Sampling Dilemma: Towards Effective Data Sampling for Click Prediction in Sponsored Search

Jun Feng
Tsinghua University
feng-j13@mails.tsinghua.edu.cn

Jiang Bian
Microsoft Research
jibian@microsoft.com

Taifeng Wang
Microsoft Research
taifengw@microsoft.com

Wei Chen
Microsoft Research
wche@microsoft.com

Xiaoyan Zhu
Tsinghua University
zxy-dcs@tsinghua.edu.cn

Tie-Yan Liu
Microsoft Research
tyliu@microsoft.com

ABSTRACT

Precise prediction of the probability that users click on ads plays a key role in sponsored search. State-of-the-art sponsored search systems typically employ a machine learning approach to conduct click prediction. While paying much attention to extracting useful features and building effective models, previous studies have overshadowed seemingly less obvious but essentially important challenges in terms of data sampling. To fulfill the learning objective of click prediction, it is not only necessary to ensure that the sampled training data implies the similar input distribution compared with the real world one, but also to guarantee that the sampled training data yield the consistent conditional output distribution, i.e. click-through rate (CTR), with the real world data. However, due to the sparseness of clicks in sponsored search, it is a bit contradictory to address these two challenges simultaneously. In this paper, we first take a theoretical analysis to reveal this sampling dilemma, followed by a thorough data analysis which demonstrates that the straightforward random sampling method may not be effective to balance these two kinds of consistency in sampling dilemma simultaneously. To address this problem, we propose a new sampling algorithm which can succeed in retaining the consistency between the sampled data and real world in terms of both input distribution and conditional output distribution. Large scale evaluations on the click-through logs from a commercial search engine demonstrate that this new sampling algorithm can effectively address the sampling dilemma. Further experiments illustrate that, by using the training data obtained by our new sampling algorithm, we can learn the model with much higher accuracy in click prediction.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval; H.3.5 [**On-line Information Services**]: Web-based services

WSDM'14, February 24–28, 2014, New York, New York, USA.

WSDM'14, February 24–28, 2014, New York, New York, USA.
http://dx.doi.org/10.1145/2556195.2556242.

General Terms

Algorithms, Experimentation, Performance

Keywords

Online Advertising, Sponsored Search, Click Prediction, Data Sampling

1. INTRODUCTION

As an online advertising system, sponsored search [8] [10] has been one of the most important business models for commercial Web search engines. It contributes most of the revenue for search engines by presenting to users sponsored search results, i.e., advertisements (ads), along with organic search results. To deliver right ads to right users, a sponsored search system consists of a couple of technical components, including query-to-ad matching [1], click prediction for matched ads [6] [9], filtration of the ads according to thresholds for relevance and click probability, and auction to determine the ranking, placement, and pricing of the remaining ads [7]. In today's industry, generalized second price auction (GSP) [7] is the most widely-used auction mechanism, in which the price that an advertiser has to pay depends on the predicted click probability of his/her own ad as well as the bid price and predicted click probability of the ad ranked in the next position. As click prediction has been widely used in the filtration, ranking, placement, and pricing of the ads, accurate prediction of click probability becomes an essential problem in sponsored search.

State-of-the-art sponsored search systems typically employ a machine learning approach to predict the probability that a user clicks on an ad. As a general machine learning process, there are a couple of key components that could influence the accuracy of click prediction, including data sampling for training, feature extraction, and model selection. Many of previous studies have spent their efforts in discovering important features for accurate click prediction. For example, as shown in previous work [6] [2] [9], the historical click information for each ad are shown to be effective for predicting the future click probability of the ad. In practical sponsored search systems, however, there are many ads without adequate historical click-through data (even after aggregation at different levels, e.g., campaign, advertiser, and query levels). To tackle this data sparseness issue, some relevance features have been investigated to improve the accuracy of click prediction, and this type of features is mostly based on the similarity between query and ad, and the quality of the ad [5] [14] [16] [15] [13] [19]. Some efforts have

also focused on model selection for click prediction. For example, many previous studies employ the maximum entropy model [15] since its strength in formulating the click probability by combining diverse forms of features, while other works apply the Bayesian framework [18] for click prediction as it is more effective to incorporate temporal and positional information. One of most recent works [17] takes advantage of continuous conditional random fields based model for click prediction which succeeds in modeling relational information between different ads.

The attention paid to feature extraction and model selection for click prediction, however, have overshadowed seemingly less obvious but essentially important challenges in terms of data sampling. A commercial search engine can obtain billions of users' impressions on sponsored search results everyday. Such huge size of real world data requires extensive resource cost involved in the storage of data as well as significant time cost spent in the training of click prediction model. Hence, it has necessitated the need for collecting a sample of sponsored search logs in order for training and updating the click prediction model both effectively and efficiently. Typically, in order to obtain precise click prediction model, effective data sampling requires that the sampled data should yield the similar input data distribution as the original whole data. Beyond that, in the context of click prediction for sponsored search, there is another important requirement for the data sampling. In particular, as the goal of click prediction in sponsored search is to predict the probability that a user clicks an ad given a query, the learning objective of click prediction model is optimizing the click probability towards the click-through rate (CTR) with respect to the same ⟨query, ad⟩ pair. To this end, it requires the sampled data yield the consistent conditional output distribution, i.e. CTR, as the original whole data. However, due to the sparseness of click in sponsored search, it is generally a bit contradictory to fulfill these two requirements simultaneously.

From one hand, some judiciously chosen sampling practice, e.g. random sampling, is widely used to obtain the sampled training data for click prediction, as it is effective to retain the distribution of each input data, i.e. the ⟨query, ad⟩ pair, in the sample data. Unfortunately, because of the sparseness of clicks in sponsored search, random sampling can cause a big difference in terms of conditional output distribution between sampled data and original whole data, which means that the CTR of one ⟨query, ad⟩ pair computed based on sampled training data is quite different from that computed based on whole real world data. For example, assuming that one ⟨query, ad⟩ pair generates totally $10,000$ impressions in the whole real world data during a certain period and 10 of these impressions cause clicks (i.e. the CTR is 0.1%), if we plan to take 1% random sampling for each ⟨query, ad⟩ pair to obtain training data, we are not able to let this pair yield the same CTR in the sampled data as in the whole data, no matter how many clicks are included in the sampled data. In this paper, a thorough data analysis will be presented to reveal how severe this problem is by using the judiciously chosen random sampling method.

On the other hand, in order to retain the ⟨query, ad⟩ pair's CTR in the sampled data, one straightforward method is to distribute different sampling ratios to various ⟨query, ad⟩ pairs based on their CTRs in the real world data. Specifically, those ⟨query, ad⟩ pairs with lower real CTRs can yield a bit higher sampling ratios while those with higher real CTRs can be sampled with lower ratios. For example, for the ⟨query, ad⟩ pair which has totally $10,000$ impressions and 10 of them are clicks, we could raise the sampling ratio to 10%. Then, it is more likely that we sample 1000 impressions with 1 click in it, which yields the same CTR to that of this pair in the whole data. This straightforward method, unfortunately, may cause that many ⟨query, ad⟩ pairs contribute very different relative volumes of traffic in the sampled data compared with those in the whole data. Consequently, the input data distributions reflected by the sampled data are quite different from the real data distributions in the whole data, which also prevents from achieving accurate click prediction.

