Cui Q, Bai FS, Gao B *et al.* Global optimization for advertisement selection in sponsored search. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 30(2): 295–310 Mar. 2015. DOI 10.1007/s11390-015-1523-4

Global Optimization for Advertisement Selection in Sponsored Search

Qing Cui [1] (崔 卿), Feng-Shan Bai [1] (白峰杉), Bin Gao [2] (高 斌), *Member, ACM, IEEE*, and Tie-Yan Liu [2] (刘铁岩), *Senior Member, CCF, ACM, IEEE*

[1] *Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China*
[2] *Microsoft Research Asia, Beijing 100080, China*

E-mail: cuiq12@mails.tsinghua.edu.cn; fbai@math.tsinghua.edu.cn; {bingao, tyliu}@microsoft.com

Received January 7, 2014; revised November 24, 2014.

Abstract Advertisement (ad) selection plays an important role in sponsored search, since it is an upstream component and will heavily influence the effectiveness of the subsequent auction mechanism. However, most existing ad selection methods regard ad selection as a relatively independent module, and only consider the literal or semantic matching between queries and keywords during the ad selection process. In this paper, we argue that this approach is not globally optimal. Our proposal is to formulate ad selection as such an optimization problem that the selected ads can work together with downstream components (e.g., the auction mechanism) to achieve the maximization of user clicks, advertiser social welfare, and search engine revenue (we call the combination of these objective functions as the marketplace objective for ease of reference). To this end, we 1) extract a bunch of features to represent each pair of query and keyword, and 2) train a machine learning model that maps the features to a binary variable indicating whether the keyword is selected or not, by maximizing the aforementioned marketplace objective. This formalization seems quite natural; however, it is technically difficult because the marketplace objective is non-convex, discontinuous, and indifferentiable regarding the model parameter due to the ranking and second-price rules in the auction mechanism. To tackle the challenge, we propose a probabilistic approximation of the marketplace objective, which is smooth and can be effectively optimized by conventional optimization techniques. We test the ad selection model learned with our proposed method using the sponsored search log from a commercial search engine. The experimental results show that our method can significantly outperform several ad selection algorithms on all the metrics under investigation.

Keywords advertisement selection, sponsored search, probability model

1 Introduction

Sponsored search is the main monetization channel for the commercial search engines. In sponsored search, the paid advertisements (ads) are presented to users along with the organic search results. First, these ads bring values to the users, in terms of product information, significant discount, etc. Second, these ads certainly bring values to the advertisers, since their marketing campaigns reach the target audience. Third, these ads also bring values to the search engine, since the search engine will gain revenue once the ads are clicked by the users.

Putting it simple, today's sponsored search systems basically work in the following manner. First of all, advertisers are required to open accounts in the sponsored search system, create ad campaigns under each account, and upload a group of ads (together with keywords and bids) into each campaign. Given a query submitted by a user, the sponsored search system selects a set of keywords by using an ad selection algorithm. Then all the ads that bid on the selected keywords will be fed into the downstream modules in the sponsored search system. Next these ads will go through an auction process. The auction mechanism determines which of these ads will be shown to users (according to a ranking rule) and how much they need to pay if they are clicked by the user (according to a pricing rule). The generalized second-price auction (GSP)[1-2] is one of the most popularly used auction mechanisms. With GSP, the ads are

Regular Paper
Special Section on Applications and Industry
The work was done when the first author was an intern at Microsoft Research Asia.

ranked in the descending order of the rank score, which is defined as the product of the ad quality score and the bid. Quality score is an estimation of how relevant the ad, keywords, and landing page are to the search engine user who is seeing the ad. Sometimes, people use the predicted ad click probability as the quality score. However, it is not a necessity, though both scores can indicate the *quality* of the ad. In some research and industry practice, the predicted ad click probability and the ad quality score are regarded as two signals. We will take the latter setting in this paper. In the ranked ad list, the top-ranked ads are shown to the web user. If the user clicks on some of the ads, the owner of the clicked ads will be charged according to a second-price rule, i.e., the payment is the minimum bid for the ad to win its current rank position. The process is shown in Fig.1.

As can be seen from the above description, the ad selection algorithm resides in the upstream of the sponsored search system. It determines which ads (more accurately, which keywords bidden by the ads) will be fed into the auction mechanism. Therefore ad selection plays a very important role. Garbage in, then garbage out. If the ad selection result is bad, it is almost impossible for the sponsored search system to deliver relevant ads to users, help the right advertisers to achieve their campaign goals, and help the search engine gain desired revenue.

There has been a rich literature on ad selection. We group the existing ad selection algorithms into two categories. The first category of methods[3-4] mainly relies on the relevance between queries and keywords. They often expand the queries and/or keywords using additional text streams like organic search results, ad copies, and landing pages, when computing the relevance score. The second category of methods[5-7] performs ad selection based on the semantic relationship between queries and keywords. They usually use historical clicks on the ads to mine the semantic relationship. It is clear that both categories of methods regard ad selection as a relatively independent module. In other words, they define a local criterion for ad selection, without considering the impact of ad selection on the auction mechanism in the downstream of the sponsored search system. In this regard, we argue that these approaches are not *globally* optimal.

In our opinion, a better solution to ad selection is to explicitly consider its global impact, i.e., with the ad selection results, whether the overall sponsored search system (including the auction mechanism) can achieve the maximization of user clicks, advertiser social welfare (defined as the expectation of the realized advertiser values), and search engine revenue. We call the combination of these objective functions as marketplace objective for simplicity.

With this global view, what we should do is to formulate ad selection as an optimization problem. In the problem, the objective function is the aforementioned marketplace objective. The auction mechanism (including its sub-component, the quality score computation algorithm, and/or the ad click prediction algorithm) is regarded as known and fixed, and will be used in the calculation of the marketplace objective, given a particular set of ads selected by the ad selection algorithm. The ad selection algorithm combines a number of features extracted from each query-keyword pair and generates a binary value, indicating whether an ad is selected or not. By solving this optimization problem over historical sponsored search logs, we will be able to learn the optimal combination coefficients (in other words, the optimal model parameters) in the ad selection algorithm, and use them to perform ad selection in the future.

We would like to state the following two remarks

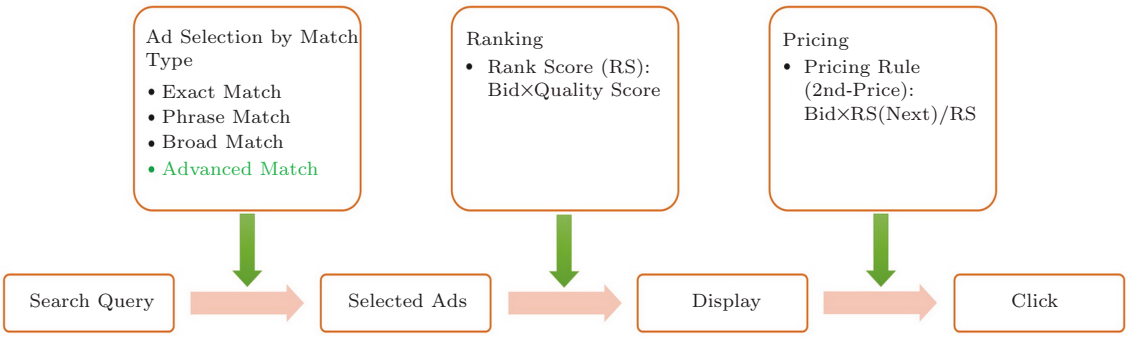


Fig.1. Sponsored search system.

about the proposed formulation. 1) The proposed method only affects the ad selection procedure and does not change the auction mechanism at all. In practice, this method can be used for advanced match which already exists in the sponsored search system as a black box, and thus it will cause little impact on the advertisers' strategy in an auction. 2) The phrase "global optimization" in the paper refers to the global view of ad selection together with the downstream components. It does not imply a global maximum in the optimization problem.

The above formalization seems quite natural; however, it is technically challenging because the marketplace objective is non-convex, discontinuous, and indifferentiable with respect to the parameters in the ad selection algorithm due to the ranking and second-price rules in the auction mechanism. To tackle the challenge, we propose a probabilistic approximation of the marketplace objective, which is smooth and can be effectively optimized by conventional optimization techniques. In particular, we treat ranking scores as random variables rather than deterministic values. With the distribution of the rank scores, we can compute the probability of an ad being ranked at any given position, and the expected pay per click according to the second-price rules. Besides, we also use some techniques to approximate the sign function and the maximum function in the marketplace objective. By doing so, all the discontinuous and indifferentiable components in the marketplace objective function are approximated to be differentiable, and then the optimization problem can be solved by conventional optimization techniques, such as the gradient method. We evaluate the proposed method in a sponsored search system. The experimental results show that our method outperforms several ad selection methods in terms of several widely used metrics.

To sum up, the contributions of our work are listed as below.

• As far as we know, this is the first ad selection method that considers optimizing the marketplace objective of the entire sponsored search system.

• We propose a probability method to smooth the marketplace objectives for ease of optimization. This provides a good reference for solving the complex optimization problems in sponsored search due to the existence of the generalized second-price auctions.

