Ranking for the Conversion Funnel

Abraham Bagherjeiran, Andrew Hatch, Adwait Ratnaparkhi
Yahoo!
Santa Clara, CA, USA
{abagher,aohatch,adwaitr}@yahoo-inc.com

ABSTRACT

In contextual advertising advertisers show ads to users so that they will click on them and eventually purchase a product. Optimizing this action sequence, called the conversion funnel, is the ultimate goal of advertising. Advertisers, however, often have very different sub-goals for their ads such as purchase, request for a quote, or simply a site visit. Often an improvement for one advertiser's goal comes at the expense of others. A single ranking function must balance these different goals in order to make an efficient system for all advertisers. We propose a ranking method that globally balances the goals of all advertisers, while simultaneously improving overall performance. Our method has been shown to improve significantly over the baseline in online traffic at a major ad network.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Classifier Design and Evaluation

General Terms

Experimentation, Measurement

1. INTRODUCTION

An advertiser creates an online ad for one of two main purposes: brand awareness or performance. In brand awareness, the advertiser simply wants to make the user aware of some message but does not expect the user to take any immediate action such as an ad click. In performance advertising, the advertiser wants the user to click on the ad so that the advertiser can offer a product or service. Traditionally contextual advertising has been associated with performance advertising where advertisers pay per click.

Although they pay for clicks, for some advertisers, showing an ad because a click is likely is not enough. When a user visits a web page, an ad search engine searches a database of ads with the content of the page as the query [2]. Ads are then ranked to optimize for the click rate or CTR [3]. Budget-conscious advertisers would prefer to spend their limited budget only on clicks that will convert.

A conversion is the business term for an action that has some benefit to the advertiser and happens after the click. In order for a conversion to occur a specific set of events, called the conversion funnel, has to occur. As shown in Figure 1, the ad impression is served, the user clicks on the impression, visits the advertiser's site, and then converts. A user is then said to *convert* from a regular user into a potential business lead or customer. However, not all conversions have the same meaning or value to all advertisers. Some advertisers want the user to purchase some product whereas others want the user to request information about a product by signing up for a newsletter. Both actions are important to the advertisers but if we treat them as equivalent we would say that the latter is more important because it is more frequent.

Our ranking problem is as follows: how to rank ad impressions for conversions but still charge for clicks. Within the conversion funnel, ranking can only influence the impression. The ideal ranking would rank impressions into three groups: conversions followed by clicks followed by non-clicking impressions. Since advertisers define conversions differently, the ranking should also take into account the importance of conversions when ranking the ads.

Our main contributions are as follows. First, we discuss some of the challenges and constraints in optimizing for conversions. Second, we propose generally useful feature construction methods for dealing with differently valued examples. Finally, we show how to rank documents into groups when the groups are highly imbalanced.

The rest of the paper is organized as follows. Section 2 gives a more detailed introduction to contextual advertising and conversions. Section 3 discusses related work. Section 4 describes the feature construction. Section 5 describes the ranking method. Finally, Section 6 describes the offline and online experiments.

2. CONTEXTUAL ADVERTISING

The following section provides a general overview of the contextual advertising system and ranking. The system operates as follows: Let's say that a user navigates to a web page that serves contextual ads. We refer to the event of user u viewing page p as an *impression*. For every impression of p, the system retrieves a set of candidate ads from an ad index. The candidate ads are selected to maximize the

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.

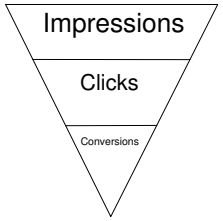

Figure 1: The conversion funnel is the sequence of events that a user experiences after visiting a web page with contextual ads. The size of each segment indicates the relative volume of the event. Typical rates are on the order of 1 click in 1,000 impressions and 1 conversion in 100 clicks–1 conversion in 100,000 impressions.

degree to which various terms and features in the ad match the given page.

A *click model* is then used to estimate the click probability for each ad a. The ads are then ranked according to their *expected cost per mille* (ECPM):

$$P\left(click\left(a\right)|p,u,a\right)V\left(a\right)$$

where (p, u, a) denotes the context of the impression and $V\left(a\right)$ is the advertiser's *bid,* which represents the maximum cost-per-click of the ad. For the remainder of the discussion, the context is implicit in each expression, so we will denote $P(click(a) \mid p, u, a)$ as $P(C)$. Given the ECPM-ranked set of ads, the system returns the top k ads to be displayed on the given page. The number k is determined by the publisher and is typically equal to 3 or 4 ads.

2.1 Conversion Funnel

The conversion funnel, depicted in Figure 1, is a three-step process. Just before a web page is rendered to a user, an ad call is issued to a server. The ad server selects an appropriate slate of ads which are rendered on the page. When the user sees these ads, we say that an impression has occurred. The user views the page and the ad and decides whether to click. If the user clicks on the ad, the user visits the landing page and other pages created by the advertiser.