To address this sampling dilemma between input data distribution and conditional output distribution, in this paper, we first formulate these challenges from a theoretical perspective, then we propose a new sampling algorithm which can balance the trade-off between these two factors so as to collect effective training data in order for achieving accurate click prediction. In this paper, we will conduct a large scale evaluation on the effectiveness of this new sampling algorithm based on the click-through logs from a commercial search engine. Experimental results demonstrate that, the click prediction model which is learned using the training data sampled by our new sampling algorithm can reach much better performance than the model learned using randomly sampled training data. Further experimental analysis also prove that our new sampling method is effective to balance the trade-off between keeping the similar input data distribution and keeping the consistent conditional output distribution (i.e. CTR) in the sampled data as in the real world whole data.

To sum up, the contributions of our work include:

- A comprehensive theoretical analysis to reveal the dilemma between retaining similar input data distribution and keeping consistent conditional output distribution during data sampling for learning the click prediction model.
- A thorough empirical analysis to illustrate that judiciously chosen random sampling is not effective at balancing the sampling dilemma for click prediction.
- An effective sampling algorithm for collecting training data in order for accomplishing accurate click prediction
- A large scale evaluations demonstrating that the new sampling algorithm can effectively balance the sampling dilemma and lead to the click prediction model with higher performance.

The remainder parts of this paper are organized as follows. In Section 2, we present our theoretical analysis to reveal the sampling dilemma in click prediction as well as the data analysis to illustrate the problem caused by judiciously chosen random sampling method. Section 3 formulates the problem of data sampling for click prediction and introduces our new sampling algorithm. After briefly introducing the click prediction modeling in Section 4, we present our experimental setup and results in Section 5. At last, we conclude the paper and discuss the future work in Section 6.

2. MOTIVATION AND DATA ANALYSIS

As mentioned above, click prediction in sponsored search is typically a machine learning process. In this section, we will first introduce the learning objective of click prediction and formulate the sampling dilemma for achieving accurate

click prediction from the theoretical perspective. To gain more understanding on why effective data sampling plays an important role in click prediction, we will conduct a large scale data analysis on real sponsored search data, which will demonstrate that the judiciously chosen random sampling method may cause a critical problem in addressing the sampling dilemma between keeping input data distribution and keeping conditional output distribution (i.e. CTR).

2.1 Sampling Dilemma for Click Prediction

In this subsection, we give detailed formulation of click prediction as a supervised machine learning task, and show how the data sampling strategy influence the performance of the learned click prediction model, followed by which we formulate the sampling dilemma for click prediction.

In the problem of click prediction, the input data are a set of ⟨query, ad⟩ pairs, denoted as $\{\langle q_i, a_i\rangle; i = 1, 2, \cdots, n\}$, and each $\langle q_i, a_i\rangle$ can be represented as a vector of features, denoted as $\{x_i = \phi(q_i, a_i) \in \mathcal{X}; i = 1, 2, \cdots, n\}$. In general, the feature space $\mathcal{X}$ is infinite. In this work, to describe the problem more clearly, we assume that $\mathcal{X}$ is finite, i.e. $\mathcal{X} = \{z_1, z_2, \cdots, z_K\}$, with an underlying probability $P : p_k; k = 1, ..., K$ where K is the size of feature space $\mathcal{X}$. Note that our description below can be naturally extended to the infinite feature space. The output consists of clicks or non-clicks on ads, denoted as $\{c_i \in \{0, 1\}; i = 1, \cdots, n\}$, which is generated from the underlying conditional probability $P(c = 1|x_i)$.

As the size of sampled data for learning click prediction is fairly large, we assume that, for each element in the ⟨query, ad⟩ space, there will be at least one sample. Thus, we re-form the data as follows: $\{(x_k, c_{k,j}); k = 1, \cdots, K; j = 1, \cdots, n_k\}$. We denote the frequency of a ⟨query, ad⟩ pair x_k as $\hat{p}_k = \frac{n_k}{\sum_{k=1}^{K} n_k}$ and the empirical CTR for ad x_k as $\hat{\text{CTR}}(x_k) = \frac{\sum_{j=1}^{n_k} c_{k,j}}{n_k}$.

Our objective is to learn a click prediction model $f : \mathcal{X} \to [0, 1]$ which can minimize the following expected loss:

$$\mathbb{E}L(f) = \sum_{k=1}^{K} p_k \left| f(x_k) - P(c = 1|x_k) \right|. \quad (1)$$

However, the underlying distributions p_k and $P(c = 1|x_k)$ are unknown. Therefore, after conducting data sampling, we can just optimize the empirical loss on the sampled data set D.

$$\hat{L}(f; D) = \sum_{k=1}^{K} \hat{p}_k \left| f(x_k) - \hat{\text{CTR}}(x_k) \right|. \quad (2)$$

To further investigate the influence of the sampling, we describe the distance between the empirical distribution and underlying distribution in terms of *input distribution* and *conditional output distribution*. To be specific, we denote the distance in terms of *input distribution* as

$$\eta_D = \max_k |\hat{p}_k(D) - p_k| \quad (3)$$

and the distance in terms of *conditional output distribution* as

$$\gamma_D = \max_k |\hat{\text{CTR}}(x_k; D) - P(c = 1|x_k)|. \quad (4)$$

Note that the distance in terms of *conditional output distribution* implies that in terms of CTR. Then, it is easy to have the following proposition:

PROPOSITION 1. *Consider data set D, for $\forall f \in \mathcal{F}$, we have*

$$\mathbb{E}L(f) \leq \left(1 + \frac{\eta_D}{\min_k n_k(D)}\right) \hat{L}(f; D) + K\left(\eta_D + \gamma_D + \eta_D\gamma_D\right).$$

This proposition explains how the distance between empirical distribution of the sampled data and the underlying distribution influence the distance between empirical loss and the expected loss. When the size of sampled data approaches infinity, all the distance vanish, and empirical loss approaches expected loss; however, if the total sampling number is limited, in order to minimize the empirical loss, there is a trade-off between η and γ, which indicates a dilemma between the *input distribution* and the *conditional output distribution*. We formulate it as a *Sampling Dilemma* for click prediction:

> **Sampling Dilemma**: *In the task of sampling a collection of data for learning click prediction, if the sample size is limited, in order to minimize the distance between empirical loss and the expected loss, smaller distance between the empirical distribution and underlying distribution in terms of input distribution, i.e. η, indicates higher distance between the empirical distribution and underlying distribution in terms of conditional output distribution, i.e. γ, and vice versa.*

Previous studies usually rely on some judiciously chosen sampling practice, e.g. random sampling. This method is easy to ensure a low distance between the empirical distribution and underlying distribution in terms of input distribution, which, unfortunately, indicates that this method may cause a big problem in retaining the consistent conditional output distribution, i.e. CTR, between the sampled data and the real world. In the following of this section, we will provide a large scale data analysis to demonstrate that random sampling could give rise to a big difference between the CTRs computed based on sampled data and those computed based on the whole real data.

