Revenue Optimization with Relevance Constraint in Sponsored Search

Yunzhang Zhu[12] Gang Wang[1] Junli Yang[13] Dakan Wang[14] Jun Yan[1] Zheng Chen[1]
[1]Microsoft Resarch Asia, Beijing, China
[2]Department of Fundamental Science, Tsinghua University, Beijing, China
[3]Software Engineering Department, Nankai University, Tianjing, China
[4]Computer Science Department, Shanghai Jiaotong University, Shanghai, China
{v-yuazhu,gawa,v-junyan,v-dakwan, junyan, jianh, zhengc}@microsoft.com

ABSTRACT

Displaying sponsored ads alongside the search results is a key monetization strategy for search engine companies. Since users are more likely to click ads that are relevant to their query, it is crucial for search engine to deliver the right ads for the query and the order in which they are displayed. There are several works investigating on how to learn a ranking function to maximize the number of ad clicks. However, this ranking optimization problem is different from algorithmic search results ranking in that the ranking scheme must take received revenue into account in order to make more profit for the search engines. In this paper, we address a new optimization problem and aim to answer the question: how to construct a ranking model that can deliver high quality ads to the user as well as maximize search engine revenue? We introduce a novel tradeoff method from machine learning perspective, and through this method we have the privilege of choosing a tradeoff parameter to achieve highest relevance ranking or highest revenue ranking or the tradeoff between them. The algorithms are built upon the click-through log data with real ad clicks and impressions. The extensively experimental results verify that the proposed algorithm has the property that the search engine could choose a proper parameter to achieve high revenue(income) without losing to much relevance.

Categories and Subject Descriptors

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

General Terms

Algorithms, Design, Performance, Economics, Experimentation.

Keywords

Sponsored search, revenue optimization, ranking, machine learning.

1. INTRODUCTION

The practice of *sponsored search* advertising, where the paid advertisements appear alongside web search results, is now one of the largest sources of revenue for search engine companies. When a user types a query, search engine delivers a list of ads that are relevant to the query adjacent to or above the search results pages. Such ads are called sponsored ads. When a user clicks on a ad, he/she is taken to the landing page of the advertisement. Under the Pay per click (PPC) advertising model, such click generates a fixed amount of revenue to search engine, where the fixed amount is the bidding price of that ad (in this article we do not consider the effect of generalized second price auction on search engines' actually received revenue). Thus, the total revenue generated by a particular ad to search engine is the number of clicks multiplied by the cost per click (CPC). The CPC has the form of a bid and is established through generalized second price auction. In this paper, we restrict ourselves to CPC billing, which means search engine is paid every time when the ad is clicked by a user. All primarily search engines such as Google, Yahoo and Microsoft use this model.

When a user types a query, the search engine delivers several ads that have high possibilities of being clicked by that user. The search engine will receive more clicks if better matched ads are delivered. Thus, high accuracy of delivering the most preferred ads to each user will help search engine maximize the number of clicks. Previous works addressing this problem mainly focused on improving the ad matching relevance in sponsored search [10, 4, 2]. However, learning such a function to maximize the number of clicks is not exactly what the search engines wanted.

The ultimate goal of search engine is to find an optimal ranking scheme which can maximize the total revenue. Specifically, besides the number of clicks, an extremely important factor, which substantially influences the total revenue, is the bidding price of each ad. In [6], the author state why should take the advertisers' bidding prices into account when constructing the ranking function to maximizing the total revenue. Their work indicate that search engine would place the ads in sponsored results that not only are relevant to the user's query, but also have potential to increase more revenue. In this paper, we introduce a novel tradeoff methods from machine learning perspective, and with it you have the privilege of choosing a tradeoff parameter to achieve highest relevance ranking or highest revenue ranking or whatever between them. The algorithms are built upon the click-through log data with real ad clicks and impressions.

The remainder of the paper is organized as follows. In section 2 we formally introduce the revenue optimization problem in [6] and introduce the click-through log. Section 3 introduces a tradeoff approach to learning from click-through data for sponsored search.

Section 4 describes twelve basic features extracted for investigation. Section 5 displays the the experimental results and the last section conclude the paper.

