Calibrating User Response Predictions in Online Advertising

Chao Deng(✉), Hao Wang, Qing Tan, Jian Xu, and Kun Gai

Alibaba Group, Beijing, China
{fengyang.dc, wh111044, qing.tan, xiyu.xj, jingshi.gk}@alibaba-inc.com

Abstract. Predicting user response probability such as click-through rate (CTR) and conversion rate (CVR) accurately is essential to online advertising systems. To obtain accurate probability, calibration is usually used to transform predicted probabilities to posterior probabilities. Due to the sparsity and latency of the user response behaviors such as clicks and conversions, traditional calibration methods may not work well in real-world online advertising systems. In this paper, we present a comprehensive calibration solution for online advertising. More specifically, we propose a calibration algorithm to exploit implicit properties of predicted probabilities to reduce negative impacts of the data sparsity problem. To deal with the latency problem in calibrating delayed responses, e.g., conversions, we propose an estimation model to leverage post-click information to approximate the real delayed user responses. We also notice that existing metrics are insufficient to evaluate the calibration performance. Therefore, we present new metrics to measure the calibration performance. Experimental evaluations on both real-world datasets and online advertising systems show that our proposed solution outperforms existing calibration methods and brings significant business values.

Keywords: Online advertising · Calibration · Click-Through Rate Prediction · Conversion Rate Prediction

1 Introduction

Online advertising is a multi-billion dollars industry with an annual revenue of 107 billion US dollars for the full year of 2018 in the United States only [27]. Compared to traditional advertising industry such as TV, online advertising provides services that tie advertisers' payment directly to measurable user responses such as clicks and conversions. Therefore, predicting user response probability accurately has become one of the essential problems in online advertising [20, 5, 12]. The most common tasks are click-through rate (CTR) prediction and conversion rate (CVR) prediction.

Predicting user response probability is usually treated as a supervised learning problem. A unique challenge is that the supervision labels are binary observations. For example, in CTR prediction, the observation is that a user either

clicks or not clicks an ad and there is no ground-truth of the underlying click probability. Therefore, most existing work for user response prediction strives to learn binary classifiers and the optimization objectives are based on classification performance such as Area-Under-Curve (AUC) of Precision-Recall (PR) and/or Receiver Operating Characteristic (ROC) curves [6]. Even if some classifiers are modeled to output the user response probability estimations directly, there are still many factors accounting for the discrepancy between *predicted probabilities* and *posterior probabilities*. These factors include inaccurate modeling assumption, deficiencies in the learning algorithm [13, 10], hidden features being not available at training and/or serving time [20], data up/down sampling [12, 15], etc. While much research effort has been endeavored to address these factors, *calibration* provides a complementary and alternative approach to resolve the discrepancies by transforming predicted probabilities to posterior probabilities directly [21, 10, 15]. There are two additional benefits associated with calibration from the perspective of advertising system designs. First, calibration is helpful for a loosely coupled system design which separates the concerns of optimization in the auction and the machine learning machinery [20]. Second, calibration is a light-weight solution to cope with the real-time changes in the online environment whenever the user response prediction models are not able to capture the changes in a timely manner.

For these reasons, in online advertising systems, calibration is usually designed as a module to transform predicted probabilities to posterior probabilities. Figure 1 shows the architecture of a common online advertising system. When an ad request arrives, a set of candidate ads are selected by an AD SELECTION module. Then the predicted probabilities of these ads are produced by a PREDICTION module. These predicted probabilities are calibrated by the CALIBRATION module to posterior probabilities, which are important input for the following RANKING module, where an auction mechanism determines which ad will be shown. Finally, the top ranked ad is shown to the user, and user behaviors are tracked. The tracked behavior data are used for prediction model training and calibration function learning.

The online advertising applications pose at least the following two unique challenges to calibrating user response predictions:

- **Sparsity**. User response behaviors are usually very rare. For example, the CTR in certain scenarios may be less than 1% [32]. The number of conversions can be even smaller. According to our experience from an e-commerce advertising platform, the CVR of some electronic product ads is less than 0.1%. The data sparsity problem makes it difficult to estimate the underlying probabilities from the observations.
- **Latency**. User response behaviors may have substantial delays. For example, it may take several days for a user to convert (e.g., place an order) after she clicks an ad. If only short-term responses are considered , the underlying probability will be underestimated. On the other hand, calibration would be stalled if we wait for a long time to collect response data for calibration.