2.2 Data Settings

All of our data used in this work are collected from a commercial search engine. In this analysis, we first collect the whole data from the sponsored search during one week. Then, based on this whole data, we conduct random sampling to generate a sample data.

Dataset 1 - Whole data: In order to obtain the real CTR for each ⟨query, ad⟩ pair, we collect the entire set of queries with all the ads displayed under them from the sponsored search logs during one week in July, 2012. Then, we compute the CTR of each ⟨query, ad⟩ pair based on its corresponding impressions and clicks in this whole data. To ensure the reliability of obtained CTR, we filter those ⟨query, ad⟩ pairs with less than $\mathcal{N}$ impressions. According to the results of cross-validation, we set $\mathcal{N}$ as 50 in this paper. Finally, we collect about 184M queries and the total number of impressions of all associated ⟨query, ad⟩ pairs is about 3.5B.

Dataset 2 - Random sampled data: In order to investigate if random sampling can retain the real CTR in the sample data, we take a random sampling on the **Whole data** with ratio 0.5%. Thus, the total number of impressions under this sample data is 0.5% of that under the **Whole data**. Finally, we collect 2M queries and the total number of impressions of all associated ⟨query, ad⟩ pairs in this sample

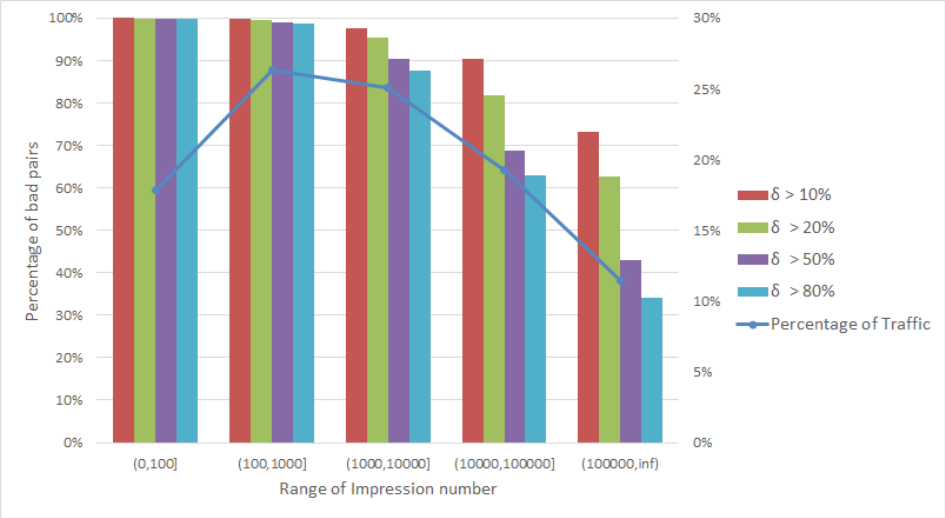

Figure 1: Respective percentage of *bad* pairs over all unique ⟨query, ad⟩ pairs in the Random sample data, where the threshold of *bad* pairs is set with multiple values. All unique ⟨query, ad⟩ pairs are separated into several bucket according to their impressions. We also illustrate the respective percentage of traffic for each bucket, shown as the line in this figure.

data is about 20.9M. We also compute CTR for each ⟨query, ad⟩ pair based on its corresponding impressions and clicks this sample data. In the next subsection, we will show that whether such CTRs are quite different with those computed based on the **Whole data**.

2.3 Inconsistency of CTR Caused by Random Sampling

In sponsored search, one of important characteristics is the extreme sparseness in terms of clicks, which makes click prediction in sponsored search be quite different with traditional machine learning process. Consequently, random data sampling may cause that the CTR of one ⟨query, ad⟩ pair computed based on randomly sampled data is quite different from that computed based on whole real world data, especially when the number of clicks on this pair is relatively very small. For example, assuming that one ⟨query, ad⟩ pair yields totally 100,000 impressions in the **Whole data** and 10 of these impressions cause clicks. Thus, the CTR of this pair in the **Whole data** is 0.1%. However, after taking random sampling with ratio 0.1%, the CTR of this pair in the **Random sample data** could only be one of {0,, 1%, 2%, $\cdots$, 10% }, which cannot be consistent with the CTR in the **Whole data**.

To measure the CTR's changing for one one ⟨query, ad⟩ pair, we compute the relative difference:

$$\Delta_{\langle q_i, a_i \rangle} = \frac{|\mathrm{CTR}_i^{\mathbb{W}} - \mathrm{CTR}_i^{\mathbb{S}}|}{\mathrm{CTR}_i^{\mathbb{W}}}$$

where $\mathrm{CTR}_i^{\mathbb{W}}$ represents the CTR of $\langle q_i, a_i \rangle$ computed based on its impressions and clicks in the **Whole data**, and $\mathrm{CTR}_i^{\mathbb{S}}$ represents the CTR of $\langle q_i, a_i \rangle$ computed based on its impressions and clicks in the **Random sample data**. If such relative difference, $\Delta_{\langle q_i, a_i \rangle}$, is larger than a threshold, δ, we call this pair a *bad* one caused by random sampling.

To investigate how random sampling will result in inconsistency of CTR, we conduct experiments to see how many *bad* pairs will be generated in the **Random sample data**. Note that the CTR of any ⟨query, ad⟩ pair, which has no click in the *Whole data*, will remain 0 in both **Whole data** and **Random sample data**. Accordingly, in this experiment, we filter out those ⟨query, ad⟩ pairs with no click in the **Whole data**. Figure 1 demonstrates the percentage of *bad* pairs over all unique ⟨query, ad⟩ pairs in the **Random sample data**. To examine the effects of varying number of impressions in the **Whole data** of different pairs, we separate all ⟨query, ad⟩ pairs into several buckets based on their impressions in the **Whole data** and compute the respective percentage of *bad* pairs. In addition, we compare the respective percentage of *bad* pairs against different settings of the threshold of the *bad* pair, i.e. δ.

From this figure, we can find that, random sampling can result in a significant inconsistency of CTR for those ⟨query, ad⟩ pairs with relatively small impressions in the **Whole data**. For example, no matter how to set the threshold of the *bad* pair, i.e. δ, the percentage of bad pairs reaches more than 95% for those pairs who have less than 1,000 impressions in the **Whole data**. And, from the figure, we can find that all of these pairs comprise of about 45% of the whole traffic. For those ⟨query, ad⟩ pairs having more impressions, i.e. larger than 1,000, the percentage of bad pairs declines with increasing number of impressions in the **Whole data**. Unfortunately, for those ⟨query, ad⟩ pairs with more than 100,000 impressions, random sampling still generate a large portion of bad pairs even the threshold δ is set with a high value. For example, when setting the threshold δ as 50%, there are still more than 40% bad pairs in the **Random sample data** caused by random sampling.

As the predicted click probability is usually used to help rank ads in the sponsored search results, we conduct further experiments to examine if random sampling can retain the order of CTRs for ⟨query, ad⟩ pairs under the same query. Specifically, for one query q_i, we compute the respective CTR for each pair associated with this query in the **Whole data**, and rank all ads associated with this query based on their respective CTRs. Then, we compute another ranking of ads under the same query based on their respective CTRs obtained from the **Random sample data**. After that, for

each query, we compare these two rankings to see whether they have overlaps among the top ranked ads. Figure 2 demonstrates the percentage of queries which have different top-k rankings based on these two ranking results. From this figure, we can find that random sampling can result in a big difference in terms of the CTR orders under the same query.