2. RELATED WORKS

In recent years, it attracts more attentions to predict if an ad is likely to be clicked and how search engine produces an optimal list of ads. The related works on sponsored search advertisement can generally be categorized into three approaches. The first approach applied learning to rank algorithms to optimize the ad relevance [10, 11, 3], and their objective is to maximize the number of clicks. If the ranking function can accurately model the user click decision and reach to the relevance accuracy close to 100%, this approach also leads to the revenue maximization. However, as two situations mentioned above, the real case is more complicated. In order to take the revenue into consideration, the second approach combines revenue and relevance through either a form of linear combination [11] or multiplying the relevance score by the bidding price [2]. The methods under this approach are heuristic so that it is not easy to find the optimal parameters. Especially when the number of tuning parameters is many, the expense of heuristic parameter tuning becomes infeasible since the computational complexity increases exponentially as the number of parameters increases. The third approach decomposes the expected revenue we aim to maximize into two parts, i.e., CTR (click-through rate) and the bidding price. Several algorithms are proposed to estimate the CTR value for an ad with high precision (low variance)[5, 13, 12, 7]. The CTR value is an absolute measurements between a query and an ad. The assumption of this approach, as indicated in [7], is that there is an intrinsic relevance of an ad, which independent of the context of the user's query. Thus, the ads with high absolute CTR value many not be relevant to a particular user's query. In summary, the first approach only produces the results with high relevance accuracy. The second and third approaches use the two steps procedure to optimize revenue instead of relevance. Although they can generate better revenue, they always lose much relevance accuracy.

However, in the previous work [6], we solve these problems by constructing some promising algorithms which aim to maximizing the total revenue. In this paper, we further introduce a tradeoff parameter which can be adjusted by search engines according to their preference between relevance or revenue. So our algorithm are based on similar assumptions with [6]: (1) A click can only serve as an indication that an ad is more relevant than the ads that are not clicked in the same ranking list, but not as an absolute indication of the ad relevance. (2) To avoid the positional bias, we only choose the first three ads in the ad list alongside the search results from the click-through data.

3. REVENUE OPTIMIZATION PROBLEM

3.1 Click-through Data

Each record in the ad click-through data is the quadruple $\langle q, Ad_q(p), c_q(p), r_q(p)\rangle$, consisting of the query q, the ad $Ad_q(p)$ at the position p, the binary variable $c_q(p)$ encoding the click information, and the bidding price $r_q(p)$ for this ad. More specifically, when a user types in a particular query q, $p \in \mathbb{N}$ is the position of ad within the displaying ads list. $Ad_q(p)$ refers to the p^{th} ad in the ads list, $c_q(p) = 1$ if and only if $Ad_q(p)$ has been clicked by that user. A typical example is shown in Table 1. The column "*revenue*" indicates the bidding price of the corresponding ad. We use the click-through data as in this format throughout the paper. Several ad pairs are constructed for each ranking list. Table 1 displays the result of such ad pairs constructed from click-through data in Table 1. In the column "*label*", +1 indicates that the user prefers the first ad to the second one in the ad pair.

Table 1: Example of the Click-through Data

queries	*ads*	*pos*	*click*	*revenue*		*ads pair*	*label*
$query_1$	ad_1	1	✓	4	$\Longrightarrow$	(ad_1,ad_2)	+1
	ad_2	2		3			
	ad_3	3	✓	1		(ad_3,ad_2)	+1
$query_2$	ad_1	1		2	$\Longrightarrow$	(ad_1,ad_2)	−1
	ad_2	2	✓	5			
	ad_3	3		4		(ad_3,ad_2)	−1
$query_3$	ad_1	1		6	$\Longrightarrow$	(ad_1,ad_3)	−1
	ad_2	2		7		(ad_2,ad_3)	−1
	ad_3	3	✓	4			
$query_4$	ad_1	1		2	$\Longrightarrow$	(ad_1,ad_2)	−1
	ad_2	2	✓	9			
	ad_3	3		4		(ad_3,ad_2)	−1

