arXiv:1603.03713v2 [cs.LG] 19 Sep 2016

Cost-sensitive Learning for Utility Optimization in Online Advertising Auctions

Flavian Vasile[1], **Damien Lefortier**[1,2] **and Olivier Chapelle**[3]
[1]Criteo, Inc.
[2]University of Amsterdam
[3]Google, Inc.
{f.vasile, d.lefortier}@criteo.com, olivier@chapelle.cc

Abstract

One of the most challenging problems in computational advertising is the prediction of click-through and conversion rates for bidding in online advertising auctions. An unaddressed problem in previous approaches is the existence of highly non-uniform misprediction costs. While for model evaluation these costs have been taken into account through recently proposed business-aware offline metrics – such as the *Utility* metric which measures the impact on advertiser profit – this is not the case when training the models themselves. In this paper, to bridge the gap, we formally analyze the relationship between optimizing the Utility metric and the log loss, which is considered as one of the state-of-the-art approaches in conversion modeling. Our analysis motivates the idea of weighting the log loss with the business value of the predicted outcome. We present and analyze a new cost weighting scheme and show that significant gains in offline and online performance can be achieved.

1 Introduction

Online advertising is becoming a large part of the global marketing reaching \$170 billion revenue in 2015 [1]. Depending on the goal of the advertising campaign, different pricing schemes exist, but, out of them, brand and performance advertising are the most prevalent. Brand advertising is used by advertisers that want to maximize the exposure of their advertising message to online users and is priced in terms of number of ad impressions, with the cost usually referred as CPM (cost-per-mille). By contrast, performance advertising is appealing to advertisers that are interested in reaching certain measurable goals such as increased number of visits to their websites, increased number of leads, sales or downloads. In this case, the cost is referred as CPC (cost-per-(ad)click) or CPA (cost-per-conversion).

The marketplace that makes online advertising possible is roughly formed out of three types of players, namely the advertiser (the demand of ad display opportunities), the publisher (the offer of ad display opportunities) and the auction house, represented by a Real-Time Bidding (RTB) platform. Most of the RTB platforms use a 2nd price model [15], where advertisers or agents representing the advertisers bid for display opportunities, and the winner pays the maximum between the bid of the second highest bidder in the auction and the reserve price. In order to determine the winner for CPC and CPA clients, where the pay-off to the publisher is conditioned on a user action, the bids get converted in expected values (also known as eCPMs) using click and conversion rate (CR) prediction models.

The focus of this paper is on improving the performance of a bidder, defined as an agent that takes the CPC or CPA that the advertiser is willing to pay and submits a CPM bid for the impression. For CPA clients, this bidder takes as input the CPA, that is the value of the sale for the advertiser, computes a predicted CR and produces a bid. One important aspect of the marketplace is that the numeric range of the possible CPAs is large and depends on the economic value of the sale. The resulting eCPMs vary from ones based on expectations over sales that are frequent and low-value (e.g. song downloads) and ones that are rare and high-value (e.g. hotel reservations). An improvement in prediction performance on high CPA sales has a bigger impact on the revenue than a similar improvement that affects low CPA traffic. To take this into account during evaluation, recently proposed metrics on bidding performance make use of the associated CPAs [5, 9].

In this paper, we investigate a novel way of taking into account the sales' CPAs in our CR prediction model for bidding in online advertising auctions, thus bridging the gap between the recently proposed business-aware offline metrics and the current state-of-the-art CR prediction models. The outline is as follows. In Section 2, we present the setting, i.e., our state-of-the-art CR model, the bidder around it, and the recent business-aware offline metrics. Then, in Section 3, we introduce our method for taking into account the advertisers' CPAs when training our CR model, which is based on our analysis of the relationship between the Utility loss [9] and the standard log loss. Finally, in Section 4, we present our experimental results, both offline and online, before concluding in Section 5.

2 Setting

In this section, we discuss the setting of our method. We define the following notation: let y_i be the binary outcome variable indicating if there was a sale or not, $\mathbf{x}_i$ the input display features vector, c_i the display cost, v_i the value of a conversion —the CPA the advertiser provided— and N the size of the dataset.

