Modeling labels for conversion value prediction

Ashwinkumar Badanidiyuru
ashwinkumarbv@google.com
Google Research
USA

Guru Guruganesh
gurug@google.com
Google Research
USA

ABSTRACT

In performance based digital advertising, one of the key technical tools is to predict the expected value of post ad click purchases (a.k.a. conversions). Some of the salient aspects of this problem such as having a non-binary label and advertisers reporting the label in different scales make it a much harder problem than predicting probability of a click. In this paper we ask what is a good way to model the label and extract as much information as possible from the features. We investigate three main issues that arise from advertiser reported labels and come up with new techniques to address them. The first issue is that the label scale can affect how the model capacity is devoted to different advertisers. The second issue is how outlier labels can cause over-fitting. Finally, we also show that the distribution of the label contains vital information and the we train our models to use them and not just rely on the mean.

KEYWORDS

Conversion, Ads, Value prediction, Label

1 INTRODUCTION

The holy grail of advertising is to maximize its' net effect, while simultaneously spending as little as possible. Over many decades, the advertising industry has repeatedly changed to be able to achieve this. Historically, advertising included print and billboard messaging, and its effectiveness was measured via surveys. These were at best good proxies. While the introduction of radio and television increased the overall reach and effectiveness, they still suffered from the same measurement issues as previous forms. With the advent of internet advertising, we are able to measure engagement through more direct proxies or even end goals.

The most popular form of advertising on internet was based on "pricing per view" impression which is most suitable for "Brand" advertising. A breakthrough change was the introduction of pay-per-click ads where the advertiser only pays if the user clicks on an ad([24]). This was substantially more effective as users who click on the ad are typically more valuable for the advertiser. This also allows for better optimization in showing relevant ads as one can predict click through rates. While pay-per-click ads are indeed better for performance advertising, it still suffered from some nontrivial challenges. These challenges include advertisers manually specifying a bid for each segment of the population which is computed painstakingly, lack of automation due to manually specifying which users to target and paying for users who are unlikely to buy on advertisers website.

Over the last few years, the industry has shifted towards optimizing end goals. In this case, advertisers report conversions (purchase events etc) which happen on their website back to RTB (real time bidding) platforms along with an associated value for these conversions (see [1]). In a CPA (cost per acquisition) product, there is a model to predict the probability of a conversion per click which is used to proportionally adjust bids for these advertisers. In a ROAS (return on advertiser spend) product there is instead a model to predict expected value of conversions per click which is also used to proportionally adjust bids for these advertisers; i.e. in both cases, bid proportionally higher on users whose prediction is higher.

We observe that conversions and their purported value are both advertiser reported data and are not observed by the search engine directly. As a result a number of issues that present themselves in this setting that are not present in more traditional models which predict for e.g. click through rates. As a result, building a machine learning model to predict expected value turns out to be a lot more challenging than building a model to predict the probability of a click or a conversion. In this paper, our main goal is to construct a highly accurate model which predicts the advertiser reported value.

The first salient issue in value prediction is that the label reported by each advertiser is in an arbitrary (but consistent) scale. As an example one advertiser can report labels in the range of 10000, while another in the range of 0.00001. This doesn't necessarily mean that dollar value generated per conversion by the first advertiser is larger than the second advertiser. As a result, if we train a generalized linear regression model, the model capacity is allocated disproportionately to the first advertiser in the above example (and more generally to advertisers who report labels in larger scale). Our first idea is to normalize the labels such that the average label is 1 across different advertisers.

The resulting normalized label still has two important issues. First, the range of normalized label is still quite wide due to the presence of many outliers. Secondly, the distribution is biased towards having many more zero-valued labels than any other. To handle the outliers, we take inspiration from robust statistics and utilize a technique known as winsorized mean and merge it with multi-task learning. To fix the excess of zero's in the conversion data, we take inspiration from the zero inflated models. Traditionally one would incorporate these new ideas into a single objective. One of our main ideas is to instead create new auxilary objectives and use this multi-objective approach to learn these properties of the resulting distribution.

Our final observation is while predicting the expected value, most standard models just predict the mean of the distribution and neglect the information in the complete distribution. We further use multi-task learning to predict different properties of the distribution which further improves accuracy of predicting label. In particular, by asking the model to solve an additional classification task, we find that the model quality improves further. Note that this is not like the use of reducing a regression task into a classification task but rather adding new tasks which are only used in training and not used in computing the final prediction. The use of multi-task learning here is not like

the classical use where we train on different tasks (for example CTR and CVR prediction). The additional tasks don't correspond to any other natural product objective and solely drive performance gains.