3.2 Problem Formulation

Our aim is to maximize the expected revenue for search engine. Suppose Q is the set of total queries collected from search engine, then the available data is represented as:

$$\{\langle q, Ad_q(p), c_q(p), r_q(p)\rangle\}_{q\in Q} \quad (1)$$

Based on a fixed ranking function $R(\cdot)$, defined as

$$R(\cdot) : \{\langle q, Ad_q(p)\rangle\}_{p=1}^{n_q} \longrightarrow \{1, 2, \dots, n_q\}, \quad (2)$$

where n_q is the number of ads in the ads list corresponding to the query q, we can rank the ads lists for all queries in Q. Here we assume that there are no ties in the rank mapping, i.e., the ranking function $R(\cdot)$ is bijective. Thus for each query q, the ranking function $R(\cdot)$ is equivalent to a set of permutations $\sigma_q(\cdot)$ on $\{1, 2, \dots, n_q\}$ $(q \in Q)$ in the way that

$$R(\langle q, Ad_q(p)\rangle) = i \iff \sigma_q(p) = i, \quad (p, i = 1, \dots, n_q, \quad q \in Q). \quad (3)$$

According to the above denotations, we define two important scores: the *revenue score* and the *relevance score*.

Definition 1. Given the evaluation set $\{\langle q, Ad_q(p), c_q(p), r_q(p)\rangle\}_{q\in Q}$ and a fixed learned function $R(\cdot)$, the revenue score *#Revenue* is:

$$\#Revenue = \frac{\sum_{q\in Q} c_q(\sigma_q^{-1}(1)) r_q(\sigma_q^{-1}(1))}{\sum_{q\in Q} \max_{p\in\{1,\dots,n_q\}}\{r_q(p)c_q(p)\}}. \quad (4)$$

$\sigma_q^{-1}(1)$ returns the original position in the ads list whose ad is ranked on the first position by the ranking model. Thus, $c_q(\sigma_q^{-1}(1)) r_q(\sigma_q^{-1}(1)) = r_q(\sigma_q^{-1}(1))$ is the received revenue for search engine. From this definition, if the ad ranked on the first position is actually clicked by the user, then the bidding from the clicked ad is assumed to be received. Otherwise, if a clicked ad is not ranked on the first position, we lose the revenue that ad generates (i.e., $c_q(\sigma_q^{-1}(1)) r_q(\sigma_q^{-1}(1)) = 0$, no revenue received). The revenue score is the revenue we actually receive divided by the largest possible revenue ($\sum_{q\in Q} \max_{p\in\{1,\dots,n_q\}}\{r_q(p)c_q(p)\}$) we could receive. Therefore, in order to maximize the search engine revenue, the revenue score acts as the criteria of both optimizing and evaluating the ranking model.

Definition 2. Given the evaluation set $\{\langle q, Ad_q(p), c_q(p), r_q(p)\rangle\}_{q\in Q}$ and a fixed learned function $R(.)$, the relevance score *#Relevance* is:

$$\#Relevance = \frac{\sum_{q\in Q} c_q(\sigma_q^{-1}(1))}{\sum_{q\in Q} \max_{p\in\{1,\dots,n_q\}}\{c_q(p)\}} = \frac{\sum_{q\in Q} c_q(\sigma_q^{-1}(1))}{\#Q}. \quad (5)$$

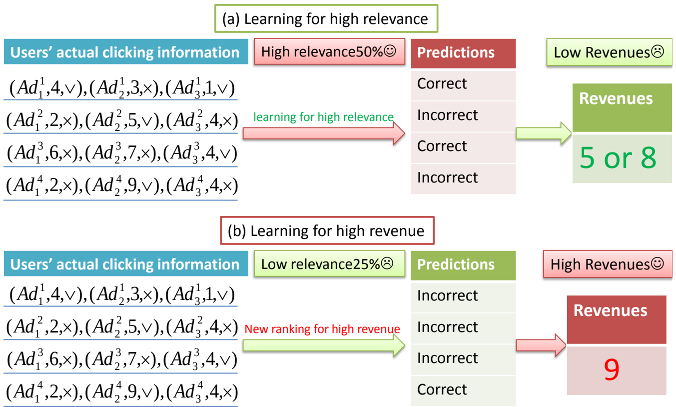

Figure 1: Two examples of the ranking function (a) The ranking with high relevance but low revenue; (b) The ranking with lower relevance but higher revenue.