An advertiser places a beacon on a conversion page, which is a specific page on his or her web site. The beacon fires when a user visits this page after having first clicked on an ad. The conversion event is associated only with the page visited. It is up to the advertiser to decide what is on the page. The page could be the receipt page after purchasing a product, but it could also be a visit to a product description page. In these cases, the conversion is attributed to the latest click and all previous clicks are designated as supporting clicks.

The key distinction between clicks and conversions is that clicks are defined and instrumented by the ad network but conversions are defined and instrumented by advertisers. Like clicks, conversions have different values for the advertisers. Unlike clicks, however, conversions are defined at the discretion of the advertiser and their rates relative to clicks vary considerably. For example, one advertiser's conversion could be filling out a form for more information while another advertiser's conversion could be to purchase the advertised product.

2.2 Pricing for Contextual Advertising

An ad server plays a dual role in contextual advertising, both making the market and participating in it. The ad server's market-making role is to select the appropriate ad a to maximize revenue for the publisher. It conducts an auction wherein each advertiser submits a bid for the impression and the highest bidder wins. Ad servers are usually run by ad networks so the ad server not only conducts an auction but also sets the bids on behalf of the advertiser. The ad server consults a click model to estimates $P(C)$ and then computes the following bid for each impression:

$$B(a) \quad = \quad \hat{P}(C)V(C)$$

where $V(C)$ is assumed to be the true value and bid for the click. The auction run by the ad server is a generalized second-price auction which means that the true price paid is less than $V(C)$ [4]. The final price charged by the ad network for each click is:

$$V(C_2)\frac{\hat{P}(C_2)}{\hat{P}(C_1)} \quad \leq \quad V(C_1) \tag{1}$$

where $\hat{P}(C_i)$ is the estimated click rate for ad a_i. In this pricing formula, the advertiser receives a discount if the ad is more *clickable* than the next ad. The discount is greater the more clickable the ad is.

2.3 Ranking for Ad Serving

When deciding what ad to serve, the ad server has considerable flexibility in ranking. Because it acts on behalf of the publisher and advertiser it must find a compromise behind highly clickable ads and high value ads. The pricing formula above gives us the method to achieve both these goals. We will show how the ad server can decrease the expected costs to the advertiser by better predicting probability of click.

A performance-focused advertiser is only interested in one thing–maximizing the return on the advertising investment. Performance ads are designed to be measurable and accountable such that each ad has an expected value $E\left[V(a)\right]$. An impression can result in one of three outcomes: user does not click (I), user clicks but does not convert (C), user clicks then converts (N). The advertiser has a value defined for each possible outcome. The expected value of an impression for an advertiser is as follows:

$$E\left[V(a)\right] \quad = \quad P(I)V(I) + P(C)V(C) + P(N)V(N) \tag{2}$$

In a market that sells impressions (cost-per-impression) we would expect the bid to converge to the expected value $E\left[V(a)\right]$.

Because the final cost of a click is determined not by the expected value in Equation 2 but by the pricing formula from Equation 1, the discount applied to the bid of ad a_1 is proportional to the difference in quality of the ad. A new ranking method can significantly alter the cost of a click with only a slight change to the scores of a pair of ads. Consider the following example, two ads have the same bid for a click and the same CTR. The cost of a click on a_1 is equal to its bid because $P(C_2) = P(C_1) = P(C)$. If a new ranking method changes $P(C_1)$ to $\alpha P(C_1)$ for $\alpha > 1$ but leaves $P(C_2)$unchanged, then the cost of clicks to a_1 decrease exponentially by a factor α, but the overall rank does not change. In the case of ranking for conversions, we would like to separate converting clicks from non-converting

clicks with a wide margin. This will cause the price of some non-converting click to increase but the cost of converting clicks will decrease. Our proposed ranking methods will have to be careful not to make the price differences too extreme.

3. RELATED WORK

Contextual advertising and ranking in this field have recently received much attention in the literature. Most major online ad networks (Yahoo!, Microsoft, and Google) offer some contextual advertising service with a pay-per-click (PPC) system, where the advertiser is charged a fee every time a user clicks on an ad. These systems use *click models* to automatically estimate the probability that a given ad impression will receive a click. Click models are typically trained on a variety of different signals. For example, most click models incorporate historical click-through-rate (CTR) information for specific ads or for specific page-ad pairs. Only recently has post-click behavior emerged for textual advertising, where it was shown that the page visited after clicks are somewhat relevant to predicting post-click activity [1].

A number of studies have used the co-occurrence of words and phrases within pages and ads to measure ad relevance (for example, see [9],[5],[2]). In these studies, the problem of matching ads with pages is translated into a similarity search in a vector space. Each page or ad is represented as a vector of features, which can include words and phrases, along with higher-level semantic classes (see [2]). The matching problem then reduces to the task of finding the set of ads that are closest to a given page.