The rest of the paper is organized as the following. We give a literature review on the related work in Section 2. In Section 3, we introduce the proposed ad selection framework by optimizing the marketplace objective. In Section 4, we describe the probability method to approximate the discontinuous and indifferentiable objective functions. In Section 5, we discuss the efficient solution of the proposed model. In Section 6, we present the experimental results to show the effectiveness of the proposed ad selection framework. In the end, we conclude the paper and suggest the future research direction in Section 7.

2 Related Work

Existing work on ad selection can be organized in two categories: one is based on the relevance matching among queries and ads/keywords, and the other is based on mining the relationship among queries and ads/keywords from the historical ad click data.

The relevance-based methods often suffer from the short text streams, i.e., the text lengths of queries, keywords, and ad copies are usually very short. Therefore, many of these methods are focused on expanding the text streams of both queries and ads/keywords. Broder *et al.*[3] enriched both queries and ads with additional knowledge features. They used the organic search results to create relevant documents for the query. As query expansion can hardly be done in a real-time system, Broder *et al.*[5] proposed another approach of matching the ads against rare queries that can be accomplished online. In this method, they built expanded query representations from the preprocessed related queries. Choi *et al.*[4] explored the usage of the landing pages to expand the text stream of ads. Wang *et al.*[8] proposed an efficient ad search solution that uses a block-based index to tackle the issues associated with query expansion. The block-based index was employed in a retrieval system to return the top relevant ads.

The other category of methods uses the historical ad click information to mine the relationship among queries and ads/keywords. Antonellis *et al.*[9] built a click graph from the historical user queries and the corresponding ad clicks, and then proposed a new schema of Simrank++ to suggest the keywords for ad selection. Fuxman *et al.*[6] conducted the keyword suggestion by making use of the query logs of the search engine. They built a bipartite graph between queries and webpages, and carried out the propagation of the concept tags on the bipartite graph to make the keyword suggestion. In the work of Hillard *et al.*[7], they introduced a machine learning approach based on the translation models to predict the ad relevance, which can help select more relevant ads for the sponsored search system.

The above methods regard ad selection as a separated module of query-keyword matching. They do not consider the marketplace objective from all the players in the sponsored search system.

3 Marketplace Objective

As mentioned in the introduction, we propose explicitly optimizing the marketplace objective when learning the ad selection algorithm. In this section, we will discuss how we define the marketplace objective, and how we compute it given the ad selection algorithm, the click prediction algorithm, and the auction mechanism.

3.1 Preliminaries

To better illustrate the marketplace objective, we first give some preliminary notations. Let $\Omega = (Q, K, A)$ be the object space in sponsored search, where Q is the set of input queries submitted by the users, K is the set of keywords given by the advertisers, and A is the set of ads composed by the advertisers. Usually, an advertiser can bid on several keywords $k_i \in K$ for an ad $a_j \in A$. Suppose the sizes of K and A are M and N respectively, then we can use the following $N \times M$ dimensional matrix $\boldsymbol{B} = \{b_{ji}\}$ to represent advertisers' selected keywords, in which

$$b_{ji} = \begin{cases} 1, & \text{if } a_j \text{ bids } k_i, \\ 0, & \text{otherwise.} \end{cases}$$

If $b_{ji} = 1$, there will be a non-zero value $\bar{v}(j, i)$ indicating the bid of ad a_j for keyword k_i.

With these preliminary notations, we will go through the major components in the sponsored search systems. In the meanwhile, we will encounter more notations. For the sake of clarity and for ease of reference, we list the major notations in Table 1.

Table 1. Notations

Notation	Explanation
$\Omega = (Q, K, A)$	Ad space with the query set Q, keyword set K, and ad set A
q	Query
a_j	Ad, $j = 1, \cdots, N$
k_i	Keyword, $i = 1, \cdots, M$
r	Ad position, $r = 0, 1, \cdots, m-1$
$\boldsymbol{\eta}$	Indicator vector for keywords. $\eta_i = 1$, if k_i is selected; otherwise, $\eta_i = 0$.
$\boldsymbol{\theta}$	Indicator vector for ads. $\theta_j = 1$, if a_j is selected; otherwise, $\theta_j = 0$.
$c(\cdot)$	Ad click prediction function
$h(\cdot)$	Ad quality score function
$v(\cdot)$	Highest bid of an ad for a query
$g(\cdot)$	Ad ranking function, $g(\cdot) = h(\cdot)v(\cdot)$
$R(\cdot)$	Marketplace objective
$\zeta_\theta(\cdot)$	Permutation function on the subset A_q
$\zeta_\theta{}^{1}(r)$	Ad ranked at position r by permutation ζ
$D(r)$	Discount at position r
$\varsigma(\cdot)$	Sigmoid function
$a_i \succ a_j$	Ad a_i is ranked higher than a_j in an auction
$j \succ_1 i$	Ad a_j is ranked just one position higher than a_i in an auction

3.2 Ad Selection

In this subsection, we consider the ad selection algorithm. This algorithm takes a pair of query and ad, and outputs a binary variable indicating whether the ad is selected or not. In practice, this is done in two steps. First, the algorithm determines whether a keyword is selected or not, given the query. Second, it employs the matrix $\boldsymbol{B}$ to obtain the ads that should be selected. The details are given as follows.

For a pair of query q and keyword k_i, one extracts a group of features $\boldsymbol{x}_i^{(q)} = (x_{i1}^{(q)}, \cdots, x_{in}^{(q)})$, where n is the number of features. Suppose we have a linear function to combine these features, i.e.,

$$f(\boldsymbol{\omega}, \boldsymbol{x}_i^{(q)}) = \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x}_i^{(q)},$$

where $\boldsymbol{\omega}$ is an n-dimension parameter vector. In real applications, usually complex non-linear models are used. Given that our goal is to demonstrate the idea of global optimization, without loss of generality, we consider the linear model. The output of the linear function is converted to a score indicating how likely k_i should be selected given the query. As a common practice, we use a sigmoid function $\varsigma(\cdot)$[1] to compute this score. By comparing the score with a threshold, we can get a binary value η_i indicating whether the keyword k_i is selected or not, i.e.,

$$\eta_i = \mathrm{sgn}(\varsigma(\boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x}_i^{(q)}) - t), \tag{1}$$

where $\mathrm{sgn}(\cdot)$ is the sign function and t is the threshold.

We use vector $\boldsymbol{\eta}$ to represent the selection results for all the keywords, whose element η_i is the binary indicator for keyword k_i.

Given the selection results for the keywords, we need to retrieve all the related ads. Here we use vector $\boldsymbol{\theta}$ to denote the selection results for ads, whose element θ_j

[1] Wikipedia. Sigmoid function. http://en.wikipedia.org/w/index.php?title=Sigmoid_function, Feb. 2013.

indicates whether ad $a_j \in A$ is selected. $\boldsymbol{\theta}$ can be computed based on the bidding relationship matrix $\boldsymbol{B}$:

$$\boldsymbol{\theta} = \text{sgn}(\boldsymbol{B}\boldsymbol{\eta}). \quad (2)$$

3.3 Auction

Only those ads whose $\theta_j = 1$ will enter this stage. First, the click probabilities will be computed for these ads. We use $c(\cdot)$ to denote the click prediction function, which takes the pair of the query q and the ad a_j as input and predicts the click probability $c(j)$ of a_j if it is ranked at the first position in the ad list. The click prediction function is usually trained using the real click through rate in the historical sponsored search log. In the sponsored search system, the click probability $c(j)$ can be used to estimate the future revenue. Meanwhile, the system will also compute the quality scores for these ads. We use $h(\cdot)$ to denote the quality score function, which also takes the pair of the query q and the ad a_j as input. The quality score is an estimation of how relevant the ad, keywords, and landing page are to the search engine user who is seeing the ad. It can be used for ad ranking and pricing. Note that in some work, the ad click probability is regarded as the ad quality score. As mentioned before, we will treat them as two separate signals.

Then since one ad may bid on multiple selected keywords, it is associated with multiple bids. We need to determine which bid to use in the auction process. According to the industry practice, the highest bid will be used, and we denote it as $v(j)$, i.e.,

$$v(j) = \max\{\bar{v}(j,i)|b_{ji} = 1, \eta_i = 1\}. \quad (3)$$

Given the quality score and the final bid, GSP ranks ads according to the following rank score[1], which is defined as the product of quality score and bid,

$$g(j) = h(j)v(j), \forall \theta_j = 1.$$

Sorting the ads in the descending order of the rank scores generates a permutation $\zeta_\theta(\cdot)$. For ease of reference, we use $\zeta_\theta^{-1}(r)$ to denote the index of the ad ranked at position r ($r = 0, 1, \cdots, m-1$) by permutation $\zeta_\theta(\cdot)$, and then the permutation can be written as,

$$\begin{aligned} & \zeta_\theta(\{a_j|\theta_j = 1\}) \\ \rightarrow\ & a_{\zeta_\theta{}^1(1)}, a_{\zeta_\theta{}^1(2)}, \cdots, a_{\zeta_\theta{}^1(r)}, \cdots \\ & s.t.\ g(\zeta_\theta^{-1}(r)) \geqslant g(\zeta_\theta^{-1}(r+1)) \\ & \forall r,\ \theta_{\zeta_\theta{}^1(r)} = 1. \end{aligned}$$

The top ranked ads in the permutation will be shown to the user. Note that for simplicity we do not consider the reserve rank score, which is used in practice to filter out less competitive ads in order to increase search engine revenue. If some ads are clicked by the user, the advertisers will be charged according to the second-price rule, i.e., the payment is

$$\frac{g(\zeta_\theta^{-1}(r+1))}{h(\zeta_\theta^{-1}(r))}.$$

3.4 Utilities

Based on the discussions on the sponsored search system in Section 1 and Section 3, let us consider how the marketplace objective is defined and computed.