$\sum_{q \in Q} c_q(\sigma_q^{-1}(1))$ is the number of the first ranked ad that is also clicked by users. This number divided by the total number of queries is the relevance score, which is similar to the ad ranking accuracy used in [4].

We use the data in Table 1 as an example, and it is clear that the largest possible revenue we could receive will be $4+5+4+9 = 22$. If a ranking model produces the first position ad as ad3, ad2, ad1 and and2 respectively, then the totally received revenue is $1+5+0+9 = 15$ and the final revenue score is $\frac{15}{22}$. Accordingly, the relevance score is $\frac{1+1+0+1}{4} = 0.75$.

After introducing these definitions, the revenue optimization can be formulated as two major problems: (1) What ranking strategy we develop in order to maximize the revenue score as well as preserve the relevance score; (2) Which features that best captures the user' click behavior and achieves high revenue score.

The revenue score reflects the purpose of maximizing search engine revenue, while the revenue score only reflects the users' relative preference. A learned ranking function may have higher relevance score but still lead to lower revenue. We expect that an appropriate ranking function which takes revenue into account can produce higher revenue even if it sacrifices the correctness of prediction. Figure 1 illustrates an example of this case. In Figure (b), the ranking function reaches to 25% relevance score, but the received revenue is 9. However, in Figure 1(a), the ranking function reaches to 50% relevance score, but the revenue is only 5 or 8. It seems that the revenue optimization contradicts the relevance optimization. However, it is not always true. In Figure 1(a), the ranking function reaches to 50% accuracy if either Ad_1^1 or Ad_3^1 is predicted as the clicked ad. If Ad_1^1 is ranked first, the received revenue is 5. On the other hand, if Ad_3^1 is on the first, better revenue is received, i.e., 8. Thus, we can receive 3 more bidding revenue but not lose the relevance. In this paper, one important issue in developing new ranking model is to maximize the revenue score and simultaneously avoid losing the relevance score.

4. LEARNING MODELS

We first introduce a learning to rank algorithm similar to [9] which aims to achieve the highest relevance score. Then give a brief review of our work [6] which aims to maximizing the total revenue for the search engines. And finally we'll combine these two objective function to a new objective function through a parameter μ. Optimizing this objective function lead to our new tradeoff model.

4.1 Learning to Rank Methods

Throughout this section, we adopt the pairwise learning paradigm. Suppose for each query-ad pair $(q, Ad_q(p))$, it corresponds to a feature vector $\vec{x}_q(p) \in \mathbb{R}^k$. Pairwise learning methods essentially encode the users' preference information (the click information in this paper) through a function $f(\cdot, \vec{w}) : \mathbb{R}^k \longrightarrow \mathbb{R}$ as:

$$f(\vec{x}_q(p^{(1)}), \vec{w}) > f(\vec{x}_q(p^{(2)}), \vec{w}) \iff (c_q(p^{(1)}) = 1) \wedge (c_q(p^{(2)}) = 0) \quad (6)$$

where $\vec{w} \in \mathbb{R}^k$, $p^{(1)}, p^{(2)} \in \{1, \ldots, n_q\}$. If f is a linear transformation on $\mathbb{R}^k$, i.e., $f(\vec{x}, \vec{w}) = \langle \vec{w}, \vec{x} \rangle$, then (6) is equivalent to:

$$\langle \vec{w}, \vec{x}_q(p^{(1)}) - \vec{x}_q(p^{(2)}) \rangle > 0 \iff (c_q(p^{(1)}) = 1) \wedge (c_q(p^{(2)}) = 0) \quad (7)$$

Based on (7), pairwise learning casts the ranking problem as a classification task, where the samples are from:

$$S = \{\vec{x}_q(p^{(1)}) - \vec{x}_q(p^{(2)}), c_q(p^{(1)}) - c_q(p^{(2)})\}$$
$$p \in Q, \quad p^{(1)}, p^{(2)} = 1, \ldots, n_q, \quad c_q(p^{(1)}) \neq c_q(p^{(2)}) \quad (8)$$

where $c_q(p^{(1)}) - c_q(p^{(2)}) = 1$ or -1 corresponding to labels. To simplify the notation, we use the following set $\tilde{S}$ to represent S

$$\tilde{S} = \{\vec{x}_i^{(1)} - \vec{x}_i^{(2)}, c_i\}_{i=1}^N \quad (9)$$

where $c_i \in \{+1, -1\}$ is the class label and N is the number of pairs in (8). We introduce two pairwise learning methods which use the same procedure to optimize a solution $\vec{w}^*$. However, two approaches of deriving the final ranking function from the learned function $f(\vec{w}^*, \vec{x})$ are different.