2.1 Logistic regression for CR modelling

Current state-of-the-art CR prediction methods range from logistic regression [6, 13], to log-linear models [2], to a combination of log-linear models with decision trees [8], and to combining pure response rate prediction with ad ranking [11]. In this paper, we use the logistic regression approach from [6] because of the confirmed state-of-the-art results on click prediction together with the relative ease of implementation and the fact that the model learning can be parallelized efficiently. In this case, the objective function to optimize is the L_2 regularized logistic loss:[1]

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{N} \log(1 + \exp(-y_i \mathbf{w} \cdot \mathbf{x}_i)) + \frac{\lambda}{2}\|\mathbf{w}\|^2, \tag{1}$$

with λ a regularization value to be tuned.

2.2 Bidder

We review in this section the way in which the probabilities of clicks and conversions are used for bidding in an online advertiser auction. The setting is as follows. A *bidder* is an agent that competes for an impression that needs to submit a CPM bid to a RTB platform for that impression. It values a certain action—click or conversion—at a certain value v and estimates the probability of the user performing

[1] While writing down the logistic loss, we assume that the labels are in $\{-1, 1\}$ instead of $\{0, 1\}$.

that action if the ad is displayed to be p. The value of the impression is thus $p \times v$ and since most RTB platforms rely on second prices auctions, the bidder uses that value for its bid.

Let c be the highest competing bid in that auction. If that value is smaller than the bid $p \times v$, the bidder wins the auction and pays the second price c. The payoff of the auction can thus be written as:

$$\begin{cases} y \times v - c & \text{if } p \times v > c \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

2.3 Offline Metrics

Let us now detail the recent business-aware offline metrics (Weighted MSE, Utility) that approximate the business impact of a model change.

Weighted MSE For a CR model, the classical mean squared error (MSE) can be interpreted as the offline metric that penalizes the volume of poorly explained observed sales. This metric can be extended to weight the display-level squared error with the CPA of the corresponding advertiser and to therefore penalize the model proportionally to the unexplained revenue[2]— thus yielding a *Weighted MSE* (MSEW).

$$\textbf{MSEW} = \frac{1}{N} \sum_i ((y_i - p(\mathbf{x}_i)) \cdot v_i)^2 \tag{3}$$

Utility On the other hand, the *Utility*[3] metric [5] allows to model offline the potential change in profit due to a prediction model change. Since the observed profit in historical data is fixed, this metric makes the assumption that the display costs are determined by the highest second bids coming from a second price auction and that they are generated according to a distribution conditioned on the observed display cost:

$$\textbf{Utility} = \sum_i \int_0^{p(\mathbf{x}_i)v_i} (y_i \cdot v_i - \tilde{c}) \Pr(\tilde{c} \mid c_i) d\tilde{c} \tag{4}$$

The distribution $\Pr(\tilde{c} \mid c)$ specifies what could have been the second price instead of the observed cost c; [5] suggests a Gamma distribution with $\alpha = \beta c + 1$ and free parameter β. The motivation for selecting this distribution is that it interpolates nicely between two limit distributions: a Dirac distribution centered at c (as $\beta \to +\infty$) and an improper uniform distribution (as $\beta \to 0$). The former limit case boils down to the *empirical* utility (2) while the latter is equivalent to the weighted MSE (3) [9, Theorem 2].

3 Method

As we have seen, there is a discrepancy between the CPA-aware offline metrics and the standard loss functions of the current models, such as the log loss function optimized in the logistic regression. This is suboptimal, as current state-of-the-art models for online bidding suffer from misspecification[4] [9]. We propose the following method to solve this problem.

[2] Because CPA times sales is commensurate to revenue.

[3] This metric is called *expected* Utility in [5], but we refer to it as Utility in this paper.

[4] A regression model is considered *misspecified* when one of the variables is correlated with the error term, both due to omitted variables bias and due to functional form misspecification.

3.1 Connection between Utility and Weighted Log Loss

To the best of our knowledge, the only solution to this problem was proposed in [9], where the authors design a specific loss function (the *Utility loss*) that take into account the bidder economic performance and which inspired the work on the *Utility metric* [5]. However, the Utility loss is non-convex as shown in Figure 1 of [9]. We start by investigating the relationship between the Utility loss and the standard log loss, which is used for training current state-of-the-art CR models, in order to determine whether we could extend the standard log loss to solve the problem at hand.