As mentioned in the introduction, we mainly consider three aspects when defining the marketplace objective, which reflects the utilities of the users, advertisers, and search engine respectively. In particular:

• We use the expected click to reflect the utility of the user, with higher click through rate usually indicating higher satisfaction of the user with the ads shown to him/her.

• We use the expected bid as the utility of the advertiser, which is a lower bound of the corresponding social welfare. Note that we assume the advertisers are conservative, which means no advertiser is bidding above his/her own valuation on the keyword. This assumption is reasonable for Leme and Tardos[10] have justified that bidding above the valuation is a dominated strategy. A strategy is called dominated if it is always better to play some other strategy, regardless of what opponents may do. Besides, the same assumption was also adopted in many other studies like [10-14]. With this assumption, the expected bid is a lower bound of the social welfare and thus the maximization of the expected bid can approach the maximization of the social welfare. Note that the meticulous analysis of the real relationship between value and bid is far beyond the scope of this paper. Besides, almost all existing models on the strategic behaviors of the advertisers assume full information is available, which is unrealistic in practice. Therefore, we take a rough approach, i.e., we regard the total expected bid, which is the lower bound of the social welfare, as the utility of the advertiser.

• We use the expected revenue as the utility of the search engine, which is the total payment from the advertisers.

With the notations given in Subsections 3.1∼3.3, we can obtain the mathematical forms of the above three

utilities. We then define the marketplace objective (denoted as $R(q)$) as the convex combination of them:

$$R(q)=\sum_{r=0}^{m-1}\Bigg(\alpha_1\times c(\zeta_\theta^{-1}(r))+\alpha_2\times v(\zeta_\theta^{-1}(r))\times c(\zeta_\theta^{-1}(r))+\alpha_3\times\frac{g(\zeta_\theta^{-1}(r+1))}{h(\zeta_\theta^{-1}(r))}\times c(\zeta_\theta^{-1}(r))\Bigg)\times D(r). \quad (4)$$

Here α_i $(i=1,2,3)$ are the balancing parameters satisfying $\alpha_i>0$ $(i=1,2,3)$, $\sum_{i=1}^{3}\alpha_i=1$, and m is the maximal number of ads shown on the search result page. Because the CTR (click through rate) varies according to different positions, we introduce a position discount function $D(r)$. In the above formulation,

• The term $c(\zeta_\theta^{-1}(r))$ is the predicted click probability of the ad $a_{\zeta_\theta{}^{-1}(r)}$ when it is ranked at the first position of the ad list. By multiplying it with the position discount $D(r)$, we will get the click probability of this ad when it is ranked at position r. This term corresponds to user utility. Note that we ignore the cases that more than one click happens in a single impression, which are very rare in practice.

• The term $v(\zeta_\theta^{-1}(r))\times c(\zeta_\theta^{-1}(r))\times D(r)$ is the product of the maximal bid of an ad and the ad click probability at position r. It corresponds to the advertiser utility.

• The term $\frac{g(\zeta_\theta{}^{-1}(r+1))}{h(\zeta_\theta{}^{-1}(r))}\times c(\zeta_\theta^{-1}(r))\times D(r)$ is the payment of the ad ranked at position r according to the second-price rule, if it is clicked. This term corresponds to the search engine utility.

Given the marketplace objective, it is not difficult to formalize the ad selection problem as the following optimization problem:

$$\begin{aligned}&\max_{\boldsymbol{\omega}}\sum_{q\in Q}R(q) \quad (5)\\ &s.t.\ \eta_i=\mathrm{sgn}(\varsigma(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i^{(q)})-t),\ i=1,\cdots,M,\\ &\boldsymbol{\theta}=\mathrm{sgn}(\boldsymbol{B}\boldsymbol{\eta}),\\ &v(j)=\max\{\bar{v}(j,i)|b_{ji}=1,\eta_i=1\},\forall\theta_j=1,\\ &g(j)=h(j)v(j),\forall\theta_j=1.\end{aligned}$$

Please note that in the above formulation, our goal is to learn the parameter $\boldsymbol{\omega}$ in the ad selection algorithm, by regarding the click prediction function, the quality score function, and the auction mechanism as known and fixed. By solving this optimization problem, we can get the optimal parameter vector $\boldsymbol{\omega}^*$ and use it in the future ad selection processes.

4 Smoothed Approximation

In Section 3, we have described the idea of learning the ad selection algorithm by maximizing the marketplace objective. However, the learning process is non-trivial. The optimization problem (5) is a nonlinear optimization while the marketplace objective (4) is non-convex, discontinuous, and indifferentiable with respect to the model parameter of the ad selection algorithm. To the best of our knowledge, there are no effective methods for directly solving this kind of problem.

To better understand and hopefully solve the problems with the discontinuity and the indifferentiability, we first need to analyze where the discontinuity and the differentiability come from. Basically, they are due to the following three functions in the marketplace objective:

• sgn *Function.* This discontinuous and indifferentiable function is used in (2), which describes the relationship between the selected keywords and the selected ads.

• max *Function.* This discontinuous and indifferentiable function is used in (3), which finds the highest bid to determine the bid of an ad for a given query.

• ζ *Function.* This discontinuous and indifferentiable function is used in (4) to define the ranking rule, which outputs a permutation of ads.

To effectively optimize the marketplace objective, we propose smoothing the aforementioned three functions. By doing so, we will be able to obtain a continuous and differentiable approximation of the marketplace objective. Then conventional optimization methods can be employed to maximize this approximated objective function to learn the parameter in the ad selection algorithm. However, even the objective is differentiable, it is still non-convex so that we may only obtain a local optimal. The good news is that the experimental results in Section 6 show that we can already get a good solution and it converges stably with random initial points.

4.1 Smoothed sgn Function

We can directly remove the sgn function as well as the threshold in (1) and let $\eta_i=\varsigma(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i^{(q)})$ to represent the probability of keyword k_i being selected. The sgn function in (2) can be effectively smoothed by using the sigmoid function $\varsigma(\cdot)$, i.e.,

$$\theta_j=\mathrm{sgn}\Big(\sum_{i=1}^{M}b_{ji}\eta_i\Big)\approx 2\varsigma\Big(\sum_{i=1}^{M}b_{ji}\eta_i\Big)-1.$$

The range of $\sum_{i=1}^{M} b_{ji}\eta_i$ is $[0, +\infty)$, and thus the range of θ_j is $[0, 1)$. We can approximately regard θ_j as the possibility of ad a_j being selected as a candidate. Furthermore, we define X_j as a Bernoulli random variable to indicate whether ad a_j is selected, with the successful probability θ_j.

4.2 Smoothed max Function

In order to smooth the max function, we employ the following probabilistic method. Specifically, given that a_j is selected, i.e., $X_j = 1$, we define the probability of $\bar{v}(j, i)$ as,

$$p(\bar{v}(j,i)|X_j = 1) = \frac{\varphi(\bar{v}(j,i))b_{ji}\eta_i}{\sum_{b_{jl}=1,\eta_l>0} \varphi(\bar{v}(j,l))b_{jl}\eta_l},$$

where $\varphi(\cdot)$ is a transformation function which can be polynomial, exponential, etc. Without loss of generality, we choose $\varphi(x) = e^{\tau x}$ as the transformation function, where τ is a positive coefficient.

Suppose $\bar{v}(j, \hat{i})$ is the highest bid among all $\bar{v}(j, i)$. Then the above formula can be rewritten as below.

$$\begin{aligned}
& p(\bar{v}(j,i)|X_j = 1) \\
= & \frac{e^{\tau\bar{v}(j,i)}b_{ji}\eta_i}{\sum_{b_{jl}=1,\eta_l>0} e^{\tau\bar{v}(j,l)}b_{jl}\eta_l} \\
= & \frac{e^{\tau(\bar{v}(j,i)-\bar{v}(j,\hat{i}))}b_{ji}\eta_i}{\sum_{b_{jl}=1,\eta_l>0} e^{\tau(\bar{v}(j,l)-\bar{v}(j,\hat{i}))}b_{jl}\eta_l} \\
= & \frac{e^{\tau(\bar{v}(j,i)-\bar{v}(j,\hat{i}))}b_{ji}\eta_i}{b_{j\hat{i}}\eta_{\hat{i}} + \sum_{b_{jl}=1,\eta_l>0,l\neq\hat{i}} e^{\tau(\bar{v}(j,l)-\bar{v}(j,\hat{i}))}b_{jl}\eta_l}.
\end{aligned}$$

Therefore, when τ is very large, $p(\bar{v}(j, \hat{i})|X_j = 1)$ will approach 1 and the probabilities corresponding to the other bids will approach 0. Thus, when we select a_j, $v(j) \equiv \bar{v}(j, \hat{i})$ can be approximately expressed as the following conditional expectation of $\bar{v}(j, i)$ on all the keyword k_i, i.e.,

$$\begin{aligned}
v(j) &\approx E(\bar{v}(j,i)|X_j = 1) \\
&= \sum_{i=1}^{M} p(\bar{v}(j,i)|X_j = 1)\bar{v}(j,i).
\end{aligned}$$

Note that we only give the definition of $v(j)$ when a_j is selected, for the input set for the maximum function in (3) will be $\emptyset$ when $\theta_j = 0$, i.e.,

$$\begin{aligned}
\theta_j = 0 &\implies \sum_i b_{ji}\eta_i = 0 \\
&\implies \{\bar{v}(j,i)|b_{ji} = 1, \eta_i = 1\} = \emptyset.
\end{aligned}$$

4.3 Smoothed ζ Function

The permutation function ζ is relatively more difficult to approximate, because it contains the ranking function. We employ a method similar to SoftRank[15] to smooth it. The basic idea is to regard the rank score of each ad as a random variable with a Gaussian distribution. Then the rank of an ad can be analytically expressed based on the score distribution of all the ads. With the rank distribution, we will be able to compute the expected payment for each ad.

