}_{mn}$, we compute the following features for a given ad:

$$CTR_{mn}(term) = \frac{1}{|\mathbf{R}_{mn}(term)|} \sum_{x \in \mathbf{R}_{mn}(term)} CTR_x$$

which is the average click through rate of the set of related ads. As with the *Term CTR* feature set, we smooth $CTR_{mn}(term)$ using the

[2] Since logistic regression estimates the *logit* of the click-through-rate as a weighted sum of features, we actually provide it with the *logit* of the smoothed term CTR.

[3] On the validation set, we also tried weighing the contribution of each ad to the Term CTR by its number of views. The principle was that more views would result in a more accurate Term CTR. Unfortunately, because of the inherent bias that more views implies a better ad, this reduced the predictability of the model.

average: $\overline{CTR_{mn}}$, and take the logit of the probability. We also provide the count as a feature:

$$v_{mn}(term) = |R_{mn}(term)|$$

We compute these features for m,n $\in$ {0,1,2,3,∗}.

The results for adding the related terms are given in the third row of Table 1. As can be seen, adding the related term CTRs improves performance by an additional 6%, bringing the total error reduction to nearly 20% from the baseline.

7. ESTIMATING AD QUALITY

In the previous section, we tackled the problem of estimating the CTR of an ad based only on its terms. However, as we discussed earlier, even within a term there is significant variation in ad CTR. For example, the maximum CTR for an ad for digital cameras is more than 3 times greater than the average, and the maximum CTR for an ad for surgery is over 5 times higher than average as illustrated in Figure 3.

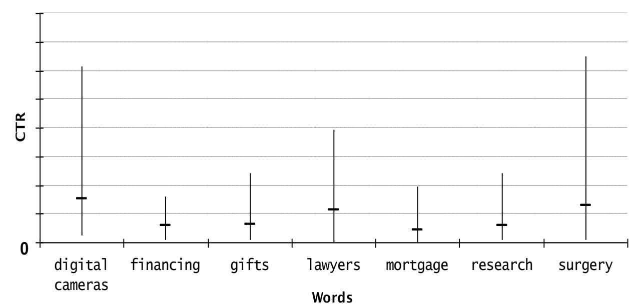

Figure 3. CTR variance across all ads for several keywords. Horizontal bars show average CTR; the bottom of the vertical bar is the minimum CTR, and the top is the maximum CTR.

In this section, we ask the question: can we use features of the ad itself to come up with an even better estimate for a given ad's CTR? The work of Jansen and Resnick [11] suggests that Web searchers consider the summary, the title, and the URL of an advertisement in deciding whether to click it. Besides these, what exactly causes a person to decide to click on an ad (or not)? We hypothesize at least four rough categories of influence on the user:

- **Appearance**: Is the ad aesthetically pleasing?
- **Attention Capture**: Does the ad draw the user in?
- **Reputation**: Is the advertiser a known or reputable brand? If the user is not familiar with the advertiser, would they guess that the advertiser is a good brand?
- **Landing page quality**: Though the landing page is only seen after the user has clicked the ad, we hypothesize that many ad clicks go to advertisers that a user is already familiar with (such as eBay, Amazon, etc). Thus, the quality of the landing page may be indicative of the probability the user will click the ad. And it is likely to result in repeat visits from users searching for new products.
- **Relevance**: How relevant is the ad to search query term

For each category, we derived a number of features that we hoped would be indicative of the quality of the ad for that category. For example:

- **Appearance**: How many words are in the title? In the body? Does the advertisement have good capitalization? Does it contain too many exclamation points, dollar signs, or other punctuation? Does it use short words or long words?
- **Attention Capture**: Does the title contain action words such as “buy”, “join”, “subscribe”, etc.? Does the body? Does the ad provide numbers (such as specific discounts, prices, etc)?
- **Reputation**: Does the display URL end with .com (similarly for .net, .org, .edu)? How long is it? How many segments are in the display URL (e.g., books.com is generally better than books.something.com)? Does it contain dashes or numbers? Because good, short, .com domain names (such as single-word names) can be expensive, some of these features can be seen also as estimating how well-established and/or large the advertiser is.
- **Landing page quality**: Does the page contain flash? What fraction of the page is covered with images? Is it W3C compliant? Does it use style sheets? Is it covered with ads?
- **Relevance**: Does the bid term appear in the title exactly? Do any subsets of the term appear in the title? In the body? What fraction of the body?

