Cost-sensitive Learning for Bidding in Online Advertising Auctions

Flavian Vasile[1] **and Damien Lefortier**[1,2]
[1] Criteo, Inc.
[2] University of Amsterdam
{f.vasile, d.lefortier}@criteo.com

Abstract

One of the most challenging problems in computational advertising is the prediction of ad click and conversion rates for bidding in online advertising auctions. State-of-the-art prediction methods include using the maximum entropy framework (also known as logistic regression) and log linear models. However, one unaddressed problem in the previous approaches is the existence of highly non-uniform misprediction costs. In this paper, we present our approach for making cost-sensitive predictions for bidding in online advertising auctions. We show that one can get significant lifts in offline and online performance by using a simple modification of the logistic loss function.

1 Introduction

Online advertising is becoming a big 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-action/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 an RTB (real-time bidding) platform. Most of the RTB platforms function using a 2nd price model [2], where advertisers or agents representing the advertisers bid for display opportunities, and the winner pays the maximum between the bid of the second participant 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 gains (also known as eCPMs) using click and conversion rate (CR) prediction models. One differentiating factor between the click and conversion models, is that for conversion (sale) prediction models, the ratio of positives to negatives is much smaller, thus making the learning problem more challenging. For this reason, we will concentrate our attention on improving the performance of conversion models.

One important aspect of the marketplace is that the numeric range of the possible CPAs is very large and it depends on the economic value of the action that the advertiser is trying to incentivize for. The resulting eCPMs vary from ones based on expectations over actions that are frequent and low-value (e.g. conversions on classified ads[1]) and ones that are extremely rare and high-value (e.g. successful mortgage applications). Importantly, 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 bidders performance are making use of the advertisers' CPAs [3, 4].

Recent business-aware offline metrics Indeed, taking the example of a conversion-rate predictor, the classical mean squared error (MSE) can be interpreted as the offline metric that penalizes the raw 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 with the unexplained revenue (mentioned as a special case in [4]) — thus yielding a *Weighted MSE* (MSEW). Another example is the *Utility* metric, recently proposed in [4], which takes into account both the potential upside of the display represented by the CPA, but also the associated display cost (modeled as a Gamma distribution over potential display costs given the observed cost). This metric can be interpreted as the offline metric counterpart of the change in profit. We detail all these metrics in Section 3.

Shortcomings of current approaches Current state-of-the-art response rate prediction methods range from logistic regression [5, 6], to log-linear models [7], to a combination of log-linear models with decision tress [8], and to combining pure response rate prediction with ad ranking [9]. We see a discrepancy between the CPA-aware offline metrics and the standard loss functions of the current prediction models, such as the log loss function optimized in the logistic regression. As a result, we focus on the following questions: *Can our predictive models benefit from taking into account the sales' CPAs? How to better add this information during learning in order to improve the performance of our models?* To the best of our knowledge, the only solution to this question was proposed recently in [3], where the authors design specific loss functions that take into account the bidder economic performance and which inspired the work on the *Utility* metric in [4]. By comparison, we investigate the use of the well-established *cost-sensitive learning framework* to tackle this task.

Proposal The learning framework where the misclassification costs vary across examples is called *example-dependent cost-sensitive learning*. In [10], the authors classify the cost-sensitive learning approaches in three main classes as follows. The first approach makes learning cost-sensitive by adding costs to the learner — this works for any model that falls under the statistical query model [11]. The second approach changes the final decision function and assigns each example to its lowest cost-sensitive risk [12]. Finally, the third approach changes directly the training dataset by duplicating each example by a factor proportional to its relative cost, which has the advantage that it works with any error minimizing classifier [10]. Our proposal falls under the first approach, where we alter the loss function to take into account the relative misprediction costs. One of the main reasons for our choice was the relative ease of implementation in our current system.

In Section 2, we present our method for taking into account the advertisers' CPAs using cost-sensitive learning and discuss its impact on learning and regularization. In Section 3, we present the experimental results when applying our method to improve a state-of-the-art conversion-rate prediction model used as a sub-model for predicting either the expected number of sales or the expected sales amount generated by a user following a display for bidding in online advertising auctions. We present both offline experiments using a large real-world dataset collected at Criteo and online experiments on live traffic through an A/B

[1] Personal ads that usually contain personal services advertising, consumer-to-consumer home renting and home selling offers.

test. We show that our method brings statistically significant improvements offline with a large effect size on our metrics. We also show that our method brings significant gains in online performance.