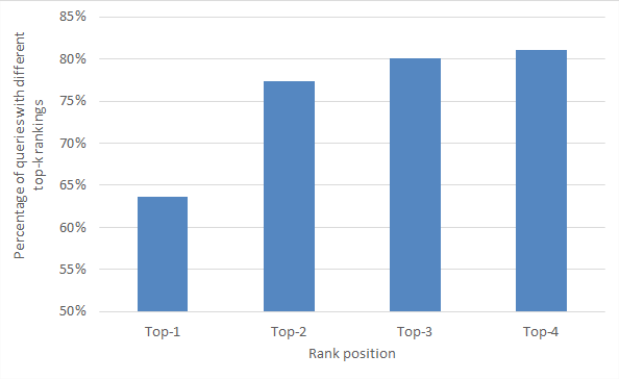

Figure 2: Percentage of queries which have different top-k rankings based on CTRs computed by the Whole data and the Random sample data, respectively.

Based on all these studies, we can conclude that random sampling is likely to cause significant inconsistency of CTR for ⟨query, ad⟩ pairs, which implies a big influence on the learning objective for click prediction. Therefore, it is quite difficult to obtain an accurate click prediction model based on the training data generated via random sampling. To address this challenge, in the next section, we will formulate the sampling for click prediction as an optimization problem and further propose a new sampling algorithm which succeeds in balancing the difference in terms of input distribution and that in terms of conditional output distribution (i.e. CTR).

3. BALANCED SAMPLING FOR CLICK PREDICTION

As a general machine learning process, click prediction necessitates the need for sampling sponsored search data in order for training and updating the click prediction model in an effective and efficient way. However, as discussed in Section 2, judiciously chosen sampling method, especially random sampling, fails to address the *sampling dilemma*. Specifically, while random sampling can keep the similar input distribution between the whole data and the sampled data, it leads to inconsistency of the conditional output distribution, i.e. CTR, for click prediction. In this section, we will formulate this task as an optimization problem, the goal of which is to conduct data sampling with balancing the two aspects in the *sampling dilemma*, one of which is minimizing the distance between the whole data and the sampled data in terms of *input distribution*, while the other of which is minimizing the distance between the whole data and the sampled data in terms of *conditional output distribution* (i.e. CTR). Then, we propose a new balanced sampling algorithm which can achieve effective data sampling with addressing the two constraints in this problem.

3.1 Problem Statement

The general problem of data sampling for learning click prediction model in sponsored search can be formulated as follows.

First, we are given an entire collection of sponsored search logs during a certain period $(t_{\text{start}}, t_{end})$, which is defined as the *full set*.

DEFINITION 1. **Full set:** *a full list of tuples,*

$$\mathcal{F} = \{\langle q_i, a_i, t_i, c_i\rangle | i = 1, 2, \cdots\}$$

where q, a, and t represent one user submitted query q to the search engine and see a presented ad a at time t ($t \in (t_{\text{start}}, t_{end})$), and c is set as 1 if the user click the ad a or 0 otherwise.

Based on this definition of *full set*, we can compute the *full relative traffic* and the *full CTR* for each $\langle q, a\rangle$ included in the *full set*.

DEFINITION 2. **Full relative traffic:** *for each $\langle q, a\rangle$ included in the full set, assuming the whole list of its associated tuples in the full set is*

$$L^{\text{f}}_{\langle q,a\rangle} = \{\langle q, a, t_i, c_i\rangle | i = 1, 2, \cdots \wedge \langle q, a, t_i, c_i\rangle \in \mathcal{F}\}$$

its full relative traffic is defined as

$$\text{TF}^{\text{f}}_{\langle q,a\rangle} = \frac{|L^{\text{f}}_{\langle q,a\rangle}|}{|\mathcal{F}|}$$

DEFINITION 3. **Full CTR:** *for each $\langle q, a\rangle$ included in the full set, assuming the whole list of its associated tuples in the full set is $L^{\text{f}}_{\langle q,a\rangle}$ defined as above, its full CTR is computed as*

$$\text{CTR}^{\text{f}}_{\langle q,a\rangle} = \frac{|\{\langle q, a, t_i, c_i\rangle | \langle q, a, t_i, c_i\rangle \in L^{\text{f}}_{\langle q,a\rangle} \wedge c_i = 1\}|}{|L^{\text{f}}_{\langle q,a\rangle}|}$$

Our goal for data sampling is to collect a subset of the *full set* with a certain sample ratio ρ, which is defined as the *sample set*.

DEFINITION 4. **Sample set:** *a list of tuples which comprise of a subset of the full set,*

$$\mathcal{S} = \{\langle q_j, a_j, t_j, c_j\rangle | j = 1, 2, \cdots \wedge \langle q_j, a_j, t_j, c_j\rangle \in \mathcal{F}\}$$

the sample ratio of which, ρ, is defined as $\rho = \frac{|\mathcal{S}|}{|\mathcal{F}|}$

Based on this definition of *sample set*, we can compute the *sample relative traffic* and the *sample CTR* for each $\langle q, a\rangle$ included in the *sample set*.

DEFINITION 5. **Sample relative traffic:** *for each $\langle q, a\rangle$ included in the full set, assuming the whole list of its associated tuples in the sample set is*

$$L^{\text{s}}_{\langle q,a\rangle} = \{\langle q, a, t_j, c_j\rangle | j = 1, 2, \cdots \wedge \langle q, a, t_j, c_j\rangle \in \mathcal{S}\}$$

its sample relative traffic is defined as

$$\text{TF}^{\text{s}}_{\langle q,a\rangle} = \frac{|L^{\text{s}}_{\langle q,a\rangle}|}{|\mathcal{S}|}$$

DEFINITION 6. **Sample CTR:** *for each $\langle q, a\rangle$ included in the sample set, assuming the whole list of its associated tuples in the sample set is $L^{\text{s}}_{\langle q,a\rangle}$ defined as above, its sample CTR is computed as*

$$\text{CTR}^{\text{s}}_{\langle q,a\rangle} = \frac{|\{\langle q, a, t_j, c_j\rangle | \langle q, a, t_j, c_j\rangle \in L^{\text{s}}_{\langle q,a\rangle} \wedge c_j = 1\}|}{|L^{\text{s}}_{\langle q,a\rangle}|}$$

We now state our problem of more formally:

Problem: (*Data sampling for learning click prediction model*) Given a *full set* $\mathcal{F}$ and a sample ratio ρ, generate a *sample set* $\mathcal{S}$ which yields $|\mathcal{S}| = \rho|\mathcal{F}|$ and can minimize both the aggregated difference between *full relative traffic* and *sample relative traffic* of each $\langle q, a\rangle$ and the aggregated difference between aggregated *full CTR* and *sample CTR* of each $\langle q, a\rangle$.