In [9], Ribeiro-Neto et al. examine vector-space representations where the page and ad vectors are based on extracted keywords. The authors show that the vocabulary used in ads does not always match the vocabulary used in pages; hence, there exists an *impedance mismatch* between pages and ads. A number of techniques have been proposed to correct this mismatch. For example, in [9], the authors use a form of *query expansion*, where the page vocabulary is augmented with terms from other similar pages. Other studies have examined the use of semantic classes to match pages with ads. For example, in [2], Broder et al. map pages and ads to a common set of semantic classes, which are then used as features in a vector-space model. In [8], Ratnaparkhi introduces a page-ad probability model in which semantic relationships between page terms and ad terms are modeled with hidden classes. Another approach uses features from machine translation techniques to improve the matching between pages and ads (see [7]).

4. FEATURE CONSTRUCTION

Our training data consists of click logs from a major contextual ad network. Each log entry consists of a page-ad pair and a class label indicating whether the ad was clicked, converted or not. We extract the following raw features from the logs:

- HTML of the web page.
- Text of the ad creative.
- Keywords bidded on by the advertiser.
- Metadata such as price paid, id of the advertiser, *etc.*

In this section we discuss methods to construct higher-level features from the logs.

4.1 Matched Keywords

Advertisers annotate their ads with keywords in addition to the words that are shown to the user. For example, the title of an ad could be "Discounts on car insurance" but the advertiser wants to show the ad on pages about new cars, so the advertiser adds keywords such as "2010 honda" or "mazda 3". These additional words are added to the term vector of the creative. The intensity of the terms for the ad is the relative frequency with which the term occurs in the ad.

A publisher chooses to place text on a web page to provide information to the user. A similar feature extraction function is created for a web page. Terms are extracted from different zones of the page such as the title, bold text, headings, *etc.* For each term, the final term frequency is the weighted combination of its frequency of the term in each zone.

Matched keywords are defined as the intersection of the terms that occur both in the page and in the ad. The intensity of the matched features are the product of the intensities of the page and ad features.

4.2 Site-Level Interactions

Although semantic similarity between page and ad influences the propensity to click, we have observed that there is little influence on conversions. We consider what other factors might be useful for discriminating conversions from clicks.

One obvious feature is the quality of the ad. If the ad comes from a reputable advertiser then, a user is likely to find what he or she needs. Alternatively, many advertisers engage in click arbitrage. A click on their ad leads to a page with more ads. Arbitrageurs implement a distributed price setting mechanism in the marketplace. They will buy unused inventory by bidding on keywords on pages that do not have any other ads, but are priced low. They then show other ads (or in some cases more of the same ads) that are presumably priced higher. These ads may themselves be very appealing to the user but of poor quality in that the advertiser is not actually selling the product advertised.

An additional feature is the site hosting the page. Consider how users arrive to different web pages. If the page is shown on a known shopping site, then the user is "in the mood" to purchase a product. If the user is reading the news, the user might simply ignore the ad altogether or click by accident. In these cases, the content on the page may be similar, as in a site for booking plane travel and an article that mentions a tragic airplane crash. We can construct features that condition the text matches on the domain of the web page.

Post-click features describe what is to the user after clicking on the ad. For example, the page immediately after the click–the landing page has some influence over actions. The specificity of the ad and whether it is a call to action or simply brand awareness also play a large role in conversions.

Much of the information provided by features such as ad quality and post-click features is highly correlated with the web page hosting the ad. Quality pages promote competition among ads such that poor-quality ads tend to self-select themselves off the page. Low quality pages often con-

tain questionable content and easily detected by the domain name.

4.3 Normalized Conversions

Unlike clicks, each advertiser defines its own conversions differently. This means that ranking must be aware of the different definitions. Consider the example of two advertisers. Advertiser a measures the number of purchases for a high-end shoe, where each conversion is worth \$10 in profit. Advertiser b simply collects e-mail addresses for the shoe mailing list, each list is worth about \$1. Both ads are similar in content and have similar CTR. If we are ranking ads by click propensity, we would be indifferent to the two ads. However, we can significantly improve the ROI for advertiser a if we can rank based on the value of a conversion.

Instead of eliciting private information from the advertisers about the sales value of conversions, we can normalize across all advertisers. The cost to an advertiser for each conversion is measured by the average cost per action (CPA), defined as follows:

$$\text{CPA}(a) \quad = \quad \frac{\text{cost to adv.}}{\#\text{of conversions}}$$

where the cost is the total amount paid for all clicks on a over some time period. We do not directly observe the true value $V(N)$, but we can approximate the relative difference as follows:

$$\frac{V(N(a))}{\sum_a V(N(a))} \quad \approx \quad \frac{\text{CPA}(a)}{\sum_a \text{CPA}(a)} \tag{3}$$

where $a \in A$. This measure tells us that if an advertiser a pays more for the conversions relative to other advertisers, then a must value these conversions with a similar relative value. When evaluating the overall impact of a ranking system we will use this normalized measure.