There are several choices to model the distribution of the highest competing bid $\Pr(\tilde{c} \mid c)$ in (4). A common distribution mentioned in [9] is the log-normal one as it nicely captures the fact that the uncertainty in the highest competing bid should be *relative* to the specific bid. We will use this distribution in this section as it makes the analysis easier. Let σ^2 be the fixed variance of the log normal distribution and $\mu = \log(c) - \sigma^2/2$ chosen in such a way that the mean value is c:

$$\Pr(\tilde{c} \mid c, \sigma) = \frac{1}{\sqrt{2\pi}\tilde{c}\sigma} \exp\left(-\frac{(\log(\tilde{c}/c) + \sigma^2/2)^2}{2\sigma^2}\right).$$

The *Utility loss* is defined as the opposite of the expected Utility:

$$\ell_\sigma(p, y, v, c) := \int_0^{pv} (\tilde{c} - yv) \Pr(\tilde{c} \mid c, \sigma) d\tilde{c}.$$

Of course we cannot make a general connection between the Utility loss and the log loss since the highest competing bid c is involved in the definition of the former, but not in the latter. We can however analyze the behavior of the Utility loss when c is close to our bid pv. Note that we expect most of the auctions to be in that regime. First of all, the derivative of the loss with respect to the prediction p is:

$$\frac{\partial \ell}{\partial p} = v^2(p - y) \Pr(\tilde{c} = pv \mid c, \sigma).$$

Assuming the highest competing bid is equal to our bid, (i.e. $c = pv$), and combining the two previous equations, we get:

$$\frac{\partial \ell}{\partial p} = v^2(p - y) \times \frac{1}{pv} \times \frac{\exp(-\sigma^2/8)}{\sqrt{2\pi}\sigma} \propto \frac{v(p - y)}{p}.$$

Let us now compare the derivatives of the Utility loss and the log loss under the additional assumptions that the probabilities are small ($p \ll 1$), which is typically the case in display advertising.

	$y = 0$	$y = 1$
Log loss	$\frac{1}{1-p} \approx 1$	$-\frac{1}{p}$
Utility loss	v	$\frac{v(p-1)}{p} \approx -\frac{v}{p}$

We observe that the derivatives are approximately equal, up to a factor v. This result motivates the idea of weighting the log loss with the value associated with the sale that we are trying to predict in order to better align the loss used during training and the offline metrics. This can be seen as an extension of the earlier result of [9] which shows that under a uniform distribution of the largest opponent bid, the Utility loss is equivalent to the squared loss weighted by the value.

3.2 Weighted Log Loss

As a result of our findings from Section 3.1, we introduce a weighted negative log likelihood (denoted as WNLL) in the context of online bidding and study its behavior. We define:

$$\text{WNLL} = \sum_{i=1}^{N} v_i \log(1 + \exp(-y_i \mathbf{w} \cdot \mathbf{x}_i)) + \frac{\lambda}{2}\|\mathbf{w}\|^2 \tag{5}$$

where N is the size of the dataset, $\mathbf{x}_i$ is the feature vector, $\mathbf{w}$ is the parameters vector, y_i is the true outcome and λ is the hyper-parameter that controls the importance of the L_2 regularization factor. Each display is weighted by v_i, the CPA of the advertiser associated with the i^{th} display. This is equivalent with generating a dataset where the examples from each advertiser are re-sampled proportionally with its CPA and where all positive examples (sales) have equal economic value, but with the advantage of not incurring an increase in storage and processing time.

Relationship with Utility loss To investigate how this weighting scheme compares to the Utility loss, we use the following toy example. In Figure 1, we plot several losses as a function of a fixed predicted conversion rate on an equal mix of two advertisers with very different conversion rates $p_a = 0.1\%$ and $p_b = 1\%$ (both $p \ll 1$). The CPAs of these advertisers are respectively 50 and 5 and the second prices follow a uniform distribution on $[0.04, 0.06]$. With this setting the advertiser optimal profits and the associated empirical losses are equal and close to zero (to simulate the $c \approx pv$ regime introduced above).

The Utility loss here and in the rest of this paper is the same as the one defined in [5] where the highest competing bid $\Pr(\tilde{c} \mid c)$ distribution follows a Gamma distribution with $\alpha = \beta c + 1$ and free parameter β. As β goes to infinity, the distribution goes to a Dirac distribution centered around c and the Utility loss converges to the *empirical utility* (2). We set here $\beta = 30$.