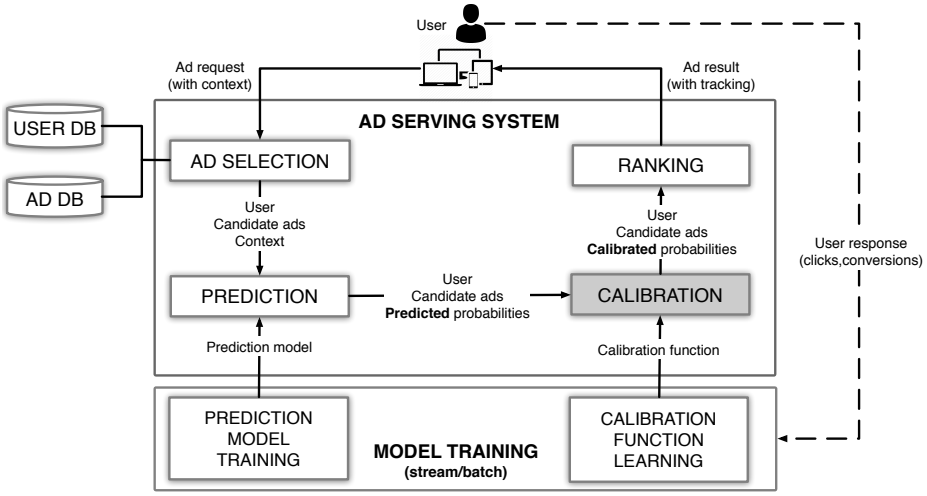

Fig. 1: An illustration of a common advertising system.

In this paper, we present a comprehensive calibration solution for online advertising. More specifically, to cope with the sparsity problem, we propose a simple yet effective calibration algorithm. This algorithm exploits the property that the predicted probabilities can rank samples well (with high AUC) and adds smoothness constraint to ensure that the calibrated probabilities keep the same order with the original predicted ones. To tackle the latency challenge, we propose an estimation model to leverage post-click information to approximate the real delayed responses.

The key contributions of the paper can be summarized as follows:

- We propose the *Smoothed Isotonic Regression* (SIR) algorithm for user response prediction calibration. The algorithm learns a monotonically increasing function to transform predicted probabilities to posterior probabilities and effectively handles data sparsity.
- We propose the *Post-Click Conversion Estimation Model* (PCCEM) for delayed response prediction calibration. The model leverages short-term post-click behaviors for conversion approximation and effectively solves the delayed response problem.
- We present new metrics to measure the calibration performance. Experimental evaluations on two real-world datasets and online advertising systems demonstrate the effectiveness of our calibration solution.

2 Related work

There has been extensive research on user response prediction [34, 28, 4, 19, 18, 17, 8], and calibration methods have been introduced as part of the prediction

solution [9, 15, 20, 3, 12, 5]. However, the importance of calibration is usually underrated, and there is no special study on calibration in online advertising to the best of our knowledge.

In a more general paradigm, calibration can be regarded as a process to produce a function to transform predicted probabilities to posterior probabilities. Existing calibration methods could be divided into parametric and nonparametric ones. Platt's method [25] is a traditional parametric method, which tries to fit a sigmoid calibration function [14]. Beta calibration [14] added more flexibility by assuming that the scores are beta distributed. These methods may fail when their parametric assumptions are not met.

The most popular nonparametric method is Isotonic Regression [31, 26, 23]. This method tries to find a monotonically increasing function to minimize the squared error between the calibrated probabilities and user response values. A commonly used algorithm for isotonic regression is the pair-adjacent-violator (PAV) [1] algorithm. On sparse datasets, the spiking problem [24] makes this method sensitive to the samples with maximum and minimum predicted probabilities. Another commonly used calibration method is binning method [30, 29], of which the main idea is to divide samples into bins and calibrate a predicted probability to the posterior probability of the bin it belongs to. One limitation of this method is that the number of bins needs to be set properly. The BBQ method [22] was then proposed to consider different number of bins and use their weighted average to yield more robust calibrations. However, It is hard to calculate accurate posterior probability of each bin on sparse dataset and binning based methods may not work well.

3 User Response Prediction Calibration

In this section, we first define the problem of calibration and give a brief overview of our calibration solution. Then we introduce the Smoothed Isotonic Regression (SIR) algorithm for a general calibration solution and the Post-Click Conversion Estimation Model (PCCEM) for solving the delayed response problem.

3.1 Problem Definition and Solution Overview

Calibration was defined as a measure: a binary classifier is perfectly calibrated if for a sample of examples with predicted probability p, the expected proportion of positives is close to p [2, 14, 21]. However, calibration has been recently used to denote the process of obtaining the posterior probabilities [20]. It is beneficial to define the calibration problem more precisely.

Let $\mathcal{X} \subseteq [0, 1]$ be the predicted probability space from a prediction model and $\mathcal{Y} = \{0, 1\}$ be the user response space where 0 denotes negative response and 1 denotes positive response. Let random variable X denote the predicted probability and Y denote the response value. We define the conditional expectation of Y given $X = x$ as