4.4 Feature Representation

The final feature representation contains:

- Matched Keywords: unigrams and frequently occurring n-grams. Features are TF-IDF normalized and must occur on both the page and ad text.
- Site Name: Hostname of the publisher's site on which the ad was shown.
- Site-Specific Matched Keywords: Keywords matched conditioned on each site. For example, if "Mazda" was matched on an ad on the site autos.yahoo.com, then the feature would be "Mazda:autos.yahoo.com".

5. RANKING FOR CONVERSIONS

The purpose of conversion modeling in contextual advertising is to improve the ranking of ads so as to improve the conversion rate without sacrificing CTR. In addition we would like to decrease the cost of a converting click.

Several constraints are imposed on the solutions for conversion modeling in our production environment. First, the ranking is generally for a pay-per-click advertising system, so ranking for clicks is still very important. The conversion model is intended as a replacement for a click model, which means that it must outperform a comparable click model with respect to clicks. Second, the conversion model must not use any second-order features, such as the score of component models or data that is not accessible at run time. Finally, several other components require that the score output by the model is the probability of a click.

5.1 Baseline Click Model

We are restricting our discussion of learning algorithms to linear models. Regularized linear have been shown to be effective for text data in many studies. Linear models are also efficient to train and score.

5.2 Problem Description

We seek to induce a very weak partial order on the set of impressions. The rank among examples in each group are not constrained, but across the different groups conversions should precede clicks which should precede impressions. The click modeling problem is defined as follows. When a user visits a page p, a sequence of text ads $a = \langle a_1, \ldots, a_k \rangle$ will be displayed on the page each at a position $1 \leq i \leq k$. We denote an impression as the tuple (p, a). For modeling, the impression is decomposed into several examples, one for each displayed ad such that $x = (p, a_i)$ for an ad displayed on the page in position i. We assume that there is a class label defined over the examples, l which is 0 for non-clicks, 1 for clicks, and 2 for conversions. We define the following sets:

$$\begin{aligned} I &= \{(x, l) \mid l = 0\} \\ C &= \{(x, l) \mid l = 1\} \\ N &= \{(x, l) \mid l = 2\} \end{aligned}$$

where examples are pairs (x, l). Click modeling is thus posed as a classification problem such that the output of the classifier $g(x)$ approximates the posterior probability $p(x_l = 1 \mid x)$. Ads are ranked according to this score given a page impression. Conversions are actions that occur after the click, so the set of conversions $N \subseteq C$ is entirely contained in the set C. In general, conversion modeling aims to find a classifier with decision function $g : \mathcal{P} \times \mathcal{A} \rightarrow \mathbb{R}$ such that $g(x) \geq g(x')$ for $x \in N$ and $x' \notin N$.

5.2.1 Conversion v. Rest Model

A naïve approach to conversion modeling is to predict $P(l = 2 \mid x)$ directly. For this model, we define the positive and negative class labels as follows:

$$l' \quad = \quad \begin{cases} 0 & x \notin N \\ 1 & x \in N \end{cases}$$

where N denotes the set of converting clicks. This class label definition describes exactly what we want in the model, namely that conversions be ranked higher than all other impressions and clicks. The model treats all non-conversions equally, so clicks are treated as negatives.

There are several potential problems with a model like this, which is why we refer to it as the naïve model. First, it treats non-converting clicks as negative examples. Just because a click is not a conversion does not mean it is undesirable. Indeed, advertisers pay per click rather than conversions. A model that performs well at the conversion prediction task may be arbitrarily bad for clicks.

5.2.2 Click Re-Weighting Model

A conversion model must improve the conversion rate but not sacrifice CTR. A standard click model treats all clicks

equally. In reality clicks are not equal; some clicks lead to conversions. These clicks should have a higher score and thus rank. Because advertisers have different values for their conversions, we weight conversions by their cost. We assume that the most expensive conversions (with respect to CPA) are the most important. We define a weight for each example as follows:

$$w(x) = \begin{cases} \frac{1}{2} & \text{if } x \in I \\ 1 & \text{if } x \in C \backslash N \\ w(A(x)) & \text{if } x \in N \end{cases}$$

where $A(x)$ is the ad id of the pair and $w(A(x))$ is the relative conversion value defined in Equation 3. The weights of the conversions are then normalized to satisfy the following constraint:

$$\sum_{x \in N} w(x) = |C|$$

where $|C|$ is the total number of clicks. The total weight of all examples can be adjusted to produce well-calibrated probabilities.

The main advantage of this model is that it still predicts clicks. The disadvantage is that predicting clicks, even with re-weighting, does not guarantee performance on conversions. For example there is no constraint that a conversion should be ranked higher than a non-converting click.