The figure shows that the Utility loss has the same optimum point with the empirical utility and that the log loss weighted by CPA (WNLL) has a minimum much closer to the empirical loss q^* that the un-weighted log loss (NLL): $q^* = 0.1\%$, $q^*_{WNLL} = 0.18\%$, $q^*_{NLL} = 0.55\%$.

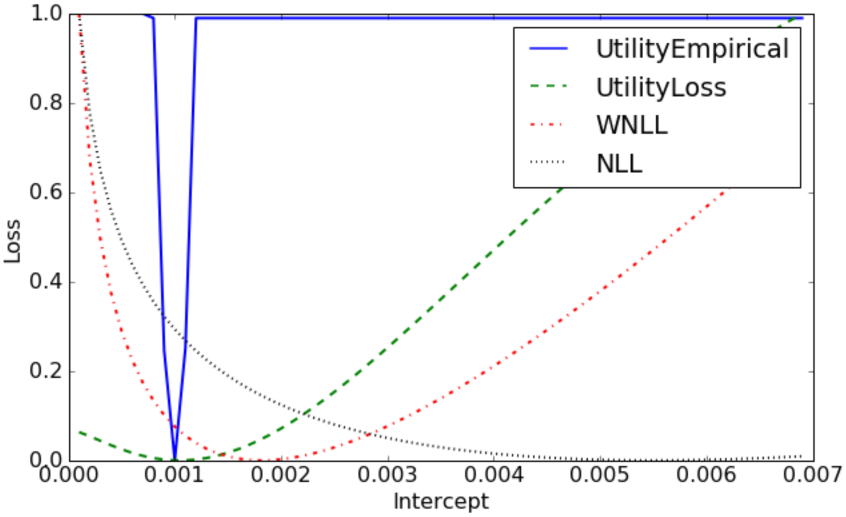

Figure 1: Weighted log loss vs. log loss and Utility loss for an intercept-only model on a synthetic dataset with two advertisers.

3.3 Impact of weighting on learning

In the following, we analyze the impact of moving from a standard log loss to weighted log loss both from the perspective of learning and of regularization.

Learning with importance weights We analyze the learning setup proposed in [6] where limited memory BFGS [12, 14] (L-BFGS) is warm-started with stochastic gradient descent [3] (SGD). For both algorithms, we multiply the gradient of the loss of each example by v, where v is the weight associated with the example. This is straightforward to implement. Note that exact importance weighting for SGD is more tricky and that dedicated weighting schemes exist [10]. However, in our case, we use SGD only for warm-starting L-BFGS and our approximate method of including importance weights proved to be sufficient.

Impact on the regularization parameter In the case of switching from log loss to the weighed log loss, the value of the λ hyper-parameter for NLL needs to be adapted to WNLL. To do that, we use the following simple rule that adapts λ depending on the value of the importance weights used, i.e. of the average CPA of each advertiser:

$$\lambda_{WNLL} = \lambda_{NLL} \times \frac{\sum_i v_i}{N} \tag{6}$$

For our experiments we use the following heuristic to set the value of λ in the un-weighted case, as suggested in [4**?**]:

$$\lambda^h_{NLL} = \frac{1}{n} \sum_{i=1}^{n} \|x_i\|_2^2 \tag{7}$$

where x_i represents a training instance vector and n is the size of the dataset.

We show how well the lambda rescaling scheme performs relative to other values of λ in the context of L_2 regularization in Table 2 in Section 4.

4 Experiments

In this section, we present our experimental results when applying our method to improve a state-of-the-art conversion-rate prediction model for bidding in online advertising auctions. We present offline results followed by online experiments on live traffic.

4.1 Offline results on public dataset

For comparing WNLL and NLL, we used a public dataset released by Criteo as supporting material of [4]. The dataset contains a sample of post-click conversions with a matching window of 30 days. For simplicity, we choose the setup (denoted by the authors as the ”oracle” setup), where we use for training the entire set of positive examples (clicked displays that converted within the next 30 days) and apply the model without leaving a 30 days window for evaluation, as it would be needed in a live system.

So far in the paper we have assumed a model predicting the conversions at the display level. Since the records in this public dataset are at the click level, we will instead consider in this section the task of predicting a probability of conversion given click. In order for this to be used in a production system the post-click probability would need to be further multiplied by a probability of click given display, as explained in [4].