Learning to rank algorithms have been investigated extensively, and we only give a brief description. In order to be consistent to the second algorithm, we use the logistic regression algorithm to learn the optimal $\vec{w}$. The learning problem is to solve the following convex optimization:

$$\min_{\vec{w}} \{\sum_{i=1}^{N} \log[1 + \exp(-c_i \langle \vec{w}, \vec{x}_i^{(1)} - \vec{x}_i^{(2)} \rangle)] + \lambda \|\vec{w}\|^2\} \quad (10)$$

where λ is the regularization parameter whose value is specified in advance. We call this method RankLogistic. (Compared to RankSVM [8], the only difference is that it uses the logistic regression algorithm). Suppose that $\vec{w}^*_{log}$ is the optimal weights. Accordingly, we arrange the samples $\vec{x}_q(p) \forall p$ in decreasing order as

$$f(\vec{x}_q(l_q^{(1)}), \vec{w}^*_{log}) > f(\vec{x}_q(l_q^{(2)}), \vec{w}^*_{log}) > \ldots > f(\vec{x}_q(l_q^{(n_q)}), \vec{w}^*_{log}) \quad (11)$$

Our ranking functions σ_q are

$$\sigma_q^{-1}(p) = l_q^{(p)}, \quad p = 1, \ldots, n_q, \ q \in Q \quad (12)$$

which determines our final ranking function $R(\cdot)$ according to (3).

4.2 Revenue Direct-Optimization

Adopting the same terminology in [?], [6] formulate the revenue optimization problem as maximizing the empirical revenue $Rev(\vec{w})$

$$Rev(\vec{w}) = \sum_{q \in Q} \sum_{p=1}^{n_q} r_q(p) c_q(p) \mathbb{I}_{\{\min_{i \neq p}\{f(\vec{w}, \vec{x}_q(p)) - f(\vec{w}, \vec{x}_q(i))\} > 0\}} \quad (13)$$

Then they assume $f(\vec{w}, \vec{x})$ is linear in w and x and approximate this loss by a convex one, and finally aims to maximize the following convex programming problem:

$$L(\vec{w}) = \lambda\|\vec{w}\|^2 + \sum_{q\in Q}\sum_{p=1}^{n_q} r_q(p)c_q(p)\sum_{i\neq p}\log\{1+\exp[-(\langle\vec{w},\vec{x}_q(p)\rangle - \langle\vec{w},\vec{x}_q(i)\rangle)]\} \quad (14)$$

Compared to the logistic loss function, the above loss function has a set of weights $r_i^{(1)}$, which corresponds to the bidding of those clicked ads. Thus if all of them equal to 1, then the loss in (**??**) is exactly the same as the loss in (10). In this case, the loss (10) treats all the potential errors of training samples equally important. However, the new loss in (**??**) incorporates the bidding as the weight into the loss function thus intensifies those ads with larger bidding. This also enables us to use the logistic regression package after revising the initial training data set.

4.3 Trade-off learning models

The above two models have totally two different objective. Specifically, one is aiming to achieve highest relevance for the purpose of maximizing the receiving clicks. The other is aiming to maximize the total revenue. In this section we do a convex combination of this two objective function to let the combination parameter for your adjustment.