Note that any unimodal distribution with good smoothness and controllable parameters is appropriate. We follow SoftRank to use Gaussian distribution as an example in the paper. The experimental results in Section 6 suggest that Gaussian distribution has already led to very promising results. We will investigate on other unimodal distributions in our future work.

In particular, we use $g(j) = h(j)v(j)$ as the mean of the Gaussian distribution, and set its variance as σ_s. That is,

$$p(s_j|X_j = 1) = \mathcal{N}(s_j|\bar{s}_j, \sigma_s^2) \equiv \mathcal{N}(s_j|g(j), \sigma_s^2).$$

Here s_j is the random variable for the rank score. Since the score distribution is only defined when a_j is selected, the probability should be conditional, given $X_j = 1$.

With the aforementioned score distribution, we can compute the probability that an ad is ranked above another. In particular, given ad $a_j \in A$, we define $\pi_{ij} \equiv \mathrm{P}(a_i \succ a_j)$ as the probability that another ad $a_i \in A$ is ranked above a_j in the final ad rank list. Note that we consider all the ads in A, because we need to go through all the ads to construct the rank distributions. Therefore, we will discuss four cases in order to define this probability.

1) If both a_i and a_j are selected, the probability that a_i beats a_j is $\mathrm{P}(S_i - S_j > 0|X_i = 1, X_j = 1)$ where S_i and S_j are drawn from $p(s_i|X_i = 1)$ and $p(s_j|X_j = 1)$ respectively. Then this probability is the integral of the difference of the two Gaussian random variables, which itself is a Gaussian. Therefore, we can write the probability as $\mathrm{P}(S_i - S_j > 0|X_i = 1, X_j = 1) = \int_0^\infty \mathcal{N}(s|\bar{s}_i - \bar{s}_j, 2\sigma_s^2)\mathrm{d}s$.

2) If a_i is not selected but a_j is selected (i.e., $X_i = 0,\ X_j = 1$), then it is easy to get $\mathrm{P}(S_i - S_j > 0|X_i, X_j) = 0$, indicating that a_i will never be ranked above a_j in the final ranked list.

3) We have $\mathrm{P}(S_i - S_j > 0|X_i, X_j) = 1$ when a_i is selected but a_j is not (i.e., $X_i = 1,\ X_j = 0$), indicating that the selected ad a_i will always be ranked above the unselected ad a_j.

4) To make the definition of the probability complete (the probabilities for all the cases sum up to one), we set $\mathrm{P}(S_i - S_j > 0 | X_i, X_j) = 1/2$ when $X_i = 0,\ X_j = 0$, which means a_i has half of the possibility to beat a_j if both of them are not selected.

The above discussions can be mathematically written as:

$$\begin{aligned}
&\mathrm{P}(a_i \succ a_j | X_i, X_j) \\
&= \begin{cases} \int_0^\infty \mathcal{N}(s|\bar{s_i} - \bar{s_j}, 2\sigma_s^2)\mathrm{d}s, & X_i = 1, X_j = 1, \\ 0, & X_i = 0, X_j = 1, \\ 1, & X_i = 1, X_j = 0, \\ 1/2, & X_i = 0, X_j = 0, \end{cases} \\
&\mathrm{P}(X_i, X_j) \\
&= \begin{cases} \theta_i\theta_j, & X_i = 1, X_j = 1, \\ (1-\theta_i)\theta_j, & X_i = 0, X_j = 1, \\ \theta_i(1-\theta_j), & X_i = 1, X_j = 0, \\ (1-\theta_i)(1-\theta_j), & X_i = 0, X_j = 0, \end{cases} \\
\pi_{ij} &\equiv \mathrm{P}(a_i \succ a_j) \\
&= \sum_{X_i=0}^{1}\sum_{X_j=0}^{1} \mathrm{P}(X_i, X_j) \times \mathrm{P}(a_i \succ a_j | X_i, X_j) \\
&= \theta_i(1-\theta_j) + (1-\theta_i)(1-\theta_j)/2 + \\
&\quad \theta_i\theta_j \int_0^\infty \mathcal{N}(s|\bar{s_i} - \bar{s_j}, 2\sigma_s^2)\mathrm{d}s. \qquad (6)
\end{aligned}$$

The permutation appears in the marketplace objective in two ways: the rank position of an ad (used in computing the click probability) and the two ads ranked adjacent to each other (used in computing the second-price payment). Therefore, to smooth it, we need to compute the rank distributions and the adjacent-pair distributions. In the following paragraphs, we demonstrate how we obtain them by using π_{ij}.

4.3.1 Rank Distribution

With π_{ij}, we can compute the rank distribution of each ad a_j. Let r_j be the rank of ad a_j, then its distribution is denoted as $p_j(r) \equiv \mathrm{P}(r_j = r)$. Here we take the same assumption as in [15] that π_{ij} $(i = 1, \cdots, j-1, j+1, \cdots, N)$ are independent with each other with the fixed index j. The range for r_j is $0, 1, \cdots, N-1$.

Then the distribution of r_j can be obtained by considering the rank r_j as a Binomial-like random variable, equal to the number of successes of $N-1$ Bernoulli trials, where the probability of success is π_{ij}. We can get this distribution by a recursive process. If we define the initial rank distribution for the ad a_j as $p_j^{(1)}(r)$, the rank can only be the position 0 since a_j is the only ad. Then we have the rest $N-1$ ads to be inserted to the ranked list.

$$p_j^{(i)}(r) = \begin{cases} \delta(r), & i = 1, \\ p_j^{(i-1)}(r-1)\pi_{ij} + p_j^{(i-1)}(r)(1-\pi_{ij}), & i \in [2, N]. \end{cases} \qquad (7)$$

Here $\delta(x) = 1$ if $x = 0$; otherwise, $\delta(x) = 0$.

We further define $p_j^{(i)}(r) = 0$ if $r < 0$ as the trivial case, and then we can get the final rank distribution $p_j(r) \equiv p_j^{(N)}(r)$. It is not difficult to see that the expectation of r_j is $E[r_j] = \sum_{i=1, i\neq j}^{N} \pi_{ij}$.

From the above recursive process, we can see that if an ad a_j is selected and it is good enough that every other ad cannot beat it, then it will be ranked at the top position with $r_j = 0$. If an ad a_j is not selected, it will be randomly ranked at some position below all the selected ads. Notice that the calculations on the rank distributions of different ads are independent, and thus it is easy to conduct a distributed implementation.

4.3.2 Adjacent-Pair Distribution

Then we compute the adjacent-pair distribution, i.e., $p_{j,i}(r) \equiv \mathrm{P}(r_j = r, r_i = r+1)$. Given π_{ji}, we can consider a_j and a_i together as a union in a recursive generating process. In the first step, we add the union of a_j and a_i into the ranked list and they are placed at position 0 and position 1 respectively. In each of the following steps, when we add a new ad a_l into the list, there may be three cases: a_l is ranked above a_j, a_l is ranked below a_i, and a_l is ranked between a_j and a_i. Given π_{ji}, the conditional probabilities of the three cases are $\tilde{\pi}_{lji}$, $\tilde{\pi}_{jil}$, and $\tilde{\pi}_{jli}$ respectively. (The definitions of the conditional probabilities are shown in Appendix A.) Here we only care about the first two cases because we have assumed that a_j and a_i are ranked adjacent to each other. Therefore, the recursive expression of $p_{j,i}(r)$ is written as,

$$\begin{aligned}
p_{j,i}^{(l)}(r|\pi_{ji}) &= \delta(r), \text{when } l = 1, \\
p_{j,i}^{(l)}(r|\pi_{ji}) &= p_{j,i}^{(l-1)}(r-1|\pi_{ji})\tilde{\pi}_{lji} + \\
&\quad p_{j,i}^{(l-1)}(r|\pi_{ji})\tilde{\pi}_{jil}, \text{when } 2 \leqslant l \leqslant N-1. \qquad (8)
\end{aligned}$$

Again, we define $p_{j,i}^{(l)}(r) = 0$ if $r < 0$ as the trivial case, and then we can get the final rank probability $p_{j,i}(r) \equiv p_{j,i}^{(N-1)}(r|\pi_{ji}) \times \pi_{ji}$. Similar to the rank distribution, we can regard $p_{j,i}(r)$ as a multinomial-like distribution with $p_{j,i}(r) = \mathrm{P}(Y_1 = r, Y_2 = 0, Y_3 =$

$N-r-2$), where Y_1, Y_2, Y_3 denote the random variables of the numbers that the ad a_l is inserted into the above three positions respectively.