Weighting	MSEW		Utility$_{\beta=10}$		Utility$_{\beta=1000}$	
	Train	Test	Train	Test	Train	Test
CPA	-50.45 +/- 0.91	**-19.57 +/- 0.65**	1.44 +/- 0.02	**0.37 +/- 0.04**	1.29 +/- 0.03	0.18 +/- 0.08
$CPA^{\frac{1}{2}}$	-36.84 +/- 0.67	-14.57 +/- 0.49	0.91 +/- 0.01	0.32 +/- 0.02	0.89 +/- 0.03	**0.30 +/- 0.04**
$CPA^{\frac{1}{4}}$	-24.54 +/- 0.46	-9.26 +/- 0.3	0.51 +/- 0.01	0.18 +/- 0.01	0.51 +/- 0.02	0.19 +/- 0.03

Table 1: Overfitting in WNLL as a function of the CPA weighting scheme. The best performing result for each metric in terms of relative improvement over NLL is indicated in bold.

Since the objective of the dataset was to show empirically the importance of modeling delayed conversions, the associated display costs and conversion revenue are not included. To be able to evaluate our method, we introduce a simplified cost and revenue scheme where all clicks have a constant cost of 1, and for each advertising campaign the CPA is inversely proportional with the historical post-click conversion rate, meaning that each advertising campaign c is assumed to be contributing equally to the overall revenue: $cost_c = 1, CPA_c = \frac{1}{AvgCR_c}$ and $AvgCR_c$ is the campaign average CR with smoothing as explained below.

For the experimental results, we compare the results of the baseline log loss (NLL) and the results of the weighted log loss (WNLL). We take a 2 weeks period (weeks 3 and 4) as the test period (each model is trained on a period of up to three weeks and used to predict the conversion rate on next day traffic). For the historical CR estimate, we use the 2 weeks period before the first test day (weeks 1 and 2) to compute average conversion rates for the campaigns. To handle the case of new campaigns appearing during the testing period, we set the final campaign CR estimator to be: $SmoothCR_c = \frac{\#sales_c + AvgCR}{\#clicks_c + 1}$ where $AvgCR$ is the overall average CR computed in the two weeks and is equal to 0.23.

All the results presented in this section are provided with confidence intervals computed using bootstrap. The Utility loss results are computed using a Gamma distribution with free parameter β as in [5].

The impact of CPA weight factor on performance. As previously covered in the cost-sensitive learning literature [7], associating big costs to the training examples can lead to overfitting. We have established experimentally on our internal traffic that the best solution to combat overfitting is reducing the amplitude of the final weights used during model learning by a combination of capping the maximum values of the CPAs and a magnitude reduction function (sqrt) on the CPA. In Table 1 we show the relative performance of various CPA dampening schemes on the open dataset, for a maximum value of CPA =20. We see that the amount of overfitting in terms of MSEW is comparable across all of the three weighting schemes, but that $CPA^{\frac{1}{2}}$ overfits less in terms of Utility$_{\beta=1000}$ and has better performance on the test set.

Impact of λ We benchmark λ values around the value λ_h produced by the heuristic proposed in Section 3.3. We observe in table 2 that the proposed λ is very close to the optimal value. This finding mirrors the results obtained on internal data. For this reason, we use the λ proposed by the heuristic in all our experiments.

WNLL vs. NLL on high CPA In Table 3, we report performance of our best WNLL setup (with weighting scheme $CPA^{\frac{1}{2}}$) on the evaluation metrics that are closest to the actual business metrics, e.g. MSEW and Utility. We observe that the biggest lift of WNLL vs. NLL is on the advertising campaigns with high CPA and a low number of sales (<30) in the period of reference (weeks 1 and 2). This confirms our hypothesis that WNLL should outperform NLL on campaigns with low volume of sales, but high CPA. Because of that, the actual economic impact of switching to WNLL could be even greater, depending of

Λ	Utility$_{\beta=1000}$
$\lambda_h - 40\%$	0.34 +/- 0.09
$\lambda_h - 20\%$	0.32 +/- 0.05
$\lambda_h - 10\%$	0.29 +/- 0.05
λ_h	**0.30 +/- 0.04**
$\lambda_h + 10\%$	0.33 +/- 0.03
$\lambda_h + 20\%$	0.21 +/- 0.03

Table 2: Comparison of different values of λ around the heuristic value λ_h computed by the method covered in Section 3.3.