In all, we have 81 features across these five categories. Clearly, some features could be placed in multiple categories, such as the number of dollar signs in the ad, which may increase its “attention capture” but decreases its appearance.

We also add unigram features: For each of the most common 10,000 words in the ad title and ad body of the training set, we add a feature which takes the value 1 if the word exists in the ad and 0 otherwise[4]. These features are intended as an automatic way to capture some of the same influences that our manual features do. For example, it may find certain words that increase the attention capture of an ad that we may not have thought of. Figure 4 shows how significant the skew in unigram frequency skew can be for good vs. bad ads. The figure illustrates unigram word frequencies for all ads, ads with a CTR that is less than half of the average CTR, and ads with a CTR that is more than twice the average. For

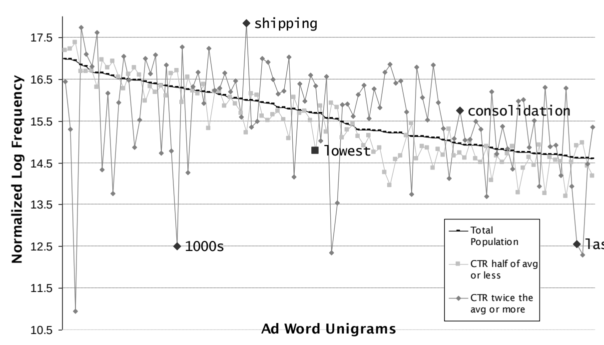

Figure 4. Frequency of advertisement word unigrams, sorted by overall frequency. The light and dark gray lines give the relative frequency of unigrams in low and high CTR ads.

[4] We also tried using the number of times the word occurred, or the log thereof, instead of the binary existence of the word. This did not improve the results. Using bigrams and trigrams also did not show significant improvement.

Table 2: *Ad Quality* Results

Features	*MSE (x 1e-3)*	*KL Divrg. (x 1e-2)*	*% Imprv.*
Baseline ($\overline{CTR}$)	4.79	4.03	-
Related term CTRs	4.12	3.24	19.67%
+Ad Quality	4.00	3.09	23.45%
+Ad Quality without unigrams	4.10	3.20	20.72%

instance, from the figure it can be seen the term "shipping" occurs much more commonly in high-CTR ads than over all ads.

We will refer to this set of features as the *Ad Quality* Feature Set. In Table 2, we give results for this feature set. As can be seen, it significantly improves performance, reducing the error by another 4% to a total of 23.45%.

A natural question is how much does each feature contribute to the gain. To determine this, we first removed the unigram features. To our surprise, this eliminated roughly three quarters of the performance gain (see Table 2, fourth row). With the non-unigram ad quality features alone, we only see a 1% improvement in model accuracy over the related term CTRs. This was surprising because we expected many of the manual features, particularly those to do with reputation, to strongly affect user CTR.

8. MEASURING ORDER SPECIFICITY

Finally, we wanted to look at how the CTR of an ad may vary depending on what variety of terms it was originally associated with. Recall that when an advertiser wishes to create advertisements, they enter an order, which is the text, title, etc, and a set of terms used to establish relevance to the user query. This generates N ads, one for each term in the order. The order text may also contain a parameter that could be filled in with the term. For example, an advertiser may enter an order of

```
Title:    Buy shoes now,
Text:     Shop at our discount shoe warehouse!
Url:      shoes.com
Terms:    {buy shoes, shoes, cheap shoes}.
```

This will result in three ads, all with the same title and text, but with different bid terms. Each pairing of the title/text/URL with a term is considered one ad, and thus far, we have considered each of these ads independently. However, we need not (since we have split our data by advertiser, all of the ads for a given advertiser appear in either our train or our test sets, thus there is no train-test contamination). In the above order, the advertiser is targeting shoe shoppers specifically. In some cases, the ads may be more broadly targeted. For example:

```
Title:    Buy [term] now,
Text:     Shop at our discount warehouse!
Url:      store.com
Terms:    {shoes, TVs, grass, paint}.
```

Because the second order generates less targeted ads (they are advertising a general store, to a broad variety of users, rather than a specific service, namely shoe selling), we might expect them to have a lower CTR than the ads generated by the first order.

In an attempt to capture how targeted an order is, we measure the category entropy of the terms. To categorize a term, we perform a Web search for it and run a text classification algorithm on the resulting result snippets (We use a naïve Bayes, trigram classifier trained on the Look Smart Directory structure). This classifies each term into one of 74 categories. We measure the entropy of the distribution of categories of the order bid terms, and use that as a feature for the model. Note that an advertising system with this feature would only need to query the search engine whenever an order is submitted that has novel terms; the load on the search engine is thus kept to a minimum.