2 Cost-sensitive Logistic Regression

2.1 Cost-sensitive learning and Risk Minimization

As discussed in [3], current state-of-the-art models for online bidding suffer from "misspecification"[2], both due to omitted variables bias and due to functional form misspecification. To improve the performance of models under misspecification, we show how to align better the loss function with the actual offline evaluation metrics using a cost-sensitive approach. In order to extend the logistic regression to cost sensitive tasks, we change the original loss function (NLL) to weighted negative log likelihood (denoted as WNLL).

$$WNLL = \frac{1}{N}\sum_{i}^{N}(-y_i \log(p_i) - (1-y_i)\log(1-p_i)) * c_i \tag{1}$$

where N is the size of the dataset, y_i is the true outcome, p_i is the prediction and c_i is the cost associated with instance i.

Advertiser-constant revenue weighted logistic loss We propose a simple cost weighting scheme that weighs each display by the CPA of the corresponding advertiser: $c_i = CPA_i$ for all displays $i = 1..N$, where CPA_i is 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:

$$WNLL_{CPA} = \frac{1}{N}\sum_{i}^{N}(-y_i \log(p_i) - (1-y_i)\log(1-p_i)) * CPA_i \tag{2}$$

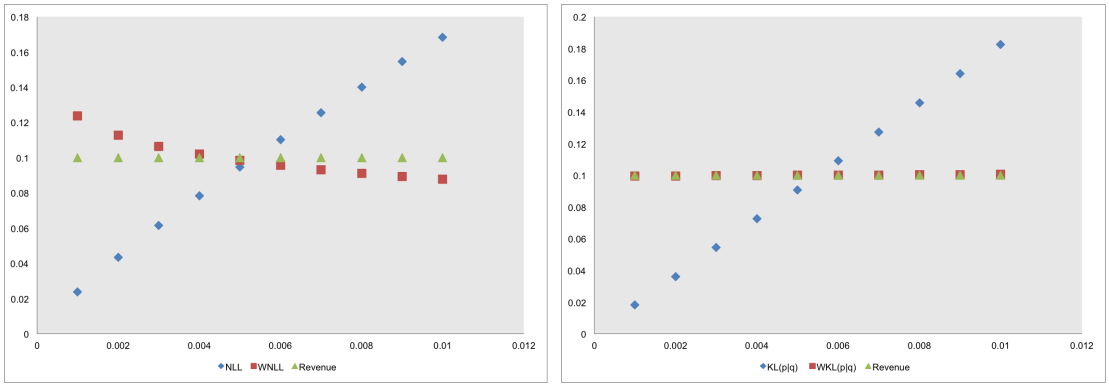
The $WNLL_{CPA}$ loss is *calibrated*, i.e. is minimized by predicting the true conversion rate of the ad.

Relationship with $Utility$ and $MSEW$ Minimizing $MSEW$ is equivalent with maximizing expected profit (a.ka. $Utility$) when display costs are uniform (see [4, 3]). By analogy, $WNLL_{CPA}$ is the logistic counterpart of $MSEW$ and in Section 3 we show that, empirically, it has good performance when evaluated using both $MSEW$ and $Utility$.

The alignment of NLL and $WNLL_{CPA}$ with ad revenue We expect that the weighted loss should align more with the revenue than the original log loss. In order to check that, we simulate the impact of a constant over-prediction of the true conversion rate: $q = p \times 1.2$ on the NLL and $WNLL_{CPA}$ as a function of an increasing p and decreasing associated CPA: $p_i \in [0.1\%, 1\%]$ and $CPA_i = 1/p_i$. We assume equal display traffic for all i, so each i has equal revenue: $Revenue_i \propto CPA_i \times p_i$.