4.4 Smoothed Objective Function

Based on the discussions in Subsections 4.1∼4.3, we can obtain the following smoothed approximation to the marketplace objective $R(\cdot)$ in (4) as $\mathcal{R}(\cdot)$,

$$\mathcal{R}(q) = \sum_{j=1}^{N}\sum_{r=0}^{m-1} c(j)D(r)\Big(\alpha_1 p_j(r) + \alpha_2 v(j)p_j(r) + \alpha_3 \sum_{i=1,i\neq j}^{N} \frac{g(i)}{h(j)} p_{j,i}(r)\Big).$$

Note in the above formula, the sum is over the ad indexes rather than the ad rank positions. As a result, it becomes continuous and differentiable, and therefore easy to optimize.

5 Solving Optimization Problem

Since the smoothed objective function becomes differentiable, we can choose to optimize it using the gradient descent method. In particular, the gradient can be computed according to the chain rule,

$$\frac{\partial \mathcal{R}}{\partial \boldsymbol{\omega}} = \frac{\partial \mathcal{R}}{\partial \boldsymbol{\eta}} \cdot \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\omega}}.$$

The first term $\frac{\partial \mathcal{R}}{\partial \boldsymbol{\eta}}$ can be derived from the differentiation of $p_j(r)$ over $\boldsymbol{\eta}$, which can be obtained recursively similar to $p_j^{(i)}(r)$, i.e.,

$$\frac{\partial \mathcal{R}}{\partial \boldsymbol{\eta}} = \frac{\partial \mathcal{R}}{\partial \boldsymbol{v}} \cdot \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{\eta}} + \frac{\partial \mathcal{R}}{\partial p_j(r)} \cdot \frac{\partial p_j(r)}{\partial \boldsymbol{\eta}} + \frac{\partial \mathcal{R}}{\partial p_{j,i}(r)} \cdot \frac{\partial p_{j,i}(r)}{\partial \boldsymbol{\eta}}.$$

The second term $\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\omega}}$ can be easily obtained by the model of the function $f(\boldsymbol{\omega}, \boldsymbol{x}_i^{(q)})$ and the sigmoid function $\eta_i = \varsigma(\boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x}_i^{(q)})$.

The derivation $\frac{\partial \mathcal{R}}{\partial \eta_t}$, $t = 1, \cdots, M$, is

$$\frac{\partial \mathcal{R}}{\partial \eta_t} = \sum_{j=1}^{N}\sum_{r=0}^{m-1} c(j)D(r)\Bigg(\alpha_1 \frac{\partial p_j(r)}{\partial \eta_t} + \alpha_2 \left(\frac{\partial v(j)}{\partial \eta_t} p_j(r) + v(j)\frac{\partial p_j(r)}{\partial \eta_t}\right) + \alpha_3 \sum_{i=1,i\neq j}^{N} \frac{h(i)}{h(j)} \left(\frac{\partial v(i)}{\partial \eta_t} p_{j,i}(r) + v(i)\frac{\partial p_{j,i}(r)}{\partial \eta_t}\right)\Bigg).$$

For simplicity, we denote $v_j = v(j)$ and $v_{ij} = \bar{v}(j,i)$. According to the definition of $v(j)$ in (3), we have

$$v_j = \frac{\sum_{i=1}^{M} v_{ji}\varphi(v_{ji})b_{ji}\eta_i}{\sum_{i=1}^{M} \varphi(v_{ji})b_{ji}\eta_i},$$

and its derivative is,

$$\begin{aligned}\frac{\partial v_j}{\partial \eta_t} &= \frac{v_{jt}\varphi(v_{jt})b_{jt}}{\sum_{i=1}^{M}\varphi(v_{ji})b_{ji}\eta_i} - \frac{\left(\sum_{i=1}^{M} v_{ji}\varphi(v_{ji})b_{ji}\eta_i\right)\varphi(v_{jt})b_{jt}}{\left(\sum_{i=1}^{M}\varphi(v_{ji})b_{ji}\eta_i\right)^2} \\ &= \left(v_{jt}\varphi(v_{jt})b_{jt}\sum_{i=1}^{M}\varphi(v_{ji})b_{ji}\eta_i - \varphi(v_{jt})b_{jt}\times\sum_{i=1}^{M} v_{ji}\varphi(v_{ji})b_{ji}\eta_i\right)\Bigg/ \left(\sum_{i=1}^{M}\varphi(v_{ji})b_{ji}\eta_i\right)^2 \\ &= \frac{\varphi(v_{jt})b_{jt}\sum_{i=1}^{M}(v_{jt}-v_{ji})\varphi(v_{ji})b_{ji}\eta_i}{\left(\sum_{i=1}^{M}\varphi(v_{ji})b_{ji}\eta_i\right)^2}.\end{aligned}$$

We also need a recursive process to obtain the derivative of $p_j(r)$. Denoting $\phi_{t,j}^{(i)}(r) = \frac{\partial p_j^{(i)}(r)}{\partial \eta_t}$, we can get the derivative from (7):

$$\begin{aligned}\phi_{t,j}^{(1)}(0) &= 0, \\ \phi_{t,j}^{(i)}(r) &= \phi_{t,j}^{(i-1)}(r-1)\pi_{ij} + \phi_{t,j}^{(i-1)}(r)(1-\pi_{ij}) + \left(p_j^{(i-1)}(r-1) - p_j^{(i-1)}(r)\right)\frac{\partial \pi_{ij}}{\partial \eta_t}.\end{aligned}$$

The derivative of $p_{j,i}(r)$ can be calculated in a similar way to that of $p_j(r)$. Denoting

$$\psi_{t,j,i}^{(l)}(r) = \frac{\partial p_{j,i}^{(l)}(r|\pi_{ji})}{\partial \eta_t},$$

we can obtain the derivative from the recursive expression (8):

$$\begin{aligned}\psi_{t,j,i}^{(1)}(0) &= 0, \\ \psi_{t,j,i}^{(l)}(r) &= \psi_{t,j,i}^{(l-1)}(r-1)\tilde{\pi}_{lji} + \psi_{t,j,i}^{(l-1)}(r)\tilde{\pi}_{jil} + p_{j,i}^{(l-1)}(r-1|\pi_{ji})\frac{\partial \tilde{\pi}_{lji}}{\partial \eta_t} + p_{j,i}^{(l-1)}(r|\pi_{ji})\frac{\partial \tilde{\pi}_{jil}}{\partial \eta_t}.\end{aligned}$$

According to the definition of $p_{j,i}(r)$, we can get

$$\frac{\partial p_{j,i}(r)}{\partial \eta_t} = \psi_{t,j,i}^{(N-1)}(r)\pi_{ji} + p_{j,i}^{(N-1)}(r|\pi_{ji})\frac{\partial \pi_{ji}}{\partial \eta_t}.$$

The derivative of π_{ij} over η_t can be expressed as

$$\frac{\partial \pi_{ij}}{\partial \eta_t} = \frac{\partial \pi_{ij}}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \boldsymbol{\theta}}{\partial \eta_t} + \frac{\partial \pi_{ij}}{\partial \boldsymbol{v}} \cdot \frac{\partial \boldsymbol{v}}{\partial \eta_t}$$
$$= \left(\frac{\partial \pi_{ij}}{\partial \theta_i} \ \frac{\partial \pi_{ij}}{\partial \theta_j} \right) \begin{pmatrix} \frac{\partial \theta_i}{\partial \eta_t} \\ \frac{\partial \theta_j}{\partial \eta_t} \end{pmatrix} + \left(\frac{\partial \pi_{ij}}{\partial v_i} \ \frac{\partial \pi_{ij}}{\partial v_j} \right) \begin{pmatrix} \frac{\partial v_i}{\partial \eta_t} \\ \frac{\partial v_j}{\partial \eta_t} \end{pmatrix}.$$

From (6) and considering the fact

$$\frac{\partial}{\partial \mu} \int_0^\infty \mathcal{N}(x|\mu, \sigma^2) \mathrm{d}x = \mathcal{N}(0|\mu, \sigma^2),$$

we can obtain the derivatives of π_{ij} over $\boldsymbol{\theta}$ and $\boldsymbol{v}$ as the following forms. $\frac{\partial \pi_{ij}}{\partial \theta_t} = \frac{1-\theta_j}{2} + \theta_j \int_0^\infty \mathcal{N}(s|\bar{s}_i - \bar{s}_j, 2\sigma_s^2)\mathrm{d}s$, when $t = i, t \neq j$; $\frac{\partial \pi_{ij}}{\partial \theta_t} = -(1+\theta_i)/2 + \theta_i \int_0^\infty \mathcal{N}(s|\bar{s}_i - \bar{s}_j, 2\sigma_s^2)\mathrm{d}s$, when $t \neq i, t = j$; $\frac{\partial \pi_{ij}}{\partial \theta_t} = 0$, when $t \neq i, t \neq j$.

$$\frac{\partial \pi_{ij}}{\partial v_t} =$$
$$\begin{cases} \theta_i \theta_j u_i \mathcal{N}(0|u_i v_i - u_j v_j, 2\sigma_s^2), & t = i, t \neq j, \\ -\theta_i \theta_j u_j \mathcal{N}(0|u_i v_i - u_j v_j, 2\sigma_s^2), & t \neq i, t = j, \\ 0, & t \neq i, t \neq j. \end{cases}$$

With the definition of $\boldsymbol{\theta}$ in Section 3, we can obtain the derivative of $\boldsymbol{\theta}$ over $\boldsymbol{\eta}$ easily. The derivatives of conditional probabilities $\tilde{\pi}_{lji}, \tilde{\pi}_{jil}$ can be calculated similarly. To save space, we put the calculation of the derivatives in an online appendix file②.