In this section, we aims to construct a relevance-revenue trade-off model. This can be achieved by optimizing the relevance objective with a revenue constrain, say the total revenue is no larger than a predetermined value l. This is equivalent to maximize

$$\lambda\|\vec{w}\|^2 + \sum_{q\in Q}\sum_{p=1}^{n_q} c_q(p)\sum_{i\neq p}\log\{1+\exp[-\langle\vec{w},\vec{x}_q(p)-\vec{x}_q(i)\rangle]\} \quad (15)$$

under the following constrain:

$$\sum_{q\in Q}\sum_{p=1}^{n_q} r_q(p))c_q(p)\sum_{i\neq p}\log\{1+\exp[-\langle\vec{w},\vec{x}_q(p)-\vec{x}_q(i)\rangle]\} > l \quad (16)$$

And this constrained optimization problem can be transformed to its equivalent form:

$$L(\vec{w}) = \lambda\|\vec{w}\|^2 + \sum_{q\in Q}\sum_{p=1}^{n_q}(1+\mu r_q(p))c_q(p)\sum_{i\neq p}\log\{1+\exp[-\langle\vec{w},\vec{x}_q(p)-\vec{x}_q(i)\rangle]\} \quad (17)$$

where μ is depend solely on l and serve as to strike a balance between revenue and relevance.

The trade-off objective function becomes: The gradient of $L(\vec{w})$ is

$$\nabla L(\vec{w}) = 2\lambda\vec{w} + \sum_{q\in Q}\sum_{p=1}^{n_q}(1+\mu r_q(p))c_q(p)\sum_{i\neq p}\frac{e^{-\langle\vec{w},\vec{x}_q(p)-\vec{x}_q(i)\rangle}(\vec{x}_q(p)-\vec{x}_q(i))}{1+e^{-\langle\vec{w},\vec{x}_q(p)-\vec{x}_q(i)\rangle}} \quad (18)$$

Using this objective and the its gradient, we resort to quasi-Newton methods to optimize this objective. And the resulting optimal parameter $\vec{w}(\mu)$ as a function of μ give us a series of models - they put different importance on revenue and relevance with different μ as their argument.

5. FEATURES

We totally extract 12 features to represent each query and ad pair (see Table 2). These features can be separated into four categories: the low-level relevance features, the high-level relevance features, the CTR features and some other features such as the bidding price and the match type between the input query and the advertiser's bidding keywords. There are four match type values indicating exact match, broad match, phrase match and smart match.

The low-level relevance features include TF(term frequency), TF * IDF(inverse document frequency) [1], and edit distance. The term frequency is the summation of the number of times each query term appears in the ad title and description. Inverse document frequency is a balance factor to evaluate how important a term is to an ad in the advertisement corpus. We compute the edit distances between each query term and the displayed url and take the largest as the edit distance value. The high-level relevance features include the outputs from the Okapi BM25 [14] and LMIR (language models for information retrieval) [15] ranking algorithms which measure the relevance between query and either ad title or ad description. In particular for LMIR, there are several smoothing methods such as Dirichlet, Jelinek-Mercer, absolute discounting and etc. We adopt the Dirichlet smoothing in this paper.

Ad CTR is a statistical measurement whose value is computed by the number of ad clicks divided by the total number of ad impressions. If one ad's CTR is larger than another ad's CTR, this ad has higher probability to be clicked. The ad CTR value is an absolute indication of the ad popularity, which is independent to the input query. The estimation of ad CTR often has high variance, especially for new ads whose number of ad impressions is small. Since new ads have little chance to be seen by users, their CTR values are likely to zero. However, these new ads cannot be judged as unpopular ads. As a result, we cannot treat new ads with little number of ad impressions the same as old ads whose number of ad impressions is large. The details on how to handle the old ads and new ads separately are described in the experimental part. Similarly, we calculate the CTR value for advertiser account and advertising campaign. Each advertiser account corresponds to multiple advertising campaigns and each advertising campaign includes multiple ads.

6. EXPERIMENTAL RESULTS

6.1 Data Set

We collect three months click-through log data with ad clicks and impressions. After randomly sampling across the overall log data and filtering out the query whose returned ads number is less than 3, we totally have 10 million queries and each query corresponds to a list of first three ads in the ad list alongside the search results. We use 80% records in the training and the left 20% records in the testing. Since the collected data samples are adequate enough in terms of 12 features, cross validation is not applied.

Table 2: A list of extracted features in the representation of each query and ad pair.