5.2.3 Ordinal Regression

The ideal ranking of page-ad pairs is: conversions, clicks, and impressions. Ordinal regression is well-suited to this problem because the class labels have a preference and the class labels are nested, such that $N \subseteq C \subseteq I$. Recent work gives a reduction from ordinal regression to binary classification. In the reduction, we learn multiple parallel hyperplanes separating the different classes [6]. With a single linear decision function $g(x)$ we have the desired ranking $g(n) \geq g(c) \geq g(i)$ for all conversions n, clicks c, and impressions i.

In the ordinal regression formulation, the class label l takes values such that $l \in \{0, 1, 2\}$ where impressions have label 0, clicks have label 1, and conversions have label 2. The ordinal regression problem can be reduced to a binary classification problem by extending the feature space such that $\mathcal{X}' = \mathcal{P} \times \mathcal{A} \times \{b_{01}, b_{12}\}$, such that dataset now contains the following examples:

$$\begin{aligned} &((x, b_{01}, b_{12}), l) \\ &\quad ((x, 1, 0), 0) \qquad \forall_{x \in I} \\ &\quad ((x, 1, 0), 1) \qquad \forall_{x \in C} \\ &\quad ((x, 1, 0), 1) \qquad \forall_{x \in N} \\ &\quad ((x, 0, 1), 0) \qquad \forall_{x \in I} \\ &\quad ((x, 0, 1), 0) \qquad \forall_{x \in C} \\ &\quad ((x, 0, 1), 1) \qquad \forall_{x \in N} \end{aligned}$$

where the feature vectors are simply copied and have the new features appended. As shown in Figure 2, the bias parameters b_{01} and b_{12} cause the model to learn an offset between the different classes. The key assumption in the parallel hyperplane approach is that the same features that discriminate conversions also discriminate clicks. Therefore we can view the training procedure as learning two separate tasks: clicks and conversions versus impressions, conversions versus clicks and impressions. In learning these separate tasks we constrain the weights to be the same for both models– information from both tasks shapes the weights. The offsets translate the different sets into the same range.

One difficulty with the original ordinal regression formulation is that it assumes that classes have an equal number of examples. In our problem, the imbalance between the different sets is severe. Without some tuning the model will converge to the degenerate case where the conversion hyperplane is the same as the click hyperplane. A popular solution for class imbalance in binary classification is to simply increase the relative weight on the rare class. This strategy can be extended to ordinal regression by weighting the examples as follows:

$$w(x) = \begin{cases} \frac{1}{2} P(C \mid I) & x \in I \\ 1 & x \in C \\ \frac{1}{2P(N|C)} & x \in N \end{cases} \quad (4)$$

where the probabilities are over all ads. With this weighting scheme examples in each group have roughly the same importance. The factor of 2 is necessary because conversions appear as positive examples twice and impressions appear as negative examples twice. Because clicks appear as both positive and negative examples, there is no need to adjust their weights. As we will see later in Figure 3 this theoretically justified value also happens to be a good break-even point empirically.

To score a new example, the ordinal regression model operates as a binary classifier where only one bias is used. The resulting model assigns higher scores to conversions without the bias.

There are several advantages to the ordinal regression model. It guarantees (in training) the desired ranking. Second, it can be easily deployed as a drop-in replacement for an existing linear model that optimizes for clicks. The scores are reasonably well-calibrated to clicks. We expect any score difference to result in a larger score for converting clicks than non-converting clicks and impressions. Unfortunately this may mean that clicks become less well-separated from each other and the cost will increase. Finally, unlike pairwise classifiers, the training is still linear in the size of the original training data.

6. EXPERIMENTAL RESULTS

Conversion models must be evaluated as a replacement for the click models. As such, any conversion model must have a comparable performance on clicks but simultaneously improve performance relative to conversions.

6.1 Offline Evaluation

A conversion model is evaluated as a classifier for multiple learning tasks. We measure the area under the ROC curve, AUC, on the test set as a single measure of performance for the conversion models in the offline analysis. One conversion model is considered superior to another if it improves on all metrics.

6.1.1 Training Data

For the offline analysis, we collected impression logs from a major ad network over 7 weeks. We used the first 6 weeks for training and the following 1 week for evaluation. In the training data, there were approximately 200,000 clicks,

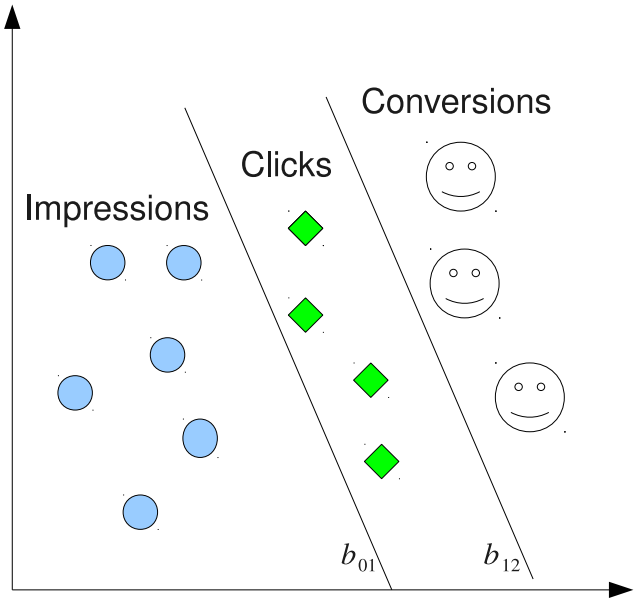

Figure 2: Ordinal regression reduction to binary classification. Two parallel hyperplanes are learned to jointly classify examples in each group.