Table 3: *Order Specificity* results

Features	*MSE (x 1e-3)*	*KL Divrg. (x 1e-2)*	*% Imprv.*
Baseline ($\overline{CTR}$)	4.79	4.03	-
CTRs & Ad Quality	4.00	3.09	23.45%
+Order Specificity	3.75	2.86	28.97%

We also add in a feature which is simply the number of unique terms in the order. Together with the entropy feature, this constitutes the *order specificity* feature set. The results give a significant lift. The order specificity feature set improves performance by another 5.5% (see Table 3), which is more than the improvement gain by the ad quality features (including unigrams). All together, the model reduces the prediction error by over 29%.

To study which of the two order specificity features was most useful, we tested the order entropy feature and order term count feature separately. With only the category entropy feature, the model achieves a 26.37% improvement over baseline (vs. 28.97% with both features and 23.45% with neither), thus indicating that both features provide significant gains.

9. EXTERNAL SOURCES OF DATA

We do not need to restrict ourselves to features that can be computed using the ad data (and ad landing pages) alone. Given an ad term, we could, for instance, look it up in an encyclopedia to see if it's a commonly known term or not, find synonyms of the terms in a thesaurus, etc. To this end, we added two more features to our model: the approximate frequency of the term occurring on the Web, and the approximate frequency with which search engine users query for the term.

For the first, we queried the search engine and noted the number of pages it claimed would contain the ad's term. Note that this is only an approximation; more advanced techniques have been proposed [14][15], but would have been infeasible to do for every ad in our data set.

As with Regelson and Fain [19], we found a relation between the frequency of an ad term as a search query and the CTR of the ad. This relation is shown in Figure 5 (note that the graph is not precisely comparable to that of Regelson and Fain, since the data available is slightly different). Thus, our second feature is the frequency with which users query for the ad term, based on a three

Table 4: *Search Engine Data* results. *AQ* means the *Ad Quality* feature set, and *OB* means the *Order Specificity.*

Features	*MSE (x 1e-3)*	*KL Divrg. (x 1e-2)*	*% Imprv.*
Baseline ($\overline{CTR}$)	4.79	4.03	-
+Search Data	4.68	3.91	3.11%
CTRs & AQ & OS	3.75	2.86	28.97%
+Search Data	3.73	2.84	29.47%

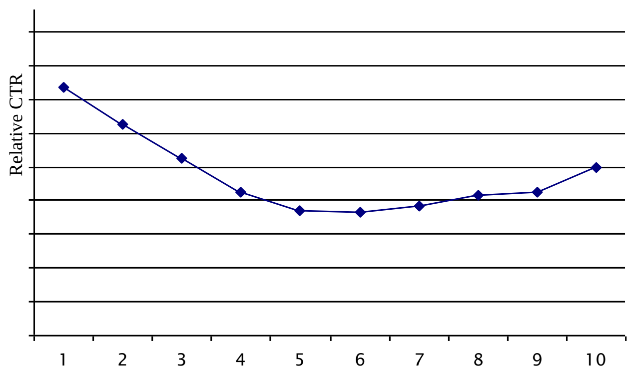

Figure 5. Relative average CTR for ads displayed for each query frequency decile (in decreasing order), aggregated across all ranks.

month period of search query logs. Additionally, we binned each feature into one of twenty bins. That is, each was converted into twenty binary features where only one of the twenty was set to true for each ad – the one that specifies the value range the feature falls in. The bin boundaries were set so that each bin had an equal number of ads.

Together, we call these the *Search Data* feature set. The results are summarized in Table 4. It is interesting to note that, while the search data does provide a useful improvement of 3% over the baseline, it shows almost no improvement (0.5%) when added to the feature sets discussed earlier. This implies there is overlap between some of the other features and the search data features. This is not an uncommon occurrence, and is also the case with many of the other features we have presented in this work.

10. DISCUSSION OF RESULTS

10.1 Utility of Features

The results we gave above, for the contribution of each feature set, were done in the context of having the related term CTRs and (sometimes) other feature sets. One interesting question is, how good is each feature set when considered purely in isolation. We measured the % reduction in KL-divergence (as in the previous tables) for each feature set when used without any additional term or related term CTRs, and no other feature sets. The improvements are: 12.0% for the ad quality feature set (10.2% for the unigram features alone), 8.9% for the order entropy set, and 3.1% for the search data set. As expected, if we include the term CTR in the baseline, we get similar, though reduced, improvements for each feature set.