Relevance	low level	TF
		TF*IDF
		edit distance
	high level	BM25 of title
		BM25 of description
		LMIR of title
		LMIR of description
CTR		ad CTR
		compaign CTR
		account CTR
Others		bidding price
		match type

Table 3: Revenue score and relevance score comparisons

Feature Set	9 features		12 features	
	Relevance Score	*Revenue Score*	*Relevance Score*	*Revenue Score*
Learning to rank (Baseline 1)	**0.5461**	0.4092	**0.6154**	0.4935
Revenue Direct-Optimization (RDO)	0.4891	**0.4899**	0.5899	**0.5312**

The estimation of the CTR features is only reliable for "old" ads with large impression number, hence, we split the experimental evaluation into two parts. In the first part, we exclude the CTR features and use other 9 features as the feature representation. In this case, all of the training data are used to learn a ranking model and all of the testing data are used to evaluate the revenue and relevance score. In the second part, in order to make use of the CTR features, we only select those ads with at least 100 impressions, where the impression number is counted in the training data. We denote these ads set as S_{CTR}. For a query one of whose returned ad is not in S_{CTR}, this query and all its returned ads will not be included. After choosing the "old" ads from the training and testing data, we can build and evaluate a ranking model based on 12 features.

6.2 Algorithm Comparisons

In the experiments, we first consider the first two methods. The first one is learning to rank introduced in Section **??**. The second one is revenue direct optimization. We use the same ads to train the ranking model in both 9 feature and 12 feature representations. Some previous works on the CTR prediction have developed new features which can boost the estimation performance. However, the CTR prediction is still a challenge work and this paper does not explore advanced features on this point. We only use 12 basic features in order to be consistent to other methods.

Table 3 illustrates the revenue and relevance score comparisons among these two methods in two different feature representations. Since learning to rank directly optimizes the relevance, it reaches highest relevance scores in both 9 features and 12 features. However, the revenue scores from learning to rank are lower. On the contrary, the revenue direct-optimization (RDO) approach always achieves the highest revenue scores, which has relative 19.7% revenue improvement in 9 features and relative 7.3% revenue improvement in 12 features compared with learning to rank. As indicated in (**??**), RDO directly takes more emphasis on the bidding price in the ranking function. Parameter tuning procedure were stated in detail in [6].

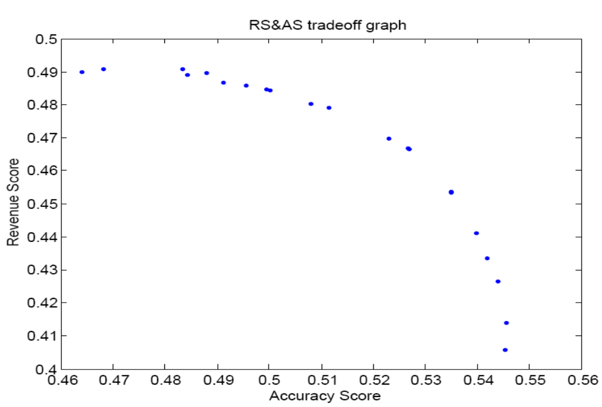

Figure 2: Revenue and relevance trade-off graph in 9 features data

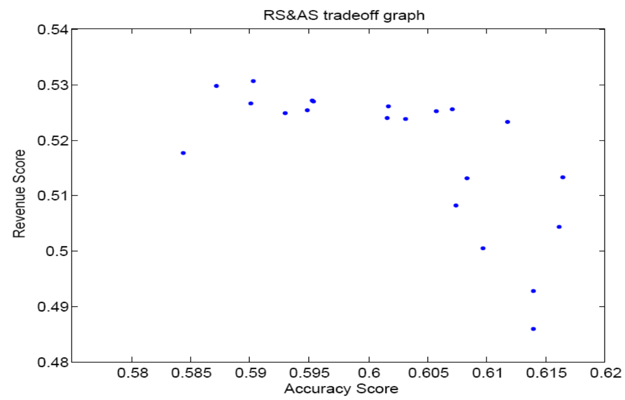

Figure 3: Revenue and relevance trade-off graph in 12 features data

The newly proposed trade-off methods aims to introducing a trade-off parameter to adjust any degree of importance you may want to put on relevance or revenue component. Figure [[**?**]] and Figure [[**?**]] shows the trade-off graph when the trade-off parameter $\lambda = \{0, 0.05, 0.1, ..., 0.95, 1\}$. We can see the for most of the time, the trade-off larger λ gives us larger revenue score and smaller λ gives us larger relevance score, as we expected. Also, we observe that when we sacrifice the relevance score a little bit, the revenue score might increase substantially and vise versa. This phenomenon suggest that search engines should choose a optimal value in (0, 1) for λ rather than go to two extremes. A proper choice may deliver high relevant ads to search users and still lead to high revenue income for search engines.