10,000 conversions, and 2B impressions. Impressions were sub-sampled uniformly at 2%.

6.1.2 Results

Table 1 compares the conversion models on offline data. All models improve over the baseline click model for predicting conversions. This is not surprising because the baseline is unaware of conversions. The conversion v. rest model has the best performance at predicting conversions but is inferior to the click model for predicting clicks. This is because non-converting clicks are treated as negative examples. The model separates conversions from clicks but because it is unaware of clicks, it loses significant performance in separating clicks from impressions. The re-weighting model improves on conversions relative to clicks but is nearly identical with respect to clicks. Finally, the ordinal regression improves further on the re-weighting model in the conversion v. click task, with only slight decrease in click performance.

The results provide some insight on the relationship between clicks and conversions. In the conversion v. rest model, although we can distinguish conversions from clicks, we could not distinguish clicks or conversions from impressions. Because conversions logically occur after clicks, a good strategy for finding conversions is simply to find the clicks. Clearly the information in the clicks helps distinguish conversions. The other two models utilize this information. The re-weighting model still predicts clicks, so it only implicitly uses the conversion information. The ordinal regression relies heavily on the clicks to distinguish conversions because it learns both tasks jointly.

6.1.3 Tradeoff Analysis

Table 1 also demonstrates a key finding in the conversion ranking problem–a tradeoff between the click and conversion performance. In the ordinal regression model, the importance of the classes can be adjusted with an appropriate

	Conv Only	Re-weight	Ordinal
C v. I	-24.33%	0.04%	-0.78%
N v. I	-9.27%	0.47%	2.09%
N v. C	25.29%	2.46%	10.65%

Table 1: Comparing conversion and click models in offline results, where C is clicks, I is impressions, and N is conversions. Performance metric is percent gain in ROC area relative to the click model baseline.

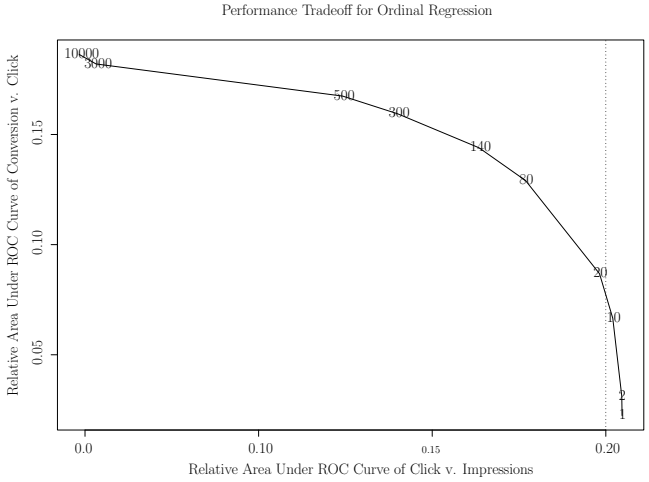

Figure 3: Comparing the performance of ordinal regression for predicting clicks and impressions. Each label on the curve indicates the relative importance of the conversions v. clicks. The vertical line indicates the break-even point in performance.

weight factor. We trained and evaluated several models with different values for the weight of conversions. As shown in Figure 3, as the weight given to the conversions increases so does the performance. However, this improvement of prediction for conversions comes at the expense of predicting clicks. Although any tradeoff is Pareto-optimal, we select the break-event point at which the model meets the performance of the baseline click model with respect to predicting clicks. From the previous discussion of the effect of weights, we see that to meet the performance of the click model and click should have the same importance in the ordinal model as in the click model during training. This corresponds to the case when the weights are balanced as in Equation 4. The best compromise in performance comes at this recommended value of weights at the class prior. We chose this weight for the subsequent analysis.

6.2 Online Evaluation

The online evaluation of a model tests the model on a sample of real traffic. The following metrics are computed for conversion models:

$$\text{CTR} = \frac{\text{clicks}}{\text{impressions}} \qquad \text{CVR} = \frac{\text{conversions}}{\text{clicks}}$$

$$\text{CPC} = \frac{\text{cost to adv.}}{\text{clicks}} \qquad \text{CPA} = \frac{\text{cost to adv.}}{\text{conversions}}$$

where the impressions, clicks, and conversions come from those actually shown on pages to users. The cost to the

	CTR	CPC	CVR	CPA
Ordinal	-2.46%	7.00%	7.98%	-0.91%

Table 2: Relative performance of ordinal regression model versus the click model baseline in an online test.

advertiser is the total amount charged to the advertiser for the clicks.