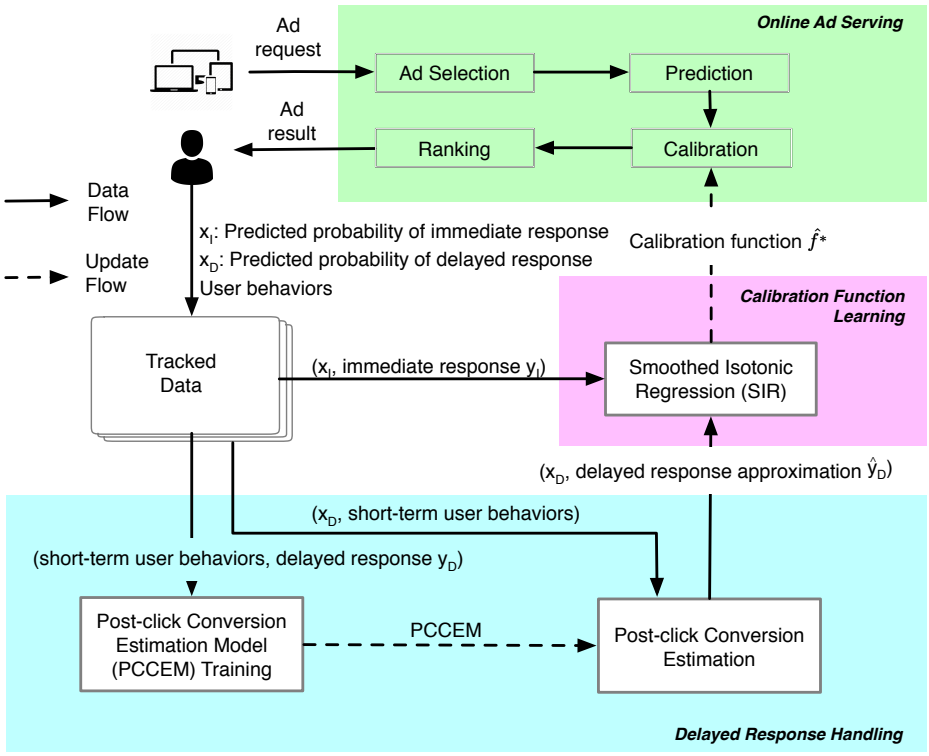

Fig. 2: Calibration solution overview.

$$\begin{aligned} E[Y|X=x] &= \sum_{y\in\mathcal{Y}} y \lim_{\epsilon\to 0^+} P(Y=y\,|\,|X-x|\le\epsilon) \\ &= \lim_{\epsilon\to 0^+} P(Y=1\,|\,|X-x|\le\epsilon) \end{aligned} \tag{1}$$

We are particularly interested in the error function

$$J(X,Y) = \int_{\mathcal{X}} (E[Y|X=x]-x)^2 dx \tag{2}$$

If $J(X,Y)=0$, the prediction model is said to be perfectly calibrated. Otherwise, let $f:\mathcal{X}\to[0,1]$ denote a function from $\mathcal{X}$ to $[0,1]$. The goal of calibration is to find the function

$$f^* = \arg\min_f \int_{\mathcal{X}} (E[Y|X=x]-f(x))^2 dx \tag{3}$$

In this paper, we slightly abuse the terminology so that we define *calibration* as the process of finding the optimal function f^*.

Algorithm 1 Smoothed Isotonic Regression

Input: training set $T = \{(x_i, y_i) | i \leq N\}$; bin size n;
Output: mapping function $f(x)$;

Phase 1 – Binning

Sort T according to x_i, get list $L = [(x_i, y_i)]$, where $\underset{i \leq j}{\forall} x_i \leq x_j$
Initialize empty list BL
Number of bins $K = \lfloor \frac{N}{n} \rfloor$
for $k = 0$ to $K - 1$ **do**
    $S = \{i | nk \leq i < n(k+1)\}$
    $l_k = \min_{i \in S} x_i$, $u_k = \max_{i \in S} x_i$, $v_k = \frac{\sum_{i \in S} y_i}{|S|}$, $c_k = |S|$
    Append bin (l_k, u_k, v_k, c_k) to BL
end for

Phase 2 – Pair Adjacent Violator for Bins

Initialize empty list IBL
for $i = 0$ to $|BL| - 1$ **do**
    Initialize $l = l_i$, $u = u_i$, $v = v_i$, $c = c_i$
    if IBL is empty **then**
        $IBL = [(l, u, v, c)]$
        continue
    end if
    Choose bin $t = (l_t, u_t, v_t, c_t)$ at the end of IBL
    while $v \leq v_t$ **do**
        $l = l_t$, $v = \frac{v \times c + v_t \times c_t}{c + c_t}$, $c = c + c_t$
        Remove t from IL
        Choose bin $t = (l_t, u_t, v_t, c_t)$ at the end of IBL
    end while
    Append new bin (l, u, v, c) to IBL
end for

Phase 3 – Interpolation

Initialize empty list ML
for $i = 0$ to $|IBL| - 2$ **do**
    $m_i = \frac{l_i + u_i}{2}$, $m_{i+1} = \frac{l_{i+1} + u_{i+1}}{2}$
    $a = \frac{v_{i+1} - v_i}{m_{i+1} - m_i}$, $b = v_i - a m_i$
    Append bin (m_i, m_j, a, b) to ML
end for
$f(x) = a_i x + b_i \qquad if \quad (l_i, u_i, a_i, b_i) \in ML, l_i < x \leq u_i$
return $f(x)$

However, finding f^* is not a trivial task. First, it is impossible to calculate $E[Y|X = x]$ directly because we can only observe limited samples drawn from the joint distribution of (X, Y). The best thing one can do is to find an approximate function $\hat{f}^*$ based on these observed samples. Second, in real world applications, the environment may change over time and the prediction model may not capture these changes in a timely manner, so that the joint distribution of (X, Y) may change over time as well. Therefore the calibration function $\hat{f}^*$ is not static and should be updated timely. Third, another unique challenge brought by online advertising is that user responses can be delayed [4]. Such delays hinder $\hat{f}^*$ to be updated in time.