Since the model is a logistic regression, we can see which features received the highest weight, and which have the lowest weight (or highest negative weight). These are given in Table 5. Note that the weight of a feature does not necessarily directly indicate its significance, because the features are not independent. For instance, the *termLength* feature (number of characters in the term) and the *termNumWords* feature (number of words in the term) are obviously very strongly correlated. If an important "feature" is actually the average length of each word, then the model might give *termLength* a high positive weight, and *termNumWords* a high negative weight to approximate this. However, it still can be interesting to observe the model weights to draw qualitative conclusions.

Table 5: Non-unigram features with highest (lowest) weight

Top ten features	*Bottom ten features*
log(#chars in term)	log(# terms in order)
v_{12}	$\log(v_{0*})$
v_{22}	$\mathrm{sqr}(p_{00})$
log(order category entropy)	sqr(order category entropy)
log(#most common word)	log(#chars in landing page)
sqr(#segments in displayurl)	$\log(a_{01})$
sqr(#action words in body)	a_{13}
p_{10}	$\mathrm{sqr}(p_{0*})$
p_{**}	log(#chars in body)
$\log(v_{00})$	sqr(#chars in term)

Table 6: Unigrams with highest (and lowest) weight.

Top ten unigrams		*Bottom ten unigrams*	
official	body	quotes	title
download	title	hotels	title
photos	body	trial	body
maps	body	deals	body
official	title	gift	body
direct	body	have	text
costumes	title	software	title
latest	body	engine	body
version	body	compare	title
complete	body	secure	body

It is also interesting to look at the unigram features that have highest and lowest weight (see Table 6). Qualitatively, the terms in the top ten seem to indicate more established entities: *official*, *direct*, *latest*, and *version* (If they use the word *version*, then at least there is more than one version of their product). The words in the bottom ten appear to be attempts to grab the consumer with deals: *quotes*, *trial*, *deals*, *gift*, *compare* (as in, compare different insurance companies, mortgages, etc). Qualitatively, then, it appears that consumers prefer to click on ads from more reputable, established entities, and tend to avoid clicking on ads with various free offers and trials.

Though it is interesting to determine the best features, and how much each feature may overlap with other features, we believe that ultimately, the best practice is to include as many feature sets as possible in the final model. Doing so provides additional robustness in adversarial situations (estimating ad CTR is adversarial because the advertiser wants us to estimate as high a CTR as possible, so they can be placed higher on the page). By including multiple overlapping features, an adversary must attack more features. Further, an adversarial attack may be easier to detect as feature sets that used to agree in CTR prediction begin to diverge significantly for advertisers that are attacking only a subset of the features used by the model.

10.2 Evolution After Initialization

Our model is able to predict the CTR of an ad with some accuracy. One question we might ask is, after how many ad views will the empirically observed clicks provide as good of an estimate of CTR as our model.

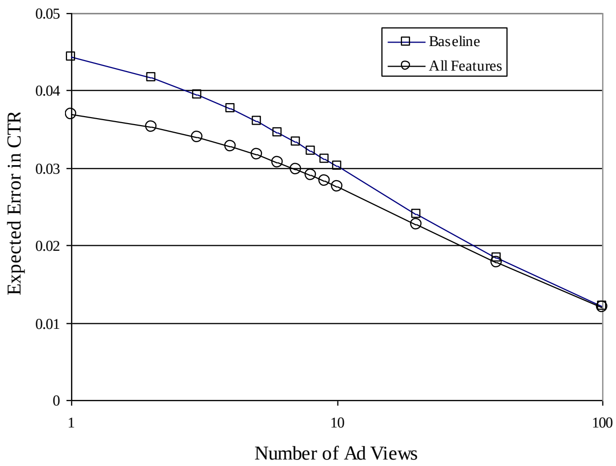

Figure 6: Expected mean absolute error in CTR as a function of the number of times an ad is viewed.

Assume that the predicted CTR and observed CTR are combined using the standard technique:

$$\hat{p} = \frac{\alpha p_0 + clicks}{\alpha + views}$$

Where $\hat{p}$ is our best estimate of the true underlying CTR of the ad, p_0 is the prior CTR predicted by the model, *clicks* is the number of times the ad has been clicked, and *views* is the number of times the ad has been viewed. α sets the strength of the prior, in terms of the equivalent number of views. This combination rule imagines that we have shown the ad α times and seen it clicked αp_0 times in addition to the actual empirical number of clicks and views.