We plot in Figure 1 the per display expectations of: NLL, $WNLL_{CPA}$ and revenue, without (a) and with (b) rescaling by substracting $H(p)$, the entropy of the true conversion rate (thus obtaining the weighted and un-weighted KL divergences of p and q: $D_{KL}(p|q) = H(p,q) - H(p)$). We rescale by $H(p)$ in order to make the expectations of the loss due to over-prediction comparable across different CR levels. As expected, we observe that the un-weighted loss puts more mass on higher CR areas, having no relationship with the expected revenue, but that the weighted loss is extremely well-aligned with the expected revenue.

[2] A regression model is considered "misspecified" when one of the variables is correlated with the error term.



(a) (b)

Figure 1: The plots of the expectations of NLL, $WNLL_{CPA}$ and revenue over a CR range, without (a) and with (b) rescaling by substracting $H(p)$.

2.2 Impact of weighting on learning with SGD and L-BFGS

Let us now discuss how to do learning in the case of cost-sensitive logistic regression, i.e. using WNLL_{CPA} as loss function, using two well-known learning algorithms: limited memory BFGS [15] (L-BFGS) and stochastic gradient descent [16] (SGD). These two algorithms are widely used for learning click and conversion prediction models in the context of display advertising (see, e.g., [9, 8] for SGD, and [5, 17] for L-BFGS initialized with SGD). Using an advertiser-constant revenue weighting scheme is equivalent to defining an *importance weight* for each display (see above). In our case, the weight is the average CPA of the advertiser associated with each display.

2.2.1 Learning with importance weights

Batch case: L-BFGS

L-BFGS is a batch algorithm (as, e.g., gradient descent), which means that one pass over the data is required before each update of the weights. In this case, using an importance weight for each example during the learning is straightforward [18]. Indeed, in batch algorithms, we typically first compute the sum of the gradients of all examples before updating the weights once, and adding x time the gradient of an example or adding x times the gradient once is the same when computing this sum. So we can simply do that, i.e. multiplying the gradient by x – which is both easy and inexpensive too add in the code of L-BFGS. Note that this is not an approximation.

Online case: SGD

On the other hand, SGD is an online, or streaming algorithm, which means that an update of the weights is performed after seeing each example. In this case, adding an importance weight of x is not equivalent to multiplying the gradient by x when doing the update as in the batch case covered above [19]. Indeed, if we would have an example duplicated x times in the training set, we would do x independent updates, which is not equivalent with doing a single update that is x times larger [19]. We could therefore do x successive updates (and multiply the gradient by the fractional part if greater than 0 as CPA_i is not necessarily an integer), which is less approximate but not efficient (although, ideally, these updates would be done at different times during the learning and not successively as the training set is often randomly

shuffled at the beginning). [19] also proposes an approximated way to perform importance weight aware updates without doing x updates, which is time-consuming, and better than multiplying by x.

Here, we simply multiply the gradient with the importance weight CPA_i, which is an approximate solution, but straightforward to implement. As discussed above we use SGD only for warm-starting L-BFGS and it seems that, in this case, even an approximate method of including importance weights is sufficient (see Section 3). Our experiments, excluded for brevity, show that this method works as well as the method of doing CPA_i successive updates in our setting. Since this method showed no additional value, there was no benefit in exploring the method from [19].

2.2.2 Impact on the regularization parameter

In most applications, the logistic regression objective function is accompanied by a regularization term Ω on the weights w (typically, L1 or L2 regularization terms) to prevent overfitting the training data. We obtain the following loss function for regularized WNLL_{CPA}:

$$L = \frac{1}{N}\sum_{i}^{N}(-y_i \log(p_i) - (1-y_i)\log(1-p_i)) * CPA_i + \lambda * \Omega(w)$$

where λ is a hyper-parameter to tune called the regularization parameter. In this paper, we use L2 regularization where $\Omega(w) = \|w\|_2^2$. Importantly, when doing cost-sensitive logistic regression, this hyper-parameter λ needs to be adapted (as, e.g., when adding new features). To do that, we use the following simple heuristic to adapt λ depending on the value of the importance weights used, i.e. of the average CPA of each advertiser:

$$\lambda_{WNLL_{CPA}} = \lambda_{NLL} * \frac{\sum_i CPA_i}{N}$$

We investigated how well this heuristic performs in comparison with the optimal value of λ (for results, see Section 3).