6.2.1 Online Testing

In the online tests, our models were used to rank ads and serve impressions for a uniform sample of 2% traffic for each model. Because conversions are very rare, the model ran online for 4 weeks to collect enough conversions for significance testing.

6.2.2 Results

The ordinal regression model was selected for bucket test against the baseline click model. Table 2 shows the change in performance relative to the click model. The online results are consistent with the offline results. CTR decreases slightly but there is a significant increase in CVR (p-value less than 1%). The CVR metric is computed with the normalized conversion method described in Section 4.3. The cost per click (CPC) has increased as well. There are two main causes for this. First, the model ranks conversions higher than clicks, which means that non-converting clicks are ranked lower. The advertiser has to either improve the conversion rate or increase the bid to compete. Consequently, the advertiser has to pay a premium associated for ads that do not lend themselves to conversions, but is discounted for converting clicks. Secondly, it was noted during the analysis that the score distribution (not shown) of the estimated $P(C)$ measure is slightly skewed upwards. The scores from the ordinal regression model are over-estimating the click probability because of the weighting of the examples, but can be easily adjusted with a corresponding corrective factor on the bid. Although the CPC has increased, the cost per conversion (CPA) has decreased slightly. This is desirable and to be expected because the improved performance makes converting click cheaper. However, because there are many more non-converting clicks, the impact to the CPA is overshadowed by the CPC changes. Again this is a simple correction.

As we have seen from the overall metrics, there is a confounding relationship between predictive performance and overall ranking. Because the ranking is multiplied by the bid and there are numerous business constraints regarding the ranking of ads, the overall metrics do not convey the entire picture. The plots in Figure 4 show the precision recall curves when ranking by the model scores $\hat{P}(C)$ as logged in the impression events during the online test. The performance is not as good as we would have expected from the offline analysis discussed earlier. This discrepancy is because the final ranking is actually the ECPM: $\hat{P}(C)V(C)$. We see from the plots in Figure 5 that when ranking by ECPM, the performance improves. Despite the improvement, it is misleading to conclude that the bid is responsible solely for the improvement. The ECPM ranking biases the results toward those examples with a high bid, which is the result of position bias. However, even with this bias, the model improves over the baseline. The bid improves the performance of the conversion model more relative to the click model. The primary reason for the further improvement is that the ECPM measures the inherent worth of an impression. In the case of a conversion model, the score can take into account the fact that conversions are worth more than click even if only the relative value is being modeled.

7. CONCLUSION

In performance advertising, advertisers are ultimately interested in finding users who would like to buy their products. An ad ranking method that optimizes for the conversion funnel must simultaneously optimize both for clicks as well as for the conversions for all advertisers. We show that although the conversion goals for advertisers are very different, they can be modeled with an ordinal regression ranking function. Our ordinal regression provides a single linear model that ensures conversions are ranked higher than clicks and clicks are ranked higher than conversions. The overall performance of the global ranking function can be controlled to balance the performance of conversions versus clicks. The models were shown to be effective both in offline log data and in online tests. The model not only improves on the ranking of conversions but also makes conversions cheaper for the advertisers. This brings efficiency to the advertising marketplace as advertisers can tailor their ads towards conversions.

8. REFERENCES

[1] Hila Becker, Andrei Z. Broder, Evgeniy Gabrilovich, Vanja Josifovski, and Bo Pang. What happens after an ad click?: quantifying the impact of landing pages in web advertising. In *CIKM*, pages 57–66, 2009.

[2] Andrei Broder, Marcus Fontoura, Vanja Josifovski, and Lance Riedel. A semantic approach to contextual advertising. In *SIGIR 2007*, pages 559–566. ACM, 2007.

[3] Deepayan Chakrabarti, Deepak Agarwal, and Vanja Josifovski. Contextual advertising by combining relevance with click feedback. In *WWW 2008*. ACM, 2008.

[4] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97, 2005.

[5] Anisio Lacerda, Marco Cristo, Marcos Goncalves, Fan Weiguo, Nivio Ziviani, and Berthier Ribeiro-Neto. Learning to advertise. In *SIGIR 2006*, pages 549–556. ACM, 2006.

[6] Ling Li and Hsuan-Tien Lin. Ordinal regression by extended binary classification. In *Advances in Neural Information Processing Systems 19*, pages 865–872, 2007.

[7] Vanessa Murdock, Massimiliano Ciaramita, and Vassilis Plachouras. A noisy-channel approach to contextual advertising. In *ADKDD 2007*. ACM, 2007.

[8] Adwait Ratnaparkhi. A hidden class page-ad probability model for contextual advertising. In *Workshop on Targeting and Ranking for Online Advertising, at WWW 2008*, Beijing, China, April 2008.