Metric	Global	HighCPA(>10)	HighCPA(>10), LowSales(<30)
MSEW	-14.57 +/- 0.49	-15.26 +/- 0.53	-22.36 +/- 1.02
Utility$_{\beta=10}$	0.32 +/- 0.02	0.78 +/- 0.06	1.70 +/- 0.16
Utility$_{\beta=1000}$	0.30 +/- 0.04	0.78 +/- 0.11	1.72 +/- 0.21

Table 3: Relative improvement of WNLL vs. NLL on the open Criteo dataset, both globally and for advertising campaigns with high CPA.

the relative proportion of traffic with high CPA and low number of positives. As we will show next, our online experiments confirm this finding.

4.2 Online experiments

We ran an A/B test of the change of the loss function to WNLL in the conversion-rate model. The A/B test was done on more than 1 Billion ad displays, on world-wide traffic. We observed significant savings in display cost, coupled with an increase in sales performance for the advertisers. Our change resulted in a +2% lift in performance, which is considerable compared to typical improvements in this field. In terms of development and operational costs, the change in the loss function took only a couple of weeks to put in production, since the code change is minimal, as shown in Section 3. Furthermore, the training time of the model did not change.

5 Conclusion

We investigated the relationship between the Utility loss and the standard log loss. This analysis motivates the idea of weighting the log loss with the value associated with the sale that we are trying to predict (CPA) in order to better align the loss used during training and the offline metrics. Then, we presented and analyzed a cost weighting scheme that takes into account the advertisers' CPAs when training our CR model for bidding in online advertising auctions and discuss its impact on learning and regularization. We showed that this cost weighting scheme leads to a loss function, whose optimal point is much closer to the optimum point (reached by optimizing the Utility loss) than the one of the standard log loss. We finally demonstrated that our method allows to improve a state-of-the-art CR prediction model used for bidding in online advertising auctions and leads to large significant lifts in offline performance (on a public data set) and online performance (through an A/B test at a major advertising company).

Future work As future work, we plan on investigating two directions. First, optimize directly the Utility loss [9], which is non-convex and thus requires careful optimization. Second, use a convex approximation of the Utility loss as proposed in [9, Section 6].

Acknowledgments We would like to thank Vianney Perchet, Nicolas Le Roux, Olivier Koch, Etienne Sanson, Cyrille Dubarry, Alexandre Gilotte and Dmitry Pavlov for their useful comments on early versions of this paper.

6 References

[1] Digital advertising spend in 2015. `http://www.statista.com/statistics/237974/online-advertising-spending-worldwide/`. Accessed: 2015-10-15.

[2] D. Agarwal, R. Agrawal, R. Khanna, and N. Kota. Estimating rates of rare events with multiple hierarchies through scalable log-linear models. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–222. ACM, 2010.

[3] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT '10*, pages 177–186. Springer, 2010.

[4] O. Chapelle. Modeling delayed feedback in display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1097–1105. ACM, 2014.

[5] O. Chapelle. Offline evaluation of response prediction in online advertising auctions. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 919–922. International World Wide Web Conferences Steering Committee, 2015.

[6] O. Chapelle, E. Manavoglu, and R. Rosales. Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(4):61, 2014.

[7] C. Cortes, Y. Mansour, and M. Mohri. Learning bounds for importance weighting. In *Advances in neural information processing systems*, pages 442–450, 2010.

[8] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1–9. ACM, 2014.

[9] P. Hummel and R. P. McAfee. Loss functions for predicted click through rates in auctions for online advertising. *Preprint, Google Inc*, 2013.

[10] N. Karampatziakis and J. Langford. Online importance weight aware updates. *arXiv preprint arXiv:1011.1576*, 2010.

[11] C. Li, Y. Lu, Q. Mei, D. Wang, and S. Pandey. Click-through prediction for advertising in twitter timeline. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1959–1968. ACM, 2015.

[12] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[13] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013.

[14] J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35 (151):773–782, 1980.

[15] S. Yuan, J. Wang, and X. Zhao. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, page 3. ACM, 2013.