Therefore, the gradient method can be implemented to compute the optimal parameter vector $\boldsymbol{\omega}^*$. In the industry practice, there are *exact match* and *advanced match* for query-keyword matching. The trained model can be used in two ways: 1) we can use it directly in the online ad selection platform; 2) we can use it offline to generate a static table of query-keyword pairs, and apply the query-keyword mappings in the table for advanced match.

6 Experimental Evaluation

In this section, we evaluate our proposed method by comparing it with four baseline algorithms on a real-world dataset. We simulated a sponsored search system to validate the benefit of the ad selection methods for the users, the advertisers, and the search engine. The experimental results show that the proposed global optimization method significantly outperforms the baselines on several metrics. Furthermore, we provide a study to elaborate the effectiveness of our method.

6.1 Dataset

The data used in the experiments is sampled from the sponsored search log of a commercial search engine in the period of two months. We sampled 14 912 queries in May 2012 for training and another 17 487 queries in June 2012 for test. We first extracted the keyword candidates for these queries using all the matching rules in the sponsored search system, and then extracted all the associated ads according to the bidding table in the ad database. The quality scores and the predicted click probability of these ads are also extracted for the simulation of the auction mechanism. Finally, we got over 300 thousand query-keyword pairs, over 700 thousand query-ad pairs and over 1.5 million query-keyword-ad tuples. The details can be seen in Table 2.

Besides, in order to prepare the training objectives for the baseline methods, we extracted the query-keyword level normalized click through rate (nCTR), social welfare, and revenue. The normalized click through rate is calculated based on the sum of adjusted clicks over the sum of adjusted impressions. The adjustments are conducted by multiplying the number of clicks/impressions in different ad positions with the position discount function $D(r)$. As explained in Subsection 3.4, we use the sum of bids of the clicked ads to approximate the social welfare. The revenue is calculated as the sum of the payoffs from advertisers for the given query-keyword pair.

6.2 Baselines

As discussed in Section 2, there are two categories of existing ad selection algorithms. We first calculated the cosine similarity between query and keyword as the baseline on behalf of the relevance-based models. Note

Table 2. Statistics of the Dataset

	Number of Queries	Number of Query-Keyword Pairs	Number of Query-Ad Pairs	Number of Query-Keyword-Ad Tuples
May 2012	14 912	342 791	743 050	1 577 683
June 2012	17 487	427 366	928 418	2 030 186

② https://www.dropbox.com/s/qczlc9mh6jh8yg2/Appendix.pdf, Jan. 2015.

that we did not choose the methods of [3-4] because both of them use external text streams like the landing pages. It is beyond the scope of our study, for we regard that the external information might be extracted as features in the models. For the click information based methods, we chose Simrank++[9]. Besides the above two categories, we used a classification method and a regression method both trained on the query-keyword features under the supervision of historical information including nCTR, social welfare, and revenue. The implementation of the four baselines is explained as below.

• For the first baseline, the cosine similarity between query q and keyword k is defined as the similarity between their vector representations based on term frequency, i.e.,

$$\mathrm{sim}_{cosine}(q,k) = \frac{\#CommonTerms}{\sqrt{Len(q)} \times \sqrt{Len(k)}},$$

where $Len(q)$ and $Len(k)$ are the numbers of terms in q and k respectively. We denote it by Cosine for ease of reference.

• For the second baseline Simrank++, we merged all the queries and keywords appeared in the test data as one side and extracted the clicked ads in May 2012 as the other side to build the click bipartite graph. There are 486 024 queries and keywords in the test data, and 306 816 of them are associated with 1.3 million clicked ads. The generated click bipartite graph contains 3.2 million edges, the scale of which is even larger than the main subgraph in [9]. Thus, we implemented the pruning technique in the original Simrank paper[16] with a radius 2, and ran 7 iterations as suggested by [16]. Though the click graph is very large, there are still many pairs of queries and keywords without predicted similarities. Among the 427 366 query-keyword pairs in the test data, there are only 231 069 pairs with Simrank++ scores, indicating that they do not have common clicked ads in May 2012, if we regard both the queries and keywords as input queries. This is also a limitation of the click information based methods.

• For the third and the fourth baselines, we combined nCTR, social welfare, and revenue as the training targets. We normalized the three values to standard normal distribution $\mathcal{N}(0,1)$ and then summed them up as the targets. Among the 342 791 query-keyword pairs in the training data, there are 233 586 pairs with impressions and only 80 552 pairs of them with clicks. For the classification model, we used the query-keyword pairs with more than 20 impressions but with zero click as the negative training examples, and used the query-keyword pairs with more than 20 impressions and with the nCTR higher than 0.04 as the positive training examples. Both groups of the examples contain more than 60 thousand query-keyword pairs. For the regression model, we took use of all the query-keyword pairs with non-zero impressions. We used SVM-light[17] to train the classification and the regression models using the linear setting. For ease of reference, we denote the two baselines as SVM-Cls and SVM-Reg respectively.

6.3 Query-Keyword Features

We extracted three categories of query-keyword features including keyword related features, query related features, and query-keyword related features.

• *Keyword Related Features.* Given a keyword, we extracted the average bid, the number of ads that bid it, the number of orders that bid it, the number of campaigns that bid it, and the number of advertisers that bid it from the advertiser database and the ad database, and extracted the number of ad impressions, the number of ad clicks, and the average ad click position for the keyword in a period of time (e.g., one month) from the auction log.

• *Query Related Features.* Given a query, we regarded the query as a keyword and extracted the similar features as the keyword-related features.

• *Query-Keyword Related Features.* Given a pair of query and keyword, we extracted the cosine similarity, edit distance, and word distance between them. We also used two features computed from the translation models[18].

6.4 Evaluation Metrics and Simulation

We run a simulation of the sponsored search system for the comparing algorithms, and check their performance on the estimated nCTR for the users, the estimated social welfare for the advertisers, and the estimated revenue for the search engine.

In the simulation, for a given query, each algorithm will select a set of keywords and the corresponding ads. With the quality score and the bid, we can calculate the rank score of each ad. Then, according to (4), we can calculate the three parts of the marketplace objective respectively. After that, we multiply the marketplace objectives by the normalized query frequencies. Thus, we can sum all the queries up and obtain the estimated nCTR, the estimated social welfare, and the estimated revenue.

6.5 Scalability

We first make a complexity analysis on the proposed model, and then explain how we implement it when the scale of the problem increases.

For a given query, we assume that there are M keywords and N ads involved, and we consider the top m positions in the ad list. Then the complexity of the probability π_{ij} is $O(N^2)$, and the complexity of the rank distribution $p_j(r)$ for the top m positions is $O(N^2m)$. The complexity of the triple probability $\tilde{\pi}_{lji}$ is $O(N^3)$, and in each calculation, it will call a numerical integration algorithm. The computation complexity of the adjacent-pair distribution for the top m positions is $O(N^3m)$, and the complexity of calculating its derivatives is similar. Therefore, the calculation of the adjacent-pair distribution is the bottleneck for the proposed model.

There are several ways to speed up the algorithm. One way is to approximate the rank distribution $p_j(r)$ with Rank-Binomial distribution as presented in SoftRank[15]. Similarly, the adjacent-pair distribution can also be approximated by multinomial-like distribution as we discussed in Subsection 4.3.2. These approximations will significantly reduce the computational complexity. Another way is to use the parallel computing techniques. As the calculations on rank distributions of different ads are independent, we can implement them in parallel. The implementation is beyond the scope of the paper.

Besides the above complexity analysis, we should note that the test process is quite fast and only needs to perform an inner product to sort the keywords. In industry practice, we can update historical features frequently but only retrain the model parameter in proper period, which is the strategy that the quality score computation algorithm and the ad click prediction algorithm adopt. We can further control the volume and quality of the training set according to the computational capabilities, and then the proposed method can be deployed to the search engine.

6.6 Parameter Setting

For the parameters of the proposed model, we set the exponent of the sigmoid function for $\boldsymbol{\eta}$ to 0.2, i.e., $\varsigma(x) = 1/(1 + e^{-0.2x})$, to make it like a linear function. We set the exponent of the sigmoid function for $\boldsymbol{\theta}$ to 3 so that it approaches to a stepwise function, which looks more similar to the definition of b_{ji}. We set the variance of the rank score to a fixed value 10 and we only consider the top four positions in the ad list, corresponding to the mainline ads. The balancing parameters $\alpha_1, \alpha_2, \alpha_3$ of the marketplace objective are set to 0.8, 0.1, 0.1 to make the three parts of the objective in the same level of magnitude. Note that a search engine might balance the benefits of users, advertisers, and itself according to some pre-defined curve. We just use these balancing parameters to show the performance of the proposed algorithm. The model is trained using stochastic gradient descent with a random initial parameter vector $\boldsymbol{\omega}$ in 20 trails and most of them converge to a stable optimal $\boldsymbol{\omega}^*$ with the relative error less than 0.2%. We use the best one to compare with the baselines.