7. CONCLUSIONS

In this paper we investigate a novel approach to learning and evaluating sponsored search ranking systems using real clickthrough log data. By optimizing the corresponding objective functions, we may reach an appropriate balance between achieving highest revenue and also high relevance. Previous works investigate on how to estimate the relevance scores for each sponsored ads. Giving high ranking to those ads with high relevance score will fulfill search engine user's satisfaction, thus has much potential to attract more user clicks which may ultimately generate business transactions. However, under Pay Per Click system, optimizing the sponsored search by delivering the most relevant ads is not equivalent to maximizing search engine's total revenue. In our previous paper, our proposed novel method, revenue post-optimization , aims to maximize the total revenue. This approach is similar to previous works [5, 7, 12, 13], since they all try to directly combine some relevance score and the revenue score in a two way procedure that might maximize the revenue. However, previous works using some heuristic combination cannot guarantee the solution is optimal in any sense. Instead, through estimating a set of probabilities to the ads in the same list, the revenue post-optimization method is optimal in decision-making when the aim is to maximize the revenue.

However, all such direct relevance and revenue maximization approaches have the problem of losing the ads quality in high ranking positions. We then propose another trade-off ranking scheme which not only can achieve the high revenue but also avoid losing the quality of returned ads. And we observe that when we sacrifice the relevance score a little bit, the revenue score might increase substantially and vise versa. This phenomenon suggest that search engines should choose a optimal value in $(0, 1)$ for λ rather than go to two extremes. A proper choice may deliver high relevant ads to search users and still lead to high revenue income for search engines. In our future works, we will continue the revenue optimization works from the direction of exploring more features. As suggested in this paper, more CTR and relevance features should be investigated to further improve the revenue score. And also, we will further explore how to choose the trade-off parameter λ, or more ambitiously, how we could construct an ideal criterion which can tell us what kind of trade-off is optimal to the search engines in more longer term.

8. REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[2] A.Z. Broder, P. Ciccolo, M. Fontoura, E. Gabrilovich, V. Josifovski, and L. Riedel. Search advertising using web relevance feedback. In *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM-08)*, 2008.

[3] D. Chakrabarti, D. Agarwal, and Vanja Josifovski. Contextual advertising by combining relevance with click feedback. In *Proceeding of the 17th international conference on World Wide Web (WWW-08)*, 2008.

[4] M. Ciaramita, V. Murdock, and V. Plachouras. Online learning from click data for sponsored search. In *Proceedings of the 17th international conference on World Wide Web (WWW-08)*, 2008.

[5] K. Dembczynski, W. Kotlowski, and D. Weiss. Predicting ads' click-through rate with decision rules. In *Workshop on Targeting and Ranking in Online Advertising 2008*, 2008.

[6] Y. Zhu et al. Optimizing search engine revenue in sponsored search. In *SIGIR*, 2009.

[7] J. Feng, H. Bhargava, and D. Pennock. Implementing sponsored search in web search engines: Computational evaluation of alternative mechanisms. *INFORMS Journal on Computing*, 19:137–148, 2005.

[8] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, 2000.

[9] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD-02)*, 2002.

[10] A. Lacerda, M. Cristo, M.A. Goncalves, W. Fan, N. Ziviani, and B. Ribeiro-Neto. Learning to advertise. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR-06)*, 2006.

[11] F. Radlinski, A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, and Lance Riedel. Optimizing relevance and revenue in ad search: a query substitution approach. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR-08)*, 2008.

[12] M. Regelson and D.C. Fain. Predicting click-through rate using keyword clusters. In *Proceedings of the 2nd Workshop on Sponsored Search Auctions*, 2006.

[13] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web (WWW-07)*, 2007.

[14] S. E. Robertson. Overview of the okapi projects. *Journal of Documentation*, 53:3–7, 1997.

[15] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR-01)*, 2001.