We propose a generic calibration solution to tackle all these challenges. Figure 2 shows the architecture of this solution. The Smoothed Isotonic Regression (SIR) module receives samples and updates the calibration function $\hat{f}^*$ for the online calibration module. For immediate response (click) prediction calibration, the calibration function can be learned with predicted probability x_I and immediate response y_I directly in the SIR module. On the other hand, for delayed response (conversion) prediction calibration, a post-click conversion estimation mechanism is designed to leverage the post-click user behaviors to approximate the delayed response. More specifically, the Post-Click Conversion Estimation Model (PCCEM) Training module collects short-term user behaviors and delayed response y_D to learn the PCCEM, which is used in the Post-Click Conversion Estimation module to produce the approximated delayed response $\hat{y}_D$. The benefit of this design is that the SIR module can also receive calibration learning samples $(x_D, \hat{y}_D)$ for delayed response prediction in a timely manner.

3.2 Smoothed Isotonic Regression (SIR)

On the one hand, the joint distribution of (X, Y) usually changes over time in real-world applications. Hence, we believe that nonparametric methods would be preferable to those based on some distribution assumption when designing the calibration function $\hat{f}^*$. On the other hand, recent advances of the prediction models that optimize objectives based on ranking performance such as AUC [16, 9, 20, 34] provide more and more accurate rankings. This property could be useful while learning calibration function. Therefore, we propose Smoothed Isotonic Regression (SIR), a practical nonparametric method.

The details of SIR is presented in Algorithm 1. The inputs of SIR are training set T and bin size n. First, a binning strategy is used to produce a sorted list of bins BL (Phase 1). Second, Isotonic Regression is applied to ensure that the posterior probability of each bin in BL is monotonically increasing. We adopt the pair-adjacent-violator (PAV) algorithm due to its computational efficiency. However, the vanilla PAV algorithm needs to be modified to be applicable to bins. For two adjacent bins, if the monotonicity is violated, they are pooled together to generate a new bin (Phase 2). Finally, an interpolation strategy is used to derive a monotonic and smoothed function. We note that SIR does not put any constraint on the choice of interpolation strategy. For simplicity, we only present the linear interpolation strategy (Phase 3).

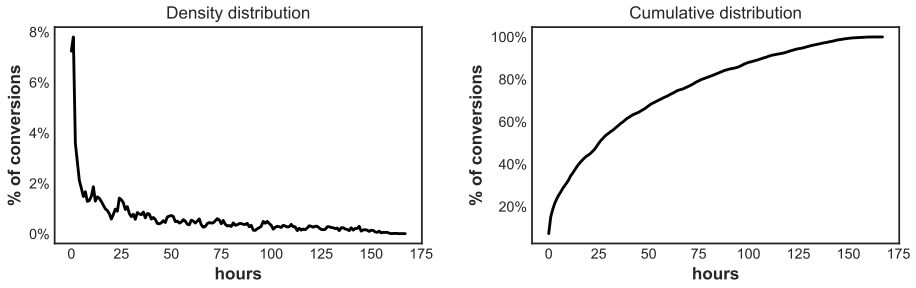

Fig. 3: Distribution of conversion within different time delays.

3.3 PCCEM based Calibration for Delayed Response

The SIR algorithm proposed in section 3.2 is a general algorithm suitable for various calibration tasks. However, there is another challenge for certain calibration tasks: when user responses have substantial delays. For example, a conversion may happen several days later after the click. Figure 3 shows a case of such conversion delays for an advertiser on an e-commerce website. As we can see from the cumulative distribution, only a small portion (8%) of the conversions happen within an hour after the click and about 17% conversions have delays for more than 100 hours. The delays of the responses result in difficulty for updating the calibration function: if we only use conversions in a short period of time after the click, the conversion rate may be substantially underestimated after calibration. However, to collect all the conversions, we may have to wait for a long time, e.g., a couple of days, which is undesirable for the calibration function to be updated timely. To deal with this problem, we introduce the Post-Click Conversion Estimation Model (PCCEM) to leverage short-term post-click user behaviors for conversion approximation. Then we use these approximated conversions to update calibration functions.

Before detailing PCCEM, we provide an intuitive example as follows. Suppose a user clicks an ad on the first day, and places an order five days later. Although we can not observe the conversion until five days later, there can be plenty of post-click user behaviors that can help us predict how likely the user will convert. For example, the user may spend a long time on the landing page and add the item to shopping cart, etc. These post-click behaviors usually happen in a short period of time, e.g., within a few minutes after the click. Strong evidence shows that these user behaviors are very good conversion predictors. Figure 4 illustrates one such example: both the landing page session duration and the number of page views have positive correlation with conversion rate.