We now can ask the question, what is the expected absolute error in the best-estimate CTR ($\hat{p}$) if we show the ad *views* times:

$$E[err \mid views] = \sum_{clicks=1}^{views} p(clicks \mid CTR, views) \cdot \left| \hat{p} - CTR \right|$$

P(clicks|CTR,views) is just the binomial distribution, and we use the absolute error in CTR as our error measure (note that this is not the same as the error measure our model optimizes; the results may be better if our model explicitly optimized mean absolute error). The results are given in Figure 6. On the y-axis is the expected error, in terms of absolute difference from the ad's true CTR. The x-axis is the number of times the ad is viewed.

As can be seen, the baseline and model predictions differ noticeably until around 50 to 100 ad views. Such a difference can result in ads being displayed in the wrong order, causing a reduction in revenue. This means that, though the effect may be diminishing, the model provides an advantage in CTR estimation for up to 100 ad views, or 200-300 search result pages (recall that, because the probability that an ad is viewed decreases rapidly with position on the page, the number of times an ad is actually displayed is many times the number of times it was viewed). For a system with millions of ads, incorrectly ordering them for the first 200-300 times each is displayed can result in a significant loss of revenue and user dissatisfaction.

Table 7: Comparison of results for a model trained and tested on ads with over 100 views vs. over 1000 views.

	%Imprv	
Features	>100 views	>1000 views
Baseline ($\overline{CTR}$)	-	-
+Term CTR	13.28	25.22
+Related CTR	19.67	32.92
+Ad Quality	23.45	33.90
+Order Specificity	28.97	40.51
+Search Data	29.47	41.88

10.3 Ads with Many Views

The choice of 100 views as a minimum cutoff for an ad to be in the data set was motivated by a desire for a reasonable level of confidence in the observed CTR for the ads used as examples. In some systems, however, it may be desirable to train on ads that have had a higher number of views. A higher cutoff produces training data with less noise, because the estimated CTR values are more accurate, but the examples may be biased away from the behavior of new ads. When we restrict the data to ads with at least 1000 views, the model achieves even more significant gains in CTR estimation accuracy, as shown in Table 7.

The strong performance of the system for ads that have been viewed many times indicates that the model extends well to ad populations of higher quality. The high cutoff implies that the ads chosen do not include ads that were estimated as low-value by the advertising system, since such ads would not be allowed to gather so many impressions. This is supported by our observation that the average CTR of ads with over 1000 views was over 40% greater than the average CTR of ads with at least 100 views.

11. DISCUSSION AND FUTURE WORK

While we find the results given here compelling, we acknowledge that current work in this area in a relatively early stage – it is both subject to the idiosyncrasies of the advertising and search system that was studied and not approachable by many researchers. In future work, it would be very desirable to settle on some standard data sets (and possibly testing frameworks), in order to increase the ease of comparison and repeatability characteristic of more mature areas of study.

One significant direction of future work is in making the CTR estimate dependent on the user's query. In this paper, we have been predicting the query independent CTR of an ad. In the case of *exact matching*, the bid term is identical to the user query. However, in the case of *broad matching*, the query may have some looser relation to the bid term (such as being a superset or containing synonyms of the bid terms). In this case, knowing what the query is may give additional insight into the expected CTR for each ad. The same model presented here could be used, with additional query-dependent features such as the similarity between the query and the bid term, number of words in the query, how many of those words appear in the ad text or landing page, etc.

We would also like to reproduce the term clustering techniques of Regelson and Fain [19] as additional features in the logistic regression; they are likely to provide additional information about related ads that is not captured by our *related terms* feature set. It would also be interesting to compare the two approaches by having just one, the other, or both in the model.

The model predicts the expected CTR for a new ad. Rather than using this for ad ranking purposes, we could also use it to inform advertisers what they should change about an ad they are creating in order to increase its CTR. For example, our model may show that their title is too short, or that they might want to remove the word "deals", etc. Bartz et al [3] propose suggesting terms to the advertiser in hopes of increasing their coverage and/or decreasing their cost per click.

We would also like to incorporate more features into our model. Those found to be useful for the static [20] and dynamic ranking [1] of Web pages might prove particularly beneficial. In particular, data on how often users have visited the ad's landing page or its domain (similarly, the display URL and its domain), how long they remain on that page, whether they click "back" or a link off of the page, etc. could prove useful.