2 RELATED WORK

Internet advertising has a very rich literature. This started with a long line of work on models for predicting clicks [25, 27, 34] and continued with topics such as designing auctions [4, 12, 30]. There also have been a series of papers studying conversion prediction in [3, 21, 26, 28]. While conversion prediction has several aspects common with classical click prediction, it also has many unique aspects such as delayed label [7], attribution [10] and computing the causal effect [22].

Another line of work which is quite relevant is on loss functions. In these works (see [6, 14, 29, 32]), there is a complete characterization of proper scoring rules – i.e. loss functions which result in unbiased estimators. At a very high level, these loss functions are exactly the functions whose gradient $\partial loss/\partial parameters = f(prediction)\cdot(prediction-label)$ for some reasonably "nice" function f. These loss functions have gradient 0 (in expectation) when the prediction is exactly the (expected) label. Throughout this paper, we will use Poisson regression which belongs to the class of proper scoring rules. This is quite standard and is used for predicting positive float labels throughout literature (see [2]). We won't be concerned with other class of loss functions. Note that while Poisson regression can be derived using Maximum likelihood for integer labels, the loss function is well defined for arbitrary positive real valued labels and gives an unbiased estimator irrespective of the underlying distribution that the label comes from.

A second line of work has been on mean estimation with heavy tails distributions, both in the setting of i.i.d random variables and in the setting of regression. An excellent survey of existing techniques and relevant references can be found in [23]. The setting relevant for us is that of regression and they describe and analyze median of mean tournaments. Today a number of different techniques have been developed in the literature. Modern techniques such as [9, 15, 18] use more sophisticated tools and produce robust estimators but are not very practical.

A third line of work is on using information in the distribution of label and not just predicting the mean of the distribution. One example of this is the work on Zero-Inflated Poisson (ZIP) regression [17, 20, 33] which assumes that the label is generated via a mixture distribution where we generated 0 with probability p and a Poisson random variable with probability 1-p. There are two challenges with this approach, one being that a ZIP model is not a proper scoring rule and hence might not give an unbiased estimator and the second being that it works only for integer labels and isn't defined for float labels.

A fourth line of work which is very relavant is how offline accuracy relates to final business metrics. While traditional machine learning research used various metrics for comparing offline accuracy of models such as log likelihood, l2 error etc, there was a need for better evaluation of models used in internet advertising due to how they can affect final metrics such as revenue. This was studied in [8, 16]. This was later extended by [31] to also allocate model capacity for different advertisers to optimize for final metric. The solution in [31] is to weight the examples by CPA and works for binary label. This work is orthogonal to our work and can work in conjunction to the label normalization, where can do CPA weighting on top of label normalization. But we won't be touching upon this topic in the rest of this paper.

3 PRELIMINARIES

In this section we list the notation used in this paper and also discuss offline evaluation metrics.

Notation. We will use X to denote the set of features used for prediction. For each click the advertiser reports to RTB a set of post click conversions denoted by $C = \{c_1, c_2, \ldots\}$. For each conversion c_i the advertiser also reports a corresponding value ℓ_i. Hence the total value of the click for the advertiser is $\sum_i \ell_i$ which we denote by ℓ. We will denote the advertiser by A which is also a feature included in the general set of features X, i.e $A \in X$. The goal of the prediction problem is to predict the quantity $E[\ell|X]$. Let cpc be the cost per click that advertiser needs to pay for that specific click.

Regression formulation as Poisson regression. There are several ways to predict a real valued random variable and we use one of the most popular formulations to predict this value. In particular, we model it as a Poisson regression task. Poisson Regression is a type of generalized linear regression model, where the corresponding label follows the expression

$$\log(\mathbb{E}[Y \mid X]) = \langle \theta, X \rangle.$$

While one can derive Poisson regression via maximum likelihood for a integer Poisson random variable, it belongs to the class of proper scoring rules [6] which give unbiased estimators even when the corresponding label is real valued. We will have a deep neural network output parameter θ of Poisson regression and the prediction is e^{θ}. If l is the label for the regression formulation then Poisson log likelihood for the example is

$$l \cdot \theta - e^{\theta}. \tag{1}$$

Offline metric as Negative Poisson log likelihood (NPLL) with respect to normalized label. When training a Poisson regression model we maximize the Poisson log likelihood or minimize the negative Poisson log likelihood (NPLL). So it is natural to evaluate the accuracy of different models by comparing NPLL on a held out dataset. Since we will be normalizing our label we will be evaluating our ideas by comparing NPLL with respect to normalized label.

4 LABEL NORMALIZATION

If we look at the gradient of Poisson log likelihood it is exactly equal to label minus prediction (see eq. (1)). As a result, gradients for advertisers