The PCCEM is built on top of the short-term user behaviors to produce a *post-click score* which quantifies the probability of the final conversion. The model is fitted with a dataset with post-click information as features and real conversions as labels, capturing patterns in the post-click behaviors that are correlated with the final conversion. It is worth noting that the conventional

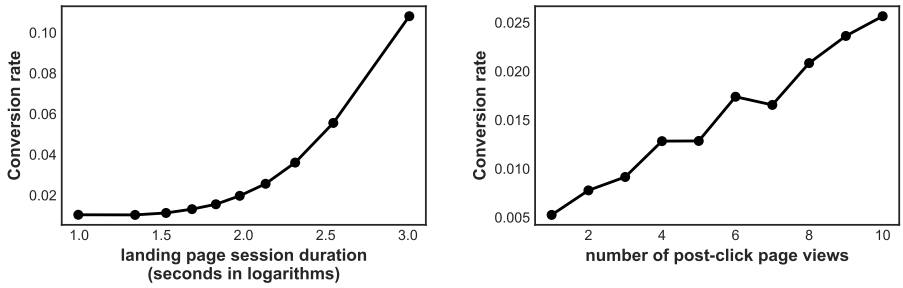

Fig. 4: The relationship between post-click information within one hour after the click and average conversion rate.

Algorithm 2 PCCEM based Calibration for Delayed Response

for long-term period $T = 1, 2, 3, ...$ **do**
  Generate training set S by using post-click information as features and real responses as labels
  Fit PCCEM m_T based on S
  for each short-term period t in T **do**
    for each request i in time period t **do**
      Collect post-click information
      Use m_T to produce user response score $\hat{y}_i$
      Generate a sample $(x_i, \hat{y}_i)$
    end for
    Generate sample set D of recent samples
    Update f by using SIR based on D
  end for
end for

conversion rate prediction model used in ad auctions is unable to leverage such information since the conversion rates are predicted and used *before* the ad impressions and clicks.

With PCCEM, the update procedure of calibration solution for delayed response consists of two parts: PCCEM is used to generate post-click score for each click. These scores are used to update the calibration function. The algorithm is shown in Algorithm 2.

4 Metrics

In this section, we first review traditional metrics and discuss their defects. Then, we propose new metrics which can better quantify the calibration performance.

Predicted click over click (PCOC)[1][12, 9] is the most commonly used quantitative metric for measuring calibration performance, which is calculated as the ratio of the average calibrated probability and the posterior probability on the whole dataset. The posterior probability is underestimated if PCOC < 1 and overestimated if PCOC > 1. The less PCOC deviates from 1, the better the calibration is. However, PCOC is insufficient to evaluate the calibration performance. The following example shows the the defect of PCOC.

EXAMPLE 1 (Defect of PCOC). *Suppose we have 20,000 samples, and half of them have calibrated probability* 0.2 *whose posterior probability is* 0.4 *(underestimated). The other samples have calibrated probability* 0.8 *and their posterior probability is* 0.6 *(overestimated). However, the PCOC of these 20,000 samples is* $\frac{0.2*10000+0.8*10000}{0.4*10000+0.6*10000} = 1$. *These samples have well-calibrated probability values according to PCOC, but they really don't.*

The misleading result roots from the fact that PCOC does not consider the distribution of calibrated probabilities. If we know the joint distribution of (X, Y), we can calculate PCOC for each given x as $\frac{E[Y|X=x]}{x}$. However, we can only get limited samples drawn from the joint distribution of (X, Y). An alternative is to aggregate samples with similar calibrated probabilities to approximately calculate $E[Y|X = x]$ and evaluate the error on different x. Based on this idea, we present a new metric *calibration-N* (Cal-N). First, the calibrated probabilities are divided into N bins with equal frequency and PCOC for each bin is calculated. Then, the error of the $i-$th bin can be defined as

$$error_i = \begin{cases} \text{PCOC}_i - 1 & \text{PCOC}_i \geq 1 \\ \frac{1}{\text{PCOC}_i} - 1 & \text{PCOC}_i < 1 \end{cases} \tag{4}$$

Note that when $\text{PCOC}_i < 1$, we use its reciprocal so that overestimation and underestimation are equally treated. Finally, we use the root mean square to accumulate these N errors. To put it formally, Cal-N is defined as

$$\text{Cal-}N = \sqrt{\frac{\sum_{i=1}^{N} error_i^2}{N}} \tag{5}$$

The lower the value is, the better the predicted probabilities are calibrated. Compared with PCOC, Cal-N can accumulate the calibration error across different calibrated probability subspaces. Consider Example 1 again, the Cal-N ($N = 2$) of the 20,000 samples is 2.4, which significantly differs from 0, which means these samples don't have well-calibrated probability values.