Another source of information could be human judges. We would like to see if a brief (5 second) "instinctive" judgment by a person could be a useful feature to our model. Since the decision to click or not is based on even less than this amount of time on behalf of the end user, we belief such quick human judgments could provide significant value to the model, while incurring a low overhead cost.

Finally, we wish to consider more than just new ads and new advertisers. Over time, we can accumulate information about the general quality of an advertiser (this could be either independent or dependent of the terms that the advertiser has bid on). A time-dependent model such as this could be kept up-to-date with information about all advertisers, ads, terms, clicks, and views, and would have the power to update its the estimated CTR of all ads any time an ad is shown.

12. CONCLUSIONS

A good initial estimate of an advertisement's click-through rate is important for an effective online advertising system. We have presented a logistic regression model that achieves a 30% reduction in the error of such estimates. The error reduction comes from reducing two primary sources of variance: the variance in CTR across terms, and the variance in CTR within a term. For the former, the model contains information about ads that have related terms. For the latter, it contains information about the ad's quality, content, and how broadly it was targeted. The model is easy to understand, quick to train, and efficient enough to be used by any of the major search engines as an integral part of their advertising system.

13. ACKNOWLEDGMENTS

We would like to thank John Platt and Galen Andrew for useful discussions and insights.

14. REFERENCES

[1] E. Agichtein, E. Brill, S. Dumais, "Improving Web Search Ranking by Incorporating User Behavior Information", In World Wide Web, 2006.

[2] L. Baker, "Google vs. Yahoo : Earnings Reports Comparison," In Search Engine Journal http://www.searchengine-journal.com/?p=3923.

[3] K. Bartz, V. Murthi, S. Sebastian, "Logistic Regression and Collaborative Filtering for Sponsored Search Term Recommendation", In Proceedings of the Second Workshop on Sponsored Search Auctions, 2006.

[4] E. Burns, "SEMs Sees Optimization PPC", In ClickZ, http://www.clickz.com/showPage.html?page=3550881

[5] Did-it, Enquiro, and Eyetools, "Eye Tracking Study", http://www.enquiro.com/eye-tracking-pr.asp

[6] B. Edelman, M. Ostrovsky. "Strategic bidder behavior in sponsored search auctions." In Workshop on Sponsored Search Auctions, ACM Electronic Commerce, 2005.

[7] D. Fain and J. Pedersen. "Sponsored Search: a Brief History", In Proceedings of the Second Workshop on Sponsored Search Auctions, 2006.

[8] J. Feng, H. Bhargava, D. Pennock, "Implementing Sponsored Search in Web Search Engines: Computational Evaluation of Alternative Mechanisms" In Informs Journal on Computing, 2006.

[9] J. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine," Technical Report, Dept. of Statistics, Stanford University, 1999.

[10] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer-Verlag, New York, 2001.

[11] B. Jansen and M. Resnick, "Examining Searcher Perceptions of and Interactions with Sponsored Results," In Proceedings of the Workshop on Sponsored Search Auctions, 2005.

[12] B. Kitts, P. Laxminarayan, B. LeBlanc, R. Meech, "A Formal Analysis of Search Auctions Including Predictions on Click Fraud and Bidding Tactics", In Workshop on Sponsored Search Auctions, ACM Electronic Commerce, 2005.

[13] S. Kullback, R. A. Leibler, "On Information and Sufficiency", Annals of Mathematical Statistics, Vol. 22, No.1, pp. 79-86, 1951.

[14] S. Lawrence, C.L. Giles, Searching the World Wide Web, Science 280, pp. 98-100, 1998.

[15] S. Lawrence, C. L. Giles, Accessibility of information of the Web, Nature 400, pp. 107-109, 1999.

[16] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," Mathematical Programming, vol. 45, no. 3, pp. 503–528, 1989.

[17] D. Murrow, "Paid Search Ad Spend Will Hit $10 Billion By 2009" In eMarketer, http://www.emarketer.com/Article-.aspx?1003861.

[18] J. Nocedal and S. J. Wright, Numerical Optimization. Springer-Verlag, 1999.

[19] M. Regelson and D. Fain, "Predicting click-through rate using keyword clusters," In Proceedings of the Second Workshop on Sponsored Search Auctions, 2006.

[20] M. Richardson, A. Prakash, E. Brill, "Beyond Page Rank: Machine Learning for Static Ranking, In World Wide Web, 2006.