6.7 Ad Selection Performance

We denote the proposed global optimization model as Global for ease of reference, and compare its performance with those of the baseline algorithms Cosine, Simrank++, SVM-Cls, and SVM-Reg. We report the performance of these algorithms with respect to estimated nCTR, estimated social welfare, and estimated revenue.

We sort the keywords of each query increasingly according to the predicted scores and split them into 20 buckets. We drop the bucket of keywords with the smallest scores in each step, and get a declining curve for every model on each of the three evaluation metrics. The curves are shown in Figs. 2∼4. Besides, we also compute the area under curve (AUC) to compare the performance of these models in Table 3. Note that we normalize the values under each evaluation metric by dividing them by a certain value in the corresponding results, to protect the business information of the involved commercial search engine.

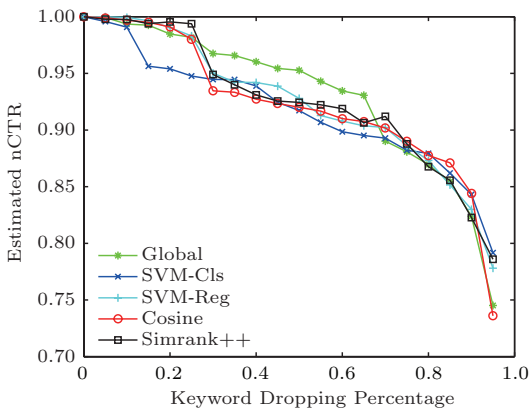


Fig.2. Dropping curve of simulated nCTR.

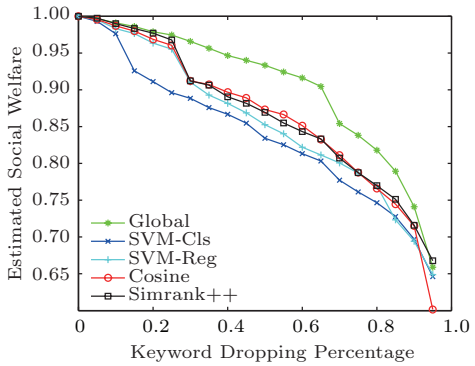


Fig.3. Dropping curve of simulated social welfare.

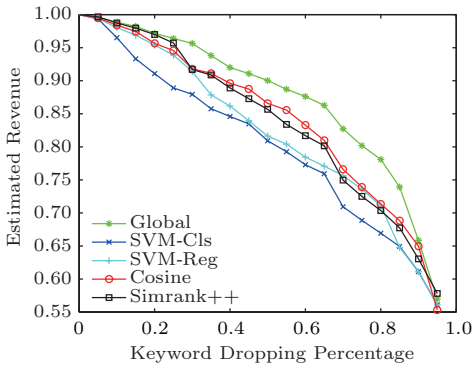


Fig.4. Dropping curve of simulated revenue.

Table 3. AUC of the Five Models on Three Metrics

	nCTR	Social Welfare	Revenue
Cosine	0.897 8	0.842 3	0.822 2
Simrank++	0.901 4	0.845 3	0.817 6
SVM-Cls	0.893 3	0.816 0	0.781 7
SVM-Reg	0.900 8	0.833 8	0.801 5
Global	**0.906 2**	**0.880 7**	**0.851 6**

From the experimental results, we have the following observations.

• The Global method outperforms the four baselines in estimated social welfare and estimated revenue in all buckets. For estimated nCTR, Global only outperforms the baselines in the middle buckets. However, the AUC of Global for estimated nCTR is still the largest in Table 3. Therefore, we can claim that Global achieves the best performance in the experiments.

• Global performs better than SVM-Cls and SVM-Reg, though the three models are trained on the same feature set. The reason is explained as below. SVM-Cls and SVM-Reg take the historical information to build the training targets, and thus the models cannot fit the future predictions very well. Global considers the effectiveness of the downstream components when computing its marketplace objective, thereby it can generate better predictions compared with SVM-Cls and SVM-Reg.

• Generally, Cosine and Simrank++ perform better than SVM-Cls and SVM-Reg. It shows that the query-keyword features are far from enough to build a good model. They might be easily beaten by some simple heuristics. We should take a global view in the ad selection problem towards the marketplace objective to combine the features.

To sum up, the proposed Global model outperforms the four baseline ad selection methods on all the three evaluation metrics.

6.8 Statistical Study

We analyze how the Global model outperforms the baseline methods. Our key conclusion is that: the baseline methods tend to select the keywords that can maximize the nCTR, social welfare, and revenue based on the historical auction and ad click data; differently, the Global method considers the downstream components like the auction mechanism in its optimization so that it can select the keywords that will maximize the nCTR, social welfare, and revenue in future auctions. To draw the above conclusion, we conduct the following statistical study.

For each of the five models in the comparison experiments, we keep the top 70% ranked keywords for each query as selected keywords and put the corresponding ads in the downstream auction. Then for each metric (nCTR, social welfare, and revenue), we compute the percentage of the metric earned by the top 70% keywords over all the keywords. We check the percentages of the metrics for the five models in the historical ad click data in the training set (May 2012), and in the simulated sponsored search results in the test set (June 2012). The values are listed in Table 4 and Table 5. For example, the last value 95.66% in the last row in Table 5 means that the top 70% keywords selected by the Global method help the search engine earn 95.66% revenue compared with selecting the 100% keywords on the simulated results in the training set.

From these tables, we have the following observations.

• From Table 4, the Global model does not always perform the best on the metrics. Simrank++ achieves the best on both nCTR and revenue, while SVM-Cls achieves the best on social welfare. The reason is explained as below. The training of Global does not rely much on the historical data; differently, the training of some baselines highly depends on the historical data. For instance, the computation of Simrank++ is conducted on the click graph built from the historical ad

click data; for another instance, the training target of SVM-Cls is a combination of historical nCTR, social welfare, and revenue.

Table 4. Metric Percentages of the Five Models on Historical Data in Training Set

	nCTR(%)	Social Welfare(%)	Revenue(%)
Cosine	75.16	99.10	98.09
Simrank++	**81.13**	99.27	**98.88**
SVM-Cls	77.25	**99.43**	97.89
SVM-Reg	76.42	98.97	97.83
Global	76.30	99.08	97.99

• From Table 5, the Global model outperforms all the four baselines in all metrics. The reason is explained below. Our method leverages much information from the downstream components in a global view in the training process which is closer to the real application, and thus our learned model can perform better when working together with the downstream components in the test process. The sponsored search system is divided into several parts, but only when these parts fit each other can it be a united system.

Table 5. Metric Percentages of the Five Models on Simulated Results in Test Set

	nCTR(%)	Social Welfare(%)	Revenue(%)
Cosine	93.46	91.20	91.81
Simrank++	94.90	91.22	91.70
SVM-Cls	94.47	88.84	87.95
SVM-Reg	95.02	91.04	91.40
Global	**96.75**	**96.57**	**95.66**

7 Conclusions and Future Work

In this paper, we argued that a good ad selection algorithm should perform global optimization for the marketplace objective for the entire sponsored search system, instead of just optimizing a locally defined objective. Given that the marketplace objective is discontinuous and indifferentiable, we proposed a set of smoothing techniques so as to obtain a smoothed approximation to the marketplace objective. After that, we employed a gradient descent method to optimize the smoothed marketplace objective, in order to learn the desired ad selection model. We tested our proposed algorithm using the sponsored search logs from a commercial search engine. The experimental results have shown that the proposed method outperforms several conventional ad selection algorithms in terms of several evaluation metrics.

For the future study, we plan to work on the following aspects. First, we will study alternative methods to smooth the marketplace objective and compare their effectiveness. Second, we will study the approximation ratio of the smoothed marketplace objective, so as to provide a theoretical guarantee on the proposed approach. Third, in this paper, we regard the click prediction algorithm and auction mechanism as fixed components when learning the ad selection algorithm. In the future, we plan to optimize all these components simultaneously, which may potentially generate even better results.

References

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

[2] Varian H R. Position auctions. *International Journal of Industrial Organization*, 2007, 25(6): 1163-1178.

[3] Broder A Z, Ciccolo P, Fontoura M, Gabrilovich E, Josifovski V, Riedel L. Search advertising using Web relevance feedback. In *Proc. the 17th ACM Conf. Information and Knowledge Management*, October 2008, pp.1013-1022.

[4] Choi Y, Fontoura M, Gabrilovich E, Josifovski V, Mediano M, Pang B. Using landing pages for sponsored search ad selection. In *Proc. the 19th WWW*, April 2010, pp.251-260.

[5] Broder A Z, Ciccolo P, Gabrilovich E, Josifovski V, Metzler D, Riedel L, Yuan J. Online expansion of rare queries for sponsored search. In *Proc. the 18th WWW*, April 2009, pp.511-520.

[6] Fuxman A, Tsaparas P, Achan K, Agrawal R. Using the wisdom of the crowds for keyword generation. In *Proc. the 17th WWW*, April 2008, pp.61-70.

[7] Hillard D, Schroedl S, Manavoglu E, Raghavan H, Leggetter C. Improving ad relevance in sponsored search. In *Proc. the 3rd ACM International Conference on Web Search and Data Mining*, February 2010, pp.361-370.

[8] Wang H, Liang Y, Fu L, Xue G R, Yu Y. Efficient query expansion for advertisement search. In *Proc. the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2009, pp.51-58.