3 Experiments

In this section, we present our experimental results when applying our method to improve a state-of-the-art conversion-rate prediction model used as a sub-model for predicting either the expected number of sales or the expected sales amount generated by a user following a display for bidding in online advertising auctions. First, we present offline experiments using a data set of more than 3 billion examples with binary labels (sale vs. no-sale) and hundreds of millions of unique attribute values from the production click logs of Criteo from March 2015. Then, we present online experiments on live traffic.

3.1 Offline metrics

For the offline evaluation of our WNLL_{CPA} model, we use offline metrics that approximate the business impact of the model change, namely *Normalized Weighted MSE* (NMSEW) and *Utility* (discussed in Section 1). The *Normalized Weighted MSE* shows the relative improvement in MSEW of the model to be evaluated versus a baseline predictor, in our case the average empirical CR rate of the dataset, similar to the normalization in [8, 17]. We denote s_i the binary outcome variable indicating if there was a sale or not, x_i the input display features vector, and c_i the display cost. In order to model offline the potential change in profit due to a prediction model change, the *Utility* measure has been proposed recently in [4]. Since the observed profit in historical data is fixed, the metric makes the assumption that the display costs

are coming from a second price auction and that they are generated according to a Gamma distribution conditioned on the observed display cost: $\mathbb{P}(\tilde{c}|c) \propto \tilde{c}^{\beta c} \exp(-\beta\tilde{c})$ (free parameter β).

Weighted Mean Squared Error (MSEW) $\frac{1}{N}\sum_i((s_i - p(x_i)\cdot CPA_i)^2$

Normalized Weighted Mean Squared Error (NMSEW) $1 - \frac{MSEW_{Model}}{MSEW_{AverageEmpiricalCR}}$

(Expected) Utility $\sum_i \int_0^{p(x_i)*CPA_i} (s_i \cdot CPA_i - \tilde{c})\mathbb{P}(\tilde{c}|c_i)d\tilde{c}$

3.2 Offline results

We compare the difference in performance between the standard logistic regression NLL and the weighted version WNLL_{CPA} using L-BFGS initialized with SGD [5, 17]. Given the fact that our cost-weigthing scheme re-weights each advertiser's negative and positive examples by a constant, the resulting conversion probabilities remain calibrated and we can make the loss function change only in the conversion-rate model, while keeping the rest of the systems unchanged. For the regularization parameter λ, our experiments show that using our heuristic (Section 2.2.2) allows us to reach results that are comparable with the best λ (tuned manually), as shown by Figure 2. We see that the optimal lambda value increases the performance on high CPA traffic, but decreases the performance on other areas of traffic (overall, the heuristic lambda is within noise of the optimal one).
In our experiments, we therefore use the updated regularization parameter λ heuristic, which has the great benefit of avoiding to re-tuning λ every time the costs, i.e. CPA_i, change:

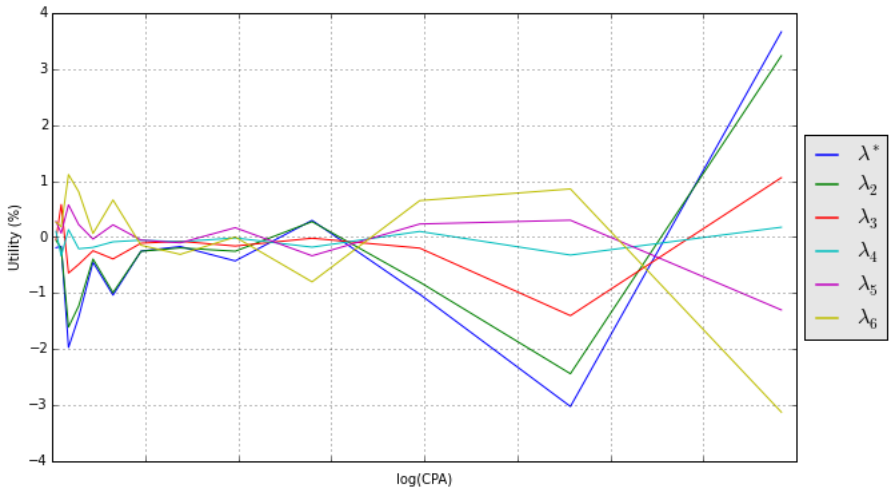


Figure 2: The relative lift in performance of $WNLL_{CPA}$ in terms of $Utility(\beta_2)$ as a function of the regularization parameter λ with respect to $\lambda_{heuristic}$