In online advertising, a *campaign* is the minimum entity for an advertiser to setup a marketing strategy, which includes budget, target ad audience, creatives and bid price, etc. Therefore, we are concerned with calibration performance of each advertising campaign. Thus we also propose a domain-specific metric

[1] In the literature this metric is called *calibration*. We use a different name here to avoid confusion.

grouped calibration-N (GC-N), which is the weighted average Cal-N of m campaigns.

$$\text{GC-}N = \frac{\sum_{j=1}^{m} w_j \text{Cal-}N_j}{\sum_{j=1}^{m} w_j} \tag{6}$$

where Cal-N_j and w_j are Cal-N and importance weight of campaign j respectively. In our experiments, w_j can be the number of samples in the j-th campaign. The lower GC-N is, the better the calibrated result is for each campaign.

It is worth mentioning that *log-loss* is commonly used to compare the prediction performance in some binary supervised learning problem literatures [18, 11]. A smaller log-loss means better probabilities. However, the absolute value of log-loss is not a good indicator of how well the predictions are calibrated. For example, suppose we get samples with 0/1 labels drawn from a binomial distribution with $p = 0.5$, the log-loss on these samples is not 0 if we predict the ground truth probability 0.5 for each sample. In real world applications, it is important to know the performance of perfectly calibrated probabilities because this could help us to measure the available performance optimizations for a certain problem.

5 Experimental Evaluation

In this section, we conduct experiments on three real-world online advertising datasets. First, we report the experiment results on CTR calibration, comparing SIR with state-of-the-art methods. For delayed response prediction calibration problem, we report the experiment results on CVR calibration. Our solution is also deployed for online A/B test.

5.1 Evaluation of SIR

Dataset The experiments are conducted on two datasets. *Dataset A* is from a world-leading advertising platform[2]. This dataset comprises roughly 50 million impressions randomly sampled from the ad serving log from July 1 to July 25, 2018. Each impression has its predicted CTR and a label indicating whether the user clicks on the ad. *Dataset B* is a public dataset from iPinYou[3]. Details of this dataset are introduced in [32]. Since there is no predicted CTR in this dataset, we first use the training set to construct a prediction model with GBDT [7], then produce predicted probabilities on test set. The test set augmented with predicted CTR is used for our experiment on calibration.

To make the experimental setup similar to real-world application scenarios, for both datasets, we update and evaluate calibration functions of different methods by an hourly sliding window: for each hour, the calibration function of each campaign is updated on the set of samples in the past 24 hours (training set) and evaluated on the set of samples in next hour (test set). Then, we aggregate all the hourly results.

[2] Dataset A is available at https://tianchi.aliyun.com/dataset/dataDetail?dataId=40792
[3] iPinYou dataset is available at http://data.computational-advertising.org

Comparative Experiment To validate the effectiveness of our method, we compare the performance of Smoothed Isotonic Regression (SIR) against the state-of-the-art methods. The methods of the comparative experiment are as follows:

(1) **BBQ** [22]: the state-of-the-art binning based method. This method considers different number of bins and uses their weighted average to yield more robust calibration results. The parameters of BBQ are set as the same as [22].
(2) **IR** [3]: the most popular nonparameteric method. We implement this method by the PAV algorithm [1].
(3) **Beta calibration** [14]: the state-of-the-art parametric method. This method assumes that the predicted probabilities and user response values are beta distributed.
(4) **SIR**: our proposed method in this paper. We set bin size n as 1,000.

Table 1 shows GC-N of these methods on the test sets with various N. Usually a larger N will help us evaluate the performance in a more detailed way (Recall that GC-N reduces to PCOC with $N = 1$). As we can see, SIR can decrease GC-N 10.4% on average on dataset A and 29.9% on average on dataset B respectively. SIR outperforms BBQ because SIR leverages the property that the predicted probabilities can rank samples well. BBQ assumes that all bins are independent with each other, so it fails to exploit the ranking relationships between different bins. SIR outperforms IR because the binning phase in SIR reduces the effect of the spiking problem [24] of the PAV algorithm. This problem makes PAV algorithm sensitive to the samples with maximum predicted probabilities and positive labels. This would make IR performs instability on sparse dataset. As we can see, the performance of IR is bad on dataset A. Beta calibration has the closet performance to SIR, which decreases GC-N 9.9% on average on dataset A and 27.7% on average on dataset B respectively. Since beta calibration has pre-defined parametric function curve, the performance is less affected by data sparsity. But it's also this parametric assumption that limits its accuracy of fitting, while SIR is more adaptive to various of distributions due to no distribution assumption.

5.2 Evaluation of PCCEM based Calibration

Dataset The dataset in this experiment is from a world-leading advertising platform[4]. This dataset comprises roughly 7 million clicks randomly sampled from the ad serving log from July 1 to July 21, 2018. Each record in the dataset contains information related to a click, including pre-click and post-click features. The pre-click features include a lot of user behavior information before click such as number of views/purchases. The post-click features includes landing page session duration after click, number of views/purchases after click in an hour, number of add items into cart/favorites in an hour. Data between July 15th to July 21th are used as test set.