We can formalize this problem into an optimization problem with the objective:

$$\min_{\mathcal{S}\subset\mathcal{F}\wedge|\mathcal{S}|=\rho|\mathcal{F}|} \sum_{\langle q,a\rangle} |\mathrm{CTR}^{\mathrm{s}}_{\langle q,a\rangle} - \mathrm{CTR}^{\mathrm{f}}_{\langle q,a\rangle}| + \lambda|\mathrm{TF}^{\mathrm{s}}_{\langle q,a\rangle} - \mathrm{TF}^{\mathrm{f}}_{\langle q,a\rangle}| \tag{5}$$

where $|\mathrm{CTR}^{\mathrm{s}}_{\langle q,a\rangle} - \mathrm{CTR}^{\mathrm{f}}_{\langle q,a\rangle}|$ corresponds to the distance between the whole data and sampled data in terms of *conditional output distribution*, while $|\mathrm{TF}^{\mathrm{s}}_{\langle q,a\rangle} - \mathrm{TF}^{\mathrm{f}}_{\langle q,a\rangle}|$ corresponds to the distance between the whole data and sampled data in terms of *input distribution*, and λ can be used to control this trade-off. This data sampling problem can be naturally viewed as a subset selection problem, which is typically NP-complete. Hence, in this paper, we propose a greedy balanced algorithm to address this problem.

3.2 Balanced Sampling Algorithm

Despite that there is one given sampling ratio for the whole sample set, we can greedily apply different sampling ratio to different $\langle q, a\rangle$ pairs, which is decided based on the corresponding CTR and number of impressions for each $\langle q, a\rangle$ pair.

Assuming that there are N unique $\langle q, a\rangle$ pairs in the full set $\mathcal{F}$,

$$\{\langle q_1, a_1\rangle, \langle q_2, a_2\rangle, \cdots, \langle q_N, a_N\rangle\},$$

the *full CTRs* of them are respectively denoted as

$$\{\mathrm{CTR}^{\mathrm{f}}_1, \mathrm{CTR}^{\mathrm{f}}_2, \cdots, \mathrm{CTR}^{\mathrm{f}}_N\},$$

and the numbers of impressions for each pair are denoted as

$$\{\mathrm{Imp}^{\mathrm{f}}_1, \mathrm{Imp}^{\mathrm{f}}_2, \cdots, \mathrm{Imp}^{\mathrm{f}}_N\}.$$

In our new sampling algorithm, we will decide the respective number of impressions we target to sample and include in the sample set for each $\langle q, a\rangle$ pair, which are denoted as

$$\{\mu_1, \mu_2, \cdots, \mu_N\},$$

where, given a specific sampling ratio, ρ, the sum of the impressions we target to sample for each pair should be subject to $\sum_{i=1}^{N} \mu_i \leqslant \rho|\mathcal{F}|$. Then, we propose our balanced algorithm, as shown in Alg 1, to decide $\{\mu_1, \mu_2, \cdots, \mu_N\}$. After that, we could apply random sampling to each $\langle q_i, a_i\rangle$ to sample the associated tuples according to the sample ratio of $\frac{\mu_i}{\mathrm{Imp}^{\mathrm{f}}_i}$.

In this algorithm, α and β are used together to trade-off the consistency between the input distribution and conditional output distribution. Specifically, if we set β as a lower value, this algorithm requires more $\langle q, a\rangle$ pairs keep the consistent CTR, which indicates more emphasis on conditional output distribution. However, to satisfy the limited size of sample set, α has to be set with a higher value. On the other hand, if we set α as a lower value, which implies more emphasis on consistent input distribution, this algorithm requires us increase β to satisfy the limited size of sample set. Therefore, α and β collaboratively play the role of λ in Eq. 5.

Algorithm 1 Balanced Data Sampling Algorithm for Learning Click Prediction Model

Input:
- $\mathcal{F}$: the full set with N unique $\langle q, a\rangle$ pairs, i.e. $\{\langle q_1, a_1\rangle, \langle q_2, a_2\rangle, \cdots, \langle q_N, a_n\rangle\}$
- $\{\mathrm{CTR}^{\mathrm{f}}_1, \mathrm{CTR}^{\mathrm{f}}_2, \cdots, \mathrm{CTR}^{\mathrm{f}}_N\}$: the full CTRs of each $\langle q, a\rangle$ pair
- $\{\mathrm{Imp}^{\mathrm{f}}_1, \mathrm{Imp}^{\mathrm{f}}_2, \cdots, \mathrm{Imp}^{\mathrm{f}}_N\}$: the number of full impressions of each $\langle q, a\rangle$ pair
- $K = \rho|\mathcal{F}|$: the upper bound of the size of sample set.
- α: the threshold for retaining input distribution.
- β: the threshold of the least precision of CTR, which is used for retaining conditional output distribution (i.e. CTR).

Output:
- $\{\mu_1, \mu_2, \cdots, \mu_N\}$: the number of impressions we target to sample and include in the sample set for each $\langle q, a\rangle$ pair

Algorithms:

1 Sort all the $\langle q, a\rangle$ pairs based on their full impression number in descending order.

2 To retain the full CTR, we should ensure that the sampled impression is large enough to include at least one click for each $\langle q_i, a_i\rangle$ pair. Thus, we compute the minimum of sampled impression, denoted as π_i, for each $\langle q_i, a_i\rangle$ pair as:

$$\pi_i = \begin{cases} \frac{1}{\mathrm{CTR}^{\mathrm{f}}_i}, & \text{if } \mathrm{CTR}^{\mathrm{f}}_i \geq \beta \\ 0, & \text{if } \mathrm{CTR}^{\mathrm{f}}_i < \beta \end{cases}$$

3 Initiate $\{\mu_1, \mu_2, \cdots, \mu_N\}$ as

$$\mu_i = \rho \cdot \mathrm{Imp}^{\mathrm{f}}_i$$

4 Scan the sorted list of all $\langle q, a\rangle$ pairs one by one, for each one (e.g $\langle q_i, a_i\rangle$):
- if $\mu_i > \pi_i$, μ_i has already met the condition and will not changed at this time
- else
  for $j = 1$ **to** $i - 1$ **:**
    while$(\frac{\mathrm{Imp}^{\mathrm{f}}_j}{\mathrm{Imp}^{\mathrm{f}}_{j+1}} - \frac{\mu_j}{\mu_{j+1}} < \alpha \wedge \mu_j > \pi_j \wedge \mu_i < \pi_i)$:
      $\mu_i = \mu_i + 1$
      $\mu_j = \mu_j - 1$
    end while;
    if $\mu_i > \pi_i$, **break;**
  end for;

The time complexity of this algorithm is $O(N^2)$ where N is the number of data rows. But, most of the time, the algorithm is efficient and takes less than $O(N^2)$ because not all the $\langle q, a\rangle$ need to be adjusted.