The results in Table 1 are divided over two optimization tasks used at Criteo that aim to respectively optimize the number of sales and the total value of sales for each advertiser. We observe that WNLL_{CPA} performs significantly better than NLL on all metrics and with a higher magnitude than typically observed model improvements.

	$NMSEW$	$Utility(\beta_1)$	$Utility(\beta_2)$
Number of Sales	+(11.34, 13.28)%	+(5.66, 6.43)%	+(4.27, 5.43)%
Sales Amount	+(7.03, 10.39)%	+(7.79, 7.97)%	+(1.53, 1.77)%

Table 1: Weighting the Logistic Regression: NLL vs. WNLL_{CPA} on the test set; ($\beta_2 > \beta_1$)

3.3 Online experiments

We ran an A/B test of the change of the loss function to WNLL_{CPA} in the conversion-rate model for both number of sales prediction and total value of sales prediction and we observed significant savings in display cost, coupled with an increase in sales performance (number of sales, total sales amount) for the advertisers. Our change therefore resulted in a significant positive impact on the projected long-term revenue. 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 2. Furthermore, the observed training time of the model did not change.

4 Conclusion

In this paper, we investigated the impact of applying the cost-sensitive learning framework in the context of bidding in online advertising auctions. We introduced a problem-specific weighting scheme for the logistic loss that we denoted WNLL_{CPA}, which weights differently displays (and their associated sales) from different advertisers, based on their expected value, namely their CPA. We applied this change to a state-of-the-art conversion-rate prediction model used as a sub-model for predicting either the expected number of sales or the expected sales amount generated by a user following a display. Compared with the vanilla version of the logistic loss, we showed large improvements in offline metrics (*Normalized Weighted MSE*, *Utility*), followed by meaningful impact on the business metrics in the live bidding system.

In the next steps, we will continue our research in two different directions: first, we plan to compare our current approach with other cost-sensitive learning methods, such as the "costing" approach [10]. As a second direction, we will look into different problem-specific weighting schemes, such as weighting-by-profit and weighting with product-level cost estimators.

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

References

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

[2] B. Edelman, M. Ostrovsky, and M. Schwarz, "Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords," tech. rep., National Bureau of Economic Research, 2005.

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

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

[5] O. Chapelle, E. Manavoglu, and R. Rosales, "Simple and scalable response prediction for display advertising," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 4, p. 61, 2014.

[6] 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*, pp. 1222–1230, ACM, 2013.

[7] 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*, pp. 213–222, ACM, 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*, pp. 1–9, ACM, 2014.

[9] 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*, pp. 1959–1968, ACM, 2015.

[10] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pp. 435–442, IEEE, 2003.

[11] M. Kearns, "Efficient noise-tolerant learning from statistical queries," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 983–1006, 1998.

[12] C. Elkan, "The foundations of cost-sensitive learning," in *International joint conference on artificial intelligence*, vol. 17, pp. 973–978, Citeseer, 2001.

[13] J. P. Dmochowski, P. Sajda, and L. C. Parra, "Maximum likelihood in cost-sensitive learning: Model specification, approximations, and upper bounds," *The Journal of Machine Learning Research*, vol. 11, pp. 3313–3332, 2010.

[14] F. Provost and T. Fawcett, "Robust classification for imprecise environments," *Machine learning*, vol. 42, no. 3, pp. 203–231, 2001.

[15] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[16] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *COMPSTAT '10*, pp. 177–186, Springer, 2010.

[17] D. Lefortier, A. Truchet, and M. de Rijke, "Sources of variability in large-scale machine learning systems," in *Machine Learning Systems (NIPS 2015 Workshop)*, 2015.

[18] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of computation*, vol. 35, no. 151, pp. 773–782, 1980.

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