[4] This dataset is available at https://tianchi.aliyun.com/dataset/dataDetail?dataId=40796

Table 1: GC-N of different methods on two test sets. For a sufficient comparison, we use $N = 3, 4, 5$ to calculate GC-N. The best value of them on test sets is highlighted.

Dataset	Method	$N = 3$	$N = 4$	$N = 5$
A	No calibration	0.63	0.67	0.72
	(1) BBQ	0.57 (-9.5%)	0.63 (-6.0%)	3.05 (+323.6%)
	(2) IR	0.82 (+30.0%)	0.89 (+32.8%)	0.98 (+36.1%)
	(3) Beta calibration	0.56 (-11.1%)	0.60 (-10.4%)	0.66 (-8.3%)
	(4) SIR	**0.56 (-11.1%)**	**0.60 (-10.4%)**	**0.65 (-9.7%)**
B	No calibration	0.40	0.45	0.46
	(1) BBQ	0.71 (+77.5%)	0.81 (+80.0%)	0.84 (+82.6%)
	(2) IR	0.31 (-22.5%)	0.35 (-22.2%)	0.40 (-13.0%)
	(3) Beta calibration	0.27 (-32.5%)	0.32 (-28.9%)	0.36 (-21.7%)
	(4) SIR	**0.27 (-32.5%)**	**0.30 (-33.3%)**	**0.35 (-23.9%)**

Comparative Experiment In this section, we compare the conversion calibration performance of different methods. To make it fair, all the methods use our proposed SIR algorithm as the calibration algorithm. We consider the following methods:

(1) **Short-term Calibration (STC)**: For each click, we only use the short-term conversions that are within one hour after the clicks to update the calibration function.
(2) **Long-term Calibration (LTC)**: For each click, we wait for 7 days to get the true conversions. In this case, the calibration function can only be updated with real conversions in the past 7 days and the clicks no later than 7 days ago.
(3) **PCCEM based Calibration (PCCEM)**: We update PCCEM on a daily basis: each day whenever the new conversion data are available, PCCEM is updated. For each hour, when the new short-term behavior data are available, post-click scores are produced by the latest PCCEM, then the calibration function is updated based on these scores.

Table 2 shows the experiment results for the three methods. As we can see, STC has the worst performance as the short-term conversions underestimates the conversion rates by a large margin. Method LTC makes calibration function updating stalled so it may not capture the relationship between predicted conversion rate and real conversion in time. Method PCCEM outperforms both methods, effectively improving calibration performance with delayed response.

5.3 Online Evaluation

To investigate whether our proposed approach can help improve the business performance, we also conducted an online A/B test experiment on a world-leading advertising platform. The experiment lasted for seven days with the setup that the control bucket uses the predicted probabilities given by a deep

Table 2: GC-N of three calibration methods for predicted conversion rate calibration.

Method	$N = 3$	$N = 4$	$N = 5$
No calibration	4.46	4.47	4.49
(1) STC	5.67 (+27.1%)	5.91 (+32.2%)	7.11 (+58.4%)
(2) LTC	1.57 (-64.8%)	1.72 (-61.5%)	1.81 (-59.7%)
(3) PCCEM	**0.50 (-88.8%)**	**0.58 (-87.0%)**	**0.71 (-84.2%)**

Table 3: Business result of A/B test.

Business metrics	RPM	CTR	ROI
Improvement(%)	+3.86%	+8.93%	+5.07%

learning-based prediction model and the test bucket uses the further calibrated probabilities produced by our proposed approach. Each bucket was assigned 10% of all the online traffic which was in the magnitude of tens of millions. Generally speaking, an advertising platform strives to provide values to the advertisers, the users, and the platform itself. A better response prediction is expected to contribute to all these three values. We use *return on investment* (ROI)[5] to indicate the advertisers' benefit, CTR to indicate the user experience and *revenue per mille* (RPM) to quantify the platform's gain. Results are shown in Table 3, we can observe that our calibration solution can increase RPM by 3.86%, CTR by 8.93%, and ROI by 5.07%.

6 Conclusion and future work

In this paper, we introduced a calibration solution for user response prediction in online advertising, including the SIR algorithm for data sparsity problem and the PCCEM for delayed response problem. We also proposed new metrics to evaluate the effectiveness of calibration. Experiment results on real-world datasets have proven that the calibration solution can lead to significantly better results both in terms of technical measurements and business performance.

There is an interesting direction for our future work. In many applications, the distribution of the observed samples can be different from the distribution of the ones whose response probabilities need to be predicted and calibrated [33]. Therefore, it is beneficial to design an unbiased calibration algorithm in this case.