[9] Antonellis I, Molina H G, Chang C C. Simrank++: Query rewriting through link analysis of the click graph. *Proc. VLDB Endow.*, 2008, 1(1): 408-421.

[10] Leme R P, Tardos E. Pure and Bayes-Nash price of anarchy for generalized second-price auction. In *Proc. the 51st IEEE Annual Symposium on Foundations of Computer Science*, October 2010, pp.735-744.

[11] Caragiannis I, Kaklamanis C, Kanellopoulos P, Kyropoulou M. On the efficiency of equilibria in generalized second-price auctions. In *Proc. the 12th ACM Conference on Electronic Commerce*, June 2011, pp.81-90.

[12] Lucier B, Paes Leme R. GSP auctions with correlated types. In *Proc. the 12th ACM Conf. Electronic Commerce*, June 2011, pp.71-80.

[13] Christodoulou G, Kovács A, Schapira M. Bayesian combinatorial auctions. In *Proc. the 35th ICALP, Part I*, July 2008, pp.820-832.

[14] Lucier B, Borodin A. Price of anarchy for greedy auctions. In *Proc. the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2010, pp.537-553.

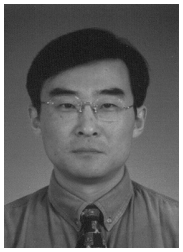
[15] Taylor M, Guiver J, Robertson S, Minka T. SoftRank: Optimizing non-smooth rank metrics. In *Proc. the Int. Conf. Web Search and Web Data Mining*, February 2008, pp.77-86.

[16] Jeh G, Widom J. SimRank: A measure of structural-context similarity. In *Proc. the 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, July 2002, pp.538-543.

[17] Joachims T. Making large-scale support vector machine learning practical. In *Advances in Kernel Methods*, Schölkopf B, Burges C J C, Smola A J (eds.), MIT Press, Cambridge, MA, USA, 1999, pp.169-184.

[18] Brown P F, Pietra V J D, Pietra S A D, Mercer R L. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 1993, 19(2): 263-311.

Qing Cui is a Ph.D. candidate of Department of Mathematical Science, Tsinghua University, Beijing. He received his B.S. degree in mathematics and applied mathematics from Tsinghua University in 2010. His research interest includes machine learning, data mining and matching theory.



Feng-Shan Bai is a professor in applied mathematics at Tsinghua University, Beijing. He received his B.S. degree in applied mathematics from Jilin University, Changchun, and M.S. and Ph.D. degrees in applied mathematics from Tsinghua University, Beijing, in 1989. His current research interests include numerical algorithms in data mining, mathematical software and mathematical modeling.

Bin Gao is a lead researcher in Internet Economics and Computational Advertising Group, Microsoft Research Asia, Beijing. Prior to joining Microsoft, he got his Ph.D. degree in applied mathematics from Peking University in 2006, and his B.S. degree in computational mathematics from Shandong University, Jinan, in 2001. His research interests include machine learning, data mining, information retrieval, and computational advertising. He has authored two book chapters, 30 papers in top conferences and journals, and over 20 granted or pending patents. He co-authored the best student paper at SIGIR (2008). He serves as PC for SIGIR (2009, 2014), WWW (2011, 2013), and senior PC for CIKM (2011). He is a reviewer for TKDE, TIST, PRL, IRJ, etc. He is a tutorial speaker at WWW (2011) and SIGIR (2012). He is a workshop organizer at ICDM (2012), SIGIR (2013), KDD (2013), and ICML (2014). He is a member of ACM and IEEE.

Tie-Yan Liu is a senior researcher and research manager at Microsoft Research Asia, Beijing. Prior to joining Microsoft, he got his Ph.D. degree in 2003 and bachelor degree in 1998 both in electronic engineering from Tsinghua University, Beijing. His research interests include machine learning, information retrieval, data mining, computational advertising, and algorithmic game theory. He is well known for his pioneer work on learning to rank for information retrieval. He has authored the first book in this area, and published tens of papers on both algorithms and theorems of learning to rank. In addition, his paper on graph mining won the Best Student Paper Award of SIGIR 2008; his paper on video shot boundary detection won the Most Cited Paper Award of the Journal of Visual Communication and Image Representation (2004∼2006); and his work on Internet economics won the Research Break-Through Award of Microsoft Research Asia (2012). Tie-Yan is a program committee co-chair of ACML 2015, WINE 2014, AIRS 2013, and RIAO 2010, a local co-chair of ICML 2014, a tutorial co-chair of SIGIR 2016 and WWW 2014, a doctorial consortium co-chair of WSDM 2015, a demo/exhibit co-chair of KDD 2012, and an area/track chair or senior program committee member of many conferences including KDD 2015, ACML 2014, SIGIR (2008∼2011), AIRS (2009∼2011), and WWW (2011, 2015). He is an associate editor of ACM Transactions on Information System, an editorial board member of Information Retrieval Journal and Foundations and Trends in Information Retrieval. He is a keynote speaker at ECML/PKDD (2014), CCIR (2011, 2014), CCML (2013), and PCM (2010), a tutorial speaker at SIGIR (2008, 2010, 2012), WWW (2008, 2009, 2011), and KDD (2012), and a plenary panelist at KDD (2011). He is a senior member of ACM and IEEE, as well as a senior member and distinguished speaker of CCF. He is currently an adjunct professor of Carnegie Mellon University (LTI), Nankai University, Sun Yat-Sen University, and University of Science and Technology of China, and an Honorary Professor of University of Nottingham.

Appendix A Conditional Probabilities

In the first step of the recursive process, we add the union of a_j and a_i into the rank list and they are placed at position 0 and position 1 respectively. In each of the following steps, when we add a new ad a_l into the list, there will be three cases: a_l is ranked above a_j, a_l is ranked below a_i, and a_l is ranked between a_j and a_i. The conditional probabilities of the three cases are calculated as below.

$$\mathrm{P}(a_l \succ a_j | a_j \succ a_i) = \frac{\mathrm{P}(a_l \succ a_j \succ a_i)}{\mathrm{P}(a_j \succ a_i)},$$
$$\mathrm{P}(a_i \succ a_l | a_j \succ a_i) = \frac{\mathrm{P}(a_j \succ a_i \succ a_l)}{\mathrm{P}(a_j \succ a_i)},$$
$$\mathrm{P}(a_j \succ a_l \succ a_i | a_j \succ a_i) = \frac{\mathrm{P}(a_j \succ a_l \succ a_i)}{\mathrm{P}(a_j \succ a_i)}.$$

Like the pairwise beat probability, the calculation of $\mathrm{P}(a_i \succ a_j \succ a_k)$ can be separated into several cases,

$$\begin{aligned}&\mathrm{P}(a_i \succ a_j \succ a_k)\\ =&\begin{cases}\mathrm{P}(S_i > S_j > S_k), & X_i = X_j = X_k = 1,\\ \pi_{ij}, & X_i = X_j = 0,\ X_k = 1,\\ 0.5, & X_i = 1,\ X_j = X_k = 0,\\ 1/6, & X_i = X_j = X_k = 0,\\ 0, & \text{otherwise.}\end{cases}\end{aligned}$$

We can see that the most difficult part in calculation is $\mathrm{P}(S_i > S_j > S_k)$. As we discussed in Section 4, S_i, S_j, S_k are drawn from the Gaussian random variables $p(s_i), p(s_j), p(s_k)$ with different means $\bar{s}_i, \bar{s}_j, \bar{s}_k$ but the same variance σ_s^2. Here we use μ_i, μ_j, μ_k to denote the means and σ^2 to denote the variance for simplicity. Since the rank score distributions are independent, we can get the joint distribution by simply multiplying them together, i.e., $p(s_i, s_j, s_k) = p(s_i)p(s_j)p(s_k)$. With the joint distribution, we can get

$$\begin{aligned}&\mathrm{P}(S_i > S_j > S_k)\\ =&\int_{-\infty}^{+\infty} \mathrm{d}s_j \int_{-\infty}^{s_j} \mathrm{d}s_k \int_{s_j}^{+\infty} p(s_i)p(s_j)p(s_k)\mathrm{d}s_i\\ =&\int_{-\infty}^{+\infty} \mathrm{d}s_j \int_{-\infty}^{s_j} \left(1 - \Phi(s_j|\mu_i, \sigma^2)\right) p(s_j)p(s_k)\mathrm{d}s_k\\ =&\int_{-\infty}^{+\infty} p(s_j)\Phi(s_j|\mu_k, \sigma^2)\left(1 - \Phi(s_j|\mu_i, \sigma^2)\right)\mathrm{d}s_j.\end{aligned}$$

Here $\Phi(x|\mu, \sigma^2) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right)$ is the cumulative distribution function of normal random variable $\mathcal{N}(\mu, \sigma^2)$, and $\mathrm{erf}(x)$ is the error function[③]. Then we can calculate this probability with a one-dimensional numerical integration. There are several standard methods to compute the numerical integration[④], and we can also make some preprocessing to trade space for time. Combined with the previous formulas, we can calculate the conditional probabilities.

③ Wikipedia. Error function. http://en.wikipedia.org/wiki/Error_function, Jan. 2015.

④ Wikipedia. Numerical integration. http://en.wikipedia.org/wiki/Numerical_integration, Jan. 2015.