[9] Berthier Ribeiro-Neto, Marco Cristo, Paulo Golgher, and Edleno Silva de Moura. Impedance coupling in content-targeted advertising. In *SIGIR 2005*. ACM, 2005.

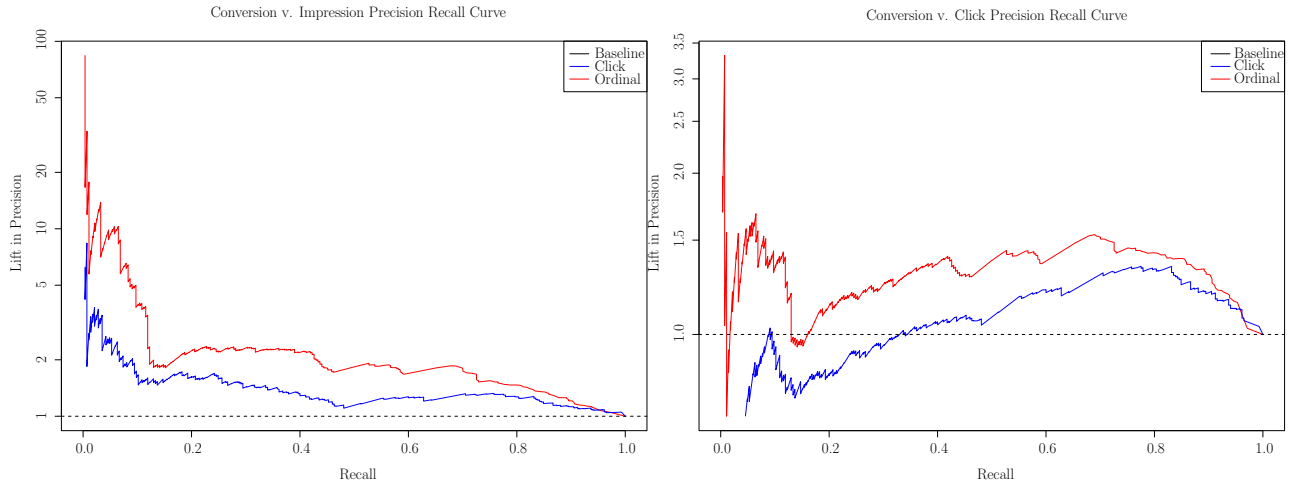

Figure 4: Precision and recall curves from online traffic comparing performance of predicting (a) conversions versus impressions and (b) conversions versus clicks. The impressions are ordered in decreasing order of model score $\hat{P}(C)$.

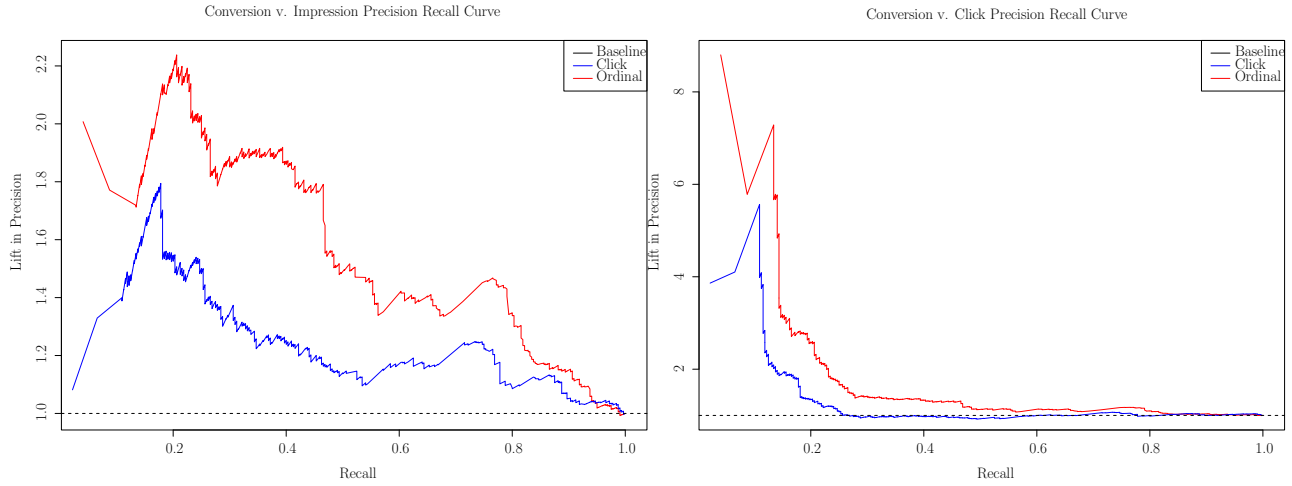

Figure 5: Precision and recall curves from online traffic comparing performance of predicting (a) conversions versus impressions and (b) conversions versus clicks. The impressions are ordered in decreasing order of ECPM estimated by the model–$\hat{P}(C)V(C)$.