4. CLICK PREDICTION MODELING

After sampling the training data, in this section, we will select a specific machine learning approach for click prediction modeling. As discussed above, given a query q and an ad a, the problem of click prediction is to compute the probability of click $p(c|q, a)$. Therefore, the maximum entropy model [3] is well suited for this learning task since its

strength in combining diverse forms of contextual information, and formulates the click probability for a ⟨query, ad⟩ pair as follows:

$$p(c|q,a) = \frac{1}{1+\exp(\sum_{j=1}^{d}\omega_j f_j(q,a))}$$

where $f_j(q,a)$ represents the j-th feature derived for ⟨query, ad⟩ pair (q,a) and $\omega_j \in \mathbf{w}$ is the associated weight. Given the training set $\mathcal{D}_{\text{train}}$, the maximum entropy model learns the weight vector $\mathbf{w}$ by maximizing the likelihood of exponential models as:

$$\mathbf{w} = \max(\sum_{i=1}^{n}\log(p(c_i|q_i,a_i)) + \log(p(\mathbf{w})))$$

where the first part represents the likelihood function and the second part utilizes a Gaussian prior on the weight vector $\mathbf{w}$ to smooth the maximum entropy model. There are many approaches available in the literature [11] to solve this kind of optimization problems including iterative scaling and its variants, quasi-Newton algorithms, and conjugate gradient ascent. Given the large collection of samples and high dimensional feature space, we use a nonlinear conjugate gradient algorithm [12].

An accurate maximum entropy model relies greatly on the design of features. According to the state-of-the-art works in click prediction, there are majorly two kinds of features, which are relevance features and historical click features. In this work, we use some representative features according to previous work [4] [15]:

- For relevance features we employed edit distance of ad and query, edit distance of ad and bid keyword, cosine similarity between ad and query, the category matching between ad and query, etc.
- For historical features we employed history COEC (position normalized CTR) for ⟨query, ad⟩ pair, query, and ad, respectively, smoothed COEC according to query term, ad term, etc.

5. EXPERIMENTS

In this section, we first describe the settings of our experiments and then report the experimental results.

5.1 Experimental Settings

5.1.1 Data set

In our experiment, we would like to examine if our proposed balanced sampling algorithm can generate the training data which is more effective to learn the click prediction model with higher accuracy in sponsored search. We use the click through logs of a commercial search engine during a half month period in July 2012 as our experimental dataset. We divide this dataset into two parts, each of which contains the data of one week. Then, we use the first week's data as the full set. Note that, this full set is the **Whole data** we use for data analysis in Section 2.2. To learn the click prediction model, we will sample the training set from this full set. In our experiments, we first randomly sample query events from the original traffic as the baseline random sampled training set. Specifically, we randomly sample about 0.5% ad impressions from logs of the first week. Note that, this random sampled set is the **Random sampled data** we use for data analysis in Section 2.2. Then, we employ our proposed new balanced sampling algorithm to sample a new training set. Detailed statistics of these two training datasets can also be found in Table 1. After collecting training set, we use the second week's data to test the performance of the model. Table 1 also shows the detailed statistics of this test data set.

5.1.2 Compared Methods

As mentioned in Section 4, we employ maximum entropy modeling to training the click prediction model. In our experiments, we will use different sampling methods to obtain different training sets. Then, we will compare the performance of different click prediction models learned based on their respective training sets. Specifically, we compare the following click prediction models:

- **Random**: We use Random Sampling method to collect the training set and apply this data to learn the click prediction model.
- **Balanced**: We employ our proposed Balanced Sampling algorithm to obtain the training set and use this data to learn the click prediction model.

5.1.3 Parameters Setting

As discussed in Section 3.2, there are two parameters to set in our new Balanced Sampling algorithm, including the threshold for retaining the input distribution, i.e. α, and the threshold for keeping the CTR precision, i.e. β. As introduced before, these two parameters are used to balance the trade-off between the consistence in terms of input distribution and that in terms of CTR. Therefore, before conducting the further evaluations, we run a cross-validation to find the best setting for these two parameters. According to the results of cross validation, we set the best parameter values as $\alpha = 0.001$ and $\beta = 0.005$. And, we will take more detailed discussions about the effects of different parameters settings in the experimental results.

5.1.4 Evaluation Metrics

In our work, the Maximum Entropy modeling is applied to predict click probability for every ad impression. We use recorded user actions, i.e. click or non-click, in the log data as labels. To evaluate the overall performance for the model, we employ average Relative Information Gain(RIG) [9] as the evaluation metric. The RIG is calculated as:

$$\text{RIG} = \frac{\text{CE} + \mathcal{H}(\tilde{p})}{\mathcal{H}(\tilde{p})}$$

where CE represents the empirical cross entropy and is computed as

$$\text{CE} = \frac{1}{N}\sum_i y_i \log\hat{p} + (1-y_i)\log(1-\hat{p})$$

where N is the total number of ⟨query, ad⟩ pairs in the testing set, $y_i = 1$ if the i-th pair is labeled as *click* otherwise $y_i = 0$, $\hat{p}$ denotes the probability of click predicted by the model; and, $\mathcal{H}(\tilde{p})$ represents the entropy of CTR, which is calculated as

$$\mathcal{H}(\tilde{p}) = -(\tilde{p}\log\tilde{p} + (1-\tilde{p})\log(1-\tilde{p}))$$

where $\tilde{p}$ denotes the total CTR and is computed as $\tilde{p} = \frac{1}{N}\sum_i y_i$. Since $\mathcal{H}(\tilde{p})$ is the maximally attainable value of information gain, i.e. $\text{CE} + \mathcal{H}(\tilde{p})$, it quantifies the information gain relative to the source entropy.

Table 1: Statistics of the datasets.

	Total impressions	# of unique ad	# of unique query	# of unique $\langle q, a \rangle$ pair
All 1st week data	3.51B	39.14M	183.9M	1.20B
All 2nd week data	3.32B	38.79M	176.1M	1.14B
Random sampled set	20.9M	0.23M	1.04M	10.8M
Balanced sampled set	20.8M	0.21M	0.68M	8.3M

5.2 Experimental Results

In the following of this section, we will present several large-scale experiments. These experiments are used to demonstrate that (1) our new Balanced Sampling method succeeds in balancing the consistent between the sampled data and the whole data in terms of both input distribution and conditional output distribution (i.e CTR); (2) the accuracy of click prediction in sponsored search can be significantly improved by using the new Balanced Sampling method to generate the training set; (3) the new Balanced Sampling method can improve the click prediction model's robustness to the limited sampling ratio.

5.2.1 Consistency of CTR

As discussed in Section 2.3, one of the most important problem caused by random sampling is the inconsistency between the whole data and the sampled data in terms of CTR. We first perform experiments to examine whether our new Balanced Sampling algorithm can address this problem. In particular, we use the same **Whole data** as described in Section 2.2, based on which we apply our new Balanced Sampling algorithm to obtain a new sample data with the same sampling ratio 0.5%. Thus, this new sample data, denoted as **New sample data**, yields the similar total impressions compared with the old **Random sample data** described in Section 2.2.

Similar to the analysis in Section 2.3, we investigate how many *bad* pairs will be generated in the **New sample data**. Figure 3 demonstrates the percentage of *bad* pairs over all unique $\langle$query, ad$\rangle$ pairs in the **New sample data**. Similar to our previous analysis, we separate all $\langle$query, ad$\rangle$ pairs into several buckets based on their impressions in the **Whole data** and compute the respective percentage of *bad* pairs. In addition, we compare the respective percentage of bad pairs against different settings of the threshold, i.e. δ.

Based on the comparison between Figure 3 and Figure 1, we can find that our new sampling method can significantly alleviate the inconsistency between whole data and sample data in terms of CTR. In particular, despite of various settings of δ, there is nearly zero percentage of *bad* ones for those $\langle$query, ad$\rangle$ pairs who have less than $1,000$ impressions or more than $100,000$ impressions in the **Whole data**. And, even for those pairs yielding impressions between $1,000$ and $100,000$ in the **Whole data**, there is a substantial reduction in the percentage of *bad* ones. For example, if we pay attention to $\langle$query, ad$\rangle$ pairs having impressions between $10,000$ and $100,000$, when setting δ as 50%, the percentage of *bad* pairs caused by our new sampling algorithm is only about 5%, which is a more than ten times reduction compared with about 68% of *bad* pairs caused by old random sampling. Note that, if δ is set as 10%, the percentage of *bad* pairs can still reach around 20% to 40% for those pairs having impressions between $1,000$ and $100,000$ in the **Whole data**. It indicates that our new sampling algorithm sacrifices a little consistency in terms of CTR to retain the consistency in terms of input distribution over all $\langle$query, ad$\rangle$ pairs.

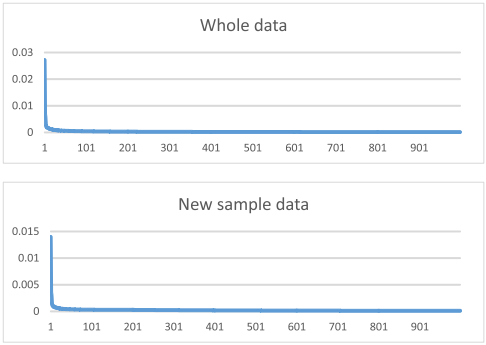

Figure 4: Relative impressions of the top 1000 pairs in both Whole data and New sample data.

5.2.2 Consistency of Input Distribution

Besides the consistency of CTR, we conduct further experiment to examine whether our new sampling algorithm can retain the consistent input distribution between the whole data and the sampled data. In particular, we first rank all $\langle$query, ad$\rangle$ pairs in the descending order based on their impressions in the **Whole data**. Figure 4 illustrates the relative impressions of the top 1000 pairs. Similarly, we rank all $\langle$query, ad$\rangle$ pairs in the descending order based on their impressions in the **New sample data** and plot top 1000 pairs' relative impressions in Figure 4. Since our algorithm has constrained that the relative difference between two consecutive pairs should be less than a threshold, i.e. $\frac{\text{Imp}^{\text{f}}_j}{\text{Imp}^{\text{f}}_{j+1}} - \frac{\mu_j}{\mu_{j+1}} < \alpha$, these two figures have the same set of 1000 pairs which also yield the same ordering in both figures. From these two figures, we can observe that the new sample data have the similar input data distribution as the whole data. We also demonstrate the relative data size of several top pairs in both **Whole data** and **New sample data** in Table 2. From this table, we can observe that our new sampling algorithm sacrifices the consistency in terms of input data distribution on those $\langle$query, ad$\rangle$ pairs with very high impressions, but, fortunately, their CTRs in the new sample data will be remained consistent with that in the whole data since there are still enough impressions of them in the new sample data. On the other hand, we can also find that, more than 99% of $\langle$query, ad$\rangle$ pairs, e.g. those ranked beyond 200, still yield the consistent input data distribution.

5.2.3 Performance of Click Prediction

After using Random Sampling method and our proposed Balanced Sampling algorithm to collect the training sets, respectively, we compare the performance of the two click prediction models trained by these two data sets separately. Table 3 demonstrates the RIG of **Balanced** model and **Random** model. From this table, we can find that our new Balanced Sampling algorithm can lead to more accurate click prediction model over the baseline random sampling. In particular, in terms of RIG, there is about 5.8% relative

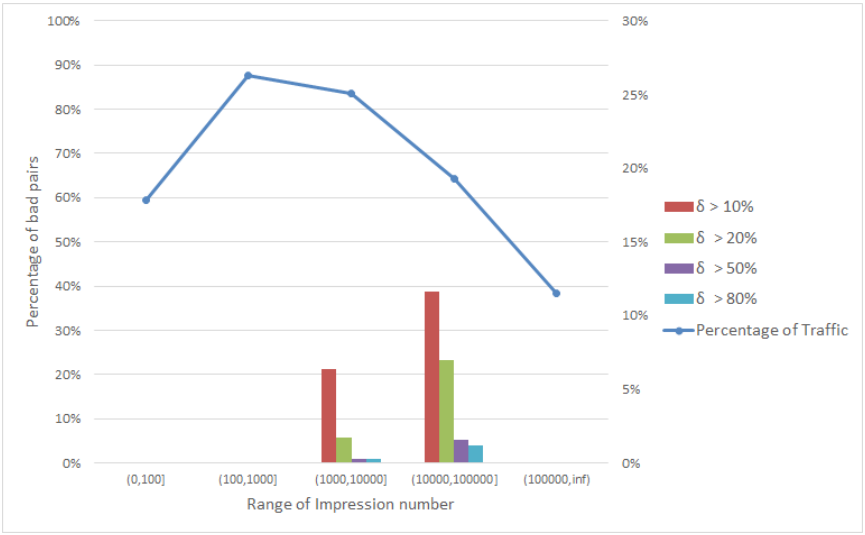

Figure 3: Respective percentage of *bad* pairs over all unique ⟨query, ad⟩ pairs in the New sample data, where the threshold of *bad* pairs is set with multiple values. All unique ⟨query, ad⟩ pairs are separated into several bucket according to their impressions. We also illustrate the respective percentage of traffic for each bucket, shown as the line in this figure.

Table 2: Relative traffic size of the top ⟨query, ad⟩ pairs in both whole data and new sample data.

⟨query, ad⟩ index	Whole data	New sample data
1	2.72%	1.40%
2	0.70%	0.36%
3	0.59%	0.31%
4	0.27%	0.15%
5	0.22%	0.12%
...	...	...
101	0.0407%	0.0313%
102	0.0406%	0.0313%
103	0.0404%	0.0309%
104	0.0403%	0.0309%
105	0.0403%	0.0309%
...	...	...
201	0.0278%	0.0276%
202	0.0276%	0.0275%
203	0.0276%	0.0275%
204	0.0275%	0.0275%
205	0.0275%	0.0275%
...	...	...

improvement when using the Balanced Sample algorithm to obtain training set rather than the Random Sampling . Actually, in real sponsor search system, increasing 1% on the click prediction accuracy is already a big improvement. According to [7], 1% accuracy improvement will drive additional hundreds of million revenue per month. In this sense, 5.8% relative improvement is really significant in click prediction.

Table 3: Performance of click prediction in terms of RIG by using Balanced Sampling compared with using Random Sampling.

Sampling Method	RIG	Relative Gain
Balanced	0.1516	+5.8%
Random	0.1434	-

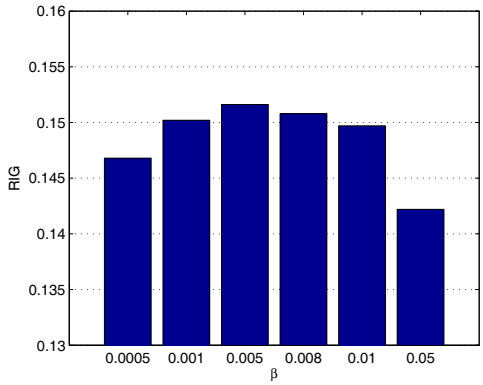

Figure 5: RIG of the Balanced model against different β during sampling.

5.2.4 *Effects of Trade-off Between CTR and Traffic Distribution*

In this experiment, we examine the effects of different settings of the parameter β, i.e. the least precision of CTR in Balanced Sampling, on the final performance of click prediction. Specifically, we conduct a comparison study by varying the value of β. Note that, under each setting of β, we are able to find a best value for the threshold of keeping input distribution, i.e. α, to optimize the performance of click prediction. Due to the limited size of sample set, the best setting of α will decreases with increasing value of β, and vice versa.

Figure 5 reports the performance of click prediction by using the **Balanced** model with varying β during sampling. From this figure, we can find that, it is unable to achieve good performance of click prediction either when β reaches a high value or when β is set as a very low one. The higher β implies that we pay more attention to keep the consistency of CTR even for those ⟨query, ad⟩ pairs without enough impressions. Therefore, it is necessary to distribute more sampling

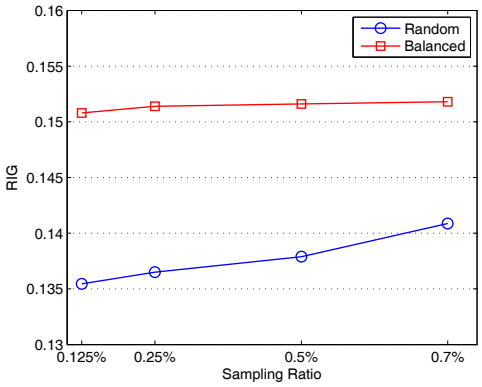

Figure 6: RIG of the Balanced model compared with the Random model against different sampling ratio.

quotes from high impressions ⟨query, ad⟩ pairs to those with low impressions, which will lead to a big difference in terms of input distribution between the whole data and the sample data. On the other hand, small β indicates that we can obtain the sample data with consistent input distribution compared to the whole data. However, it will give rise to more ⟨query, ad⟩ pairs with inconsistent CTR between the whole data and the sample data. According to our definition in Eq. 5, higher β leads to smaller $\sum_{\langle q,a\rangle} |\mathrm{CTR}^{\mathrm{s}}_{\langle q,a\rangle} - \mathrm{CTR}^{\mathrm{f}}_{\langle q,a\rangle}|$ but higher $\sum_{\langle q,a\rangle} |\mathrm{TF}^{\mathrm{s}}_{\langle q,a\rangle} - \mathrm{TF}^{\mathrm{f}}_{\langle q,a\rangle}|$, and vice versa. Therefore, the setting of the parameters β and α implies our trade-off between these two factors in sampling dilemma.

5.2.5 *Effects of Sampling Ratio*

We now explore the influence of the sampling ratio on the effectiveness of the sampled data. Figure 6 demonstrates the RIG of the **Balanced** model compared with the **Random** model against different sampling ratio ρ. Under each sampling ratio, we leverage cross validation to find the best parameters setting of **Balanced** model. From this figure, we can see that RIG of **Random** model decreases drastically with descending sampling ratio while **Balanced** model is more robust to the limited sampling ratio, which also indicates that our new sampling algorithm is more robust to the sparseness of clicks in building click prediction model.

Note that, although increasing sampling ratio can lead to better performance of **Random** model, it will give rise to more time complexity for model training. To achieve the same accuracy of click prediction, our new sampling algorithm can leverage smaller set of data with much fewer training time compared to random sampling.

6. CONCLUSION AND FUTURE WORK

In this paper, we point out that data sampling is an essentially important challenge in building click prediction model in sponsored search. Through a thorough theoretical and data analysis, we find that, due to the sparseness of clicks in sponsored search, there exists a sampling dilemma between retaining similar input distribution and consistent conditional output distribution, and the straightforward random sampling method fails to address this problem. Accordingly, we propose a new sampling algorithm to balance these two aspects. Large scale evaluations on the click-through logs from a commercial search engine demonstrate that the accuracy of click prediction can be improved significantly by using the training data obtained based on this new sampling approach.

According to our findings in this paper, input distribution and conditional output distribution comprise of a dilemma for achieving effective click prediction model. In this paper, we make an effort to address this dilemma from the perspective of data sampling. In future, we plan to conduct more investigation of this dilemma from the other perspectives, such as feature engineering and modeling, and take further attempts to address it through these new perspectives.

7. REFERENCES

[1] V. Abhishek and K. Hosanagar. Keyword generation for search engine advertising using semantic similarity between terms. In *Proc. of EC*, 2007.
[2] J. Attenberg, S. Pandey, and T. Suel. Modeling and predicting user behavior in sponsored search. In *Proc. of KDD*, 2009.
[3] A. Berger and V. Pietra. A maximum entropy approach to natural language processing. In *Computational Linguistics*, 1996.
[4] H. Cheng and E. Cantu-Paz. Personalized click prediction in sponsored search. In *Proc. of WSDM*, 2010.
[5] C. Clarke, E. Agichtein, S. Dumais, and R. White. The influence of caption features oh clickthrough patterns in web search. In *Proc. of SIGIR*, 2007.
[6] 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.
[7] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet adverstising and the generalized second-price auction: selling billions of dollars worth of keywords. In *The American Economic Review*, 2007.
[8] D. Fain and J. Pedersen. Sponsored search: a brief history. In *Proc. of 2nd Workshop on Sponsored Search Auctions*, 2006.
[9] T. Graepel, J. Candela, T. Borchert, and R. Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *Proc. of ICML*, 2010.
[10] B. Jansen and T. Mullen. Sponsored search: an overview of the concept, history, and technology. In *International Journal of Electric Business*, 2008.
[11] T. P. Minka. A comparison of numerical optimizers for logistic regression. In *Technical report, Microsoft*, 2003.
[12] A. Mordecai. *Nonlinear Programming: Analysis and Methods*.
[13] F. Radlinski, A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, and L. Riedel. Optimizing relevance and revenue in ad search: a query substitution approach. In *Proc. of SIGIR*, 2008.
[14] H. Raghavan and R. Iyer. Evaluating vector-space and probabilistic models for query to ad matching. In *Proc. of SIGIR Workshop on Information Retrieval for Advertising*, 2008.
[15] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proc. of WWW*, 2007.
[16] B. Shaparenko, O. Cetin, and R. Iyer. Data-driven text features for sponsored search click prediction. In *Proc. of ADKDD*, 2009.
[17] C. Xiong, T. Wang, W. Ding, Y. Shen, and T.-Y. Liu. Relational click prediction for sponsored search. In *Proc. of WSDM*, 2012.
[18] W. Xu, E. Manavoglu, and E. Cantu-Paz. Temporal click model for sponsored search. In *Proc. of SIGIR*, 2010.
[19] W. Zhang, X. He, B. Rey, and R. Jones. Query rewriting using active learning for sponsored search. In *Proc. of SIGIR*, 2007.