References

1. Ayer, M., Brunk, H.D., Ewing, G.M., Reid, W.T., Silverman, E., et al.: An empirical distribution function for sampling with incomplete information. The annals of

[5] The *return* is the value of the conversions and the *investment* is the cost charged by the advertising platform.

mathematical statistics **26**(4), 641–647 (1955)
2. Bella, A., Ferri, C., Hernández-orallo, J., Ramírez-quintana, M.J.: Calibration of machine learning models.
3. Borisov, A., Kiseleva, J., Markov, I., de Rijke, M.: Calibration: A simple way to improve click models. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 1503–1506. ACM (2018)
4. Chapelle, O.: Modeling delayed feedback in display advertising. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1097–1105. KDD '14, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2623330.2623634, http://doi.acm.org/10.1145/2623330.2623634
5. Chappelle, O., Manavoglu, E., Rosales, R.: Simple and scalable response prediction for display advertising. ACM Transactions on Intelligent Systems and Technology **2**(3), Article 1 (2015). https://doi.org/10.1145/0000000.0000000, http://arxiv.org/abs/1502.07526
6. Fawcett, T.: An introduction to roc analysis. Pattern recognition letters **27**(8), 861–874 (2006)
7. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
8. Gentile, C., Li, S., Kar, P., Karatzoglou, A., Etrue, E., Zappella, G.: On context-dependent clustering of bandits. arXiv preprint arXiv:1608.03544 (2016)
9. Graepel, T., Candela, J.Q., Borchert, T., Herbrich, R.: Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 13–20 (2010)
10. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. pp. 1321–1330 (2017), http://proceedings.mlr.press/v70/guo17a.html
11. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: DeepFM: A factorization-machine based neural network for CTR prediction. In: IJCAI International Joint Conference on Artificial Intelligence. pp. 1725–1731 (2017). https://doi.org/10.1145/2988450.2988454
12. He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., et al.: Practical lessons from predicting clicks on ads at facebook. In: Proceedings of the Eighth International Workshop on Data Mining for Online Advertising. pp. 1–9. ACM (2014)
13. King, G., Zeng, L.: Logistic regression in rare events data. Political analysis **9**(2), 137–163 (2001)
14. Kull, M., Silva Filho, T., Flach, P.: Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In: Artificial Intelligence and Statistics. pp. 623–631 (2017)
15. Lee, K.c., Orten, B., Dasdan, A., Li, W.: Estimating conversion rate in display advertising from past erformance data. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 768–776. ACM (2012)
16. Li, C., Lu, Y., Mei, Q., Wang, D., Pandey, S.: Click-through prediction for advertising in twitter timeline. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1959–1968. ACM (2015)

17. Li, S., Karatzoglou, A., Gentile, C.: Collaborative filtering bandits. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 539–548. ACM (2016)
18. Liu, Q., Yu, F., Wu, S., Wang, L.: A Convolutional Click Prediction Model. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM '15. pp. 1743–1746 (2015). https://doi.org/10.1145/2806416.2806603, http://dl.acm.org/citation.cfm?doid=2806416.2806603
19. Lu, Q., Pan, S., Wang, L., Pan, J., Wan, F., Yang, H.: A practical framework of conversion rate prediction for online display advertising. In: Proceedings of the ADKDD'17. p. 9. ACM (2017)
20. Mcmahan, H.B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Hrafnkelsson, A.M., Boulos, T., Kubica, J.: Ad click prediction: a view from the trenches. In: In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD. pp. 1222–1230 (2013)
21. Menon, A.K., Jiang, X.J., Vembu, S., Elkan, C., Ohno-Machado, L.: Predicting accurate probabilities with a ranking loss. In: Proceedings of the... International Conference on Machine Learning. International Conference on Machine Learning. vol. 2012, p. 703. NIH Public Access (2012)
22. Naeini, M.P., Cooper, G.F., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. In: AAAI. pp. 2901–2907 (2015)
23. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: Proceedings of the 22nd international conference on Machine learning. pp. 625–632. ACM (2005)
24. Pal, J.K.: Spiking problem in monotone regression: Penalized residual sum of squares. Statistics & Probability Letters **78**(12), 1548–1556 (2008)
25. Platt, J., et al.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers **10**(3), 61–74 (1999)
26. Robertson, T., Robertson, T.: Order restricted statistical inference. Tech. rep. (1988)
27. Statista: Online advertising revenue in the united states from 2000 to 2018 (2019), https://www.statista.com/statistics/183816/us-online-advertising-revenue-since-2000/,Last Last accessed on 2020-04-02
28. Yang, H., Lu, Q., Qiu, A.X., Han, C.: Large scale cvr prediction through dynamic transfer learning of global and local features. In: Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications. pp. 103–119 (2016)
29. Zadrozny, B., Elkan, C.: Learning and making decisions when costs and probabilities are both unknown. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 204–213. ACM (2001)
30. Zadrozny, B., Elkan, C.: Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In: ICML. vol. 1, pp. 609–616 (2001)
31. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 694–699. ACM (2002)
32. Zhang, W., Yuan, S., Wang, J., Shen, X.: Real-time bidding benchmarking with ipinyou dataset. arXiv preprint arXiv:1407.7073 (2014)

33. Zhang, W., Zhou, T., Wang, J., Xu, J.: Bid-aware gradient descent for unbiased learning with censored data in display advertising. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 665–674. ACM (2016)
34. Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., Gai, K.: Deep interest network for click-through rate prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1059–1068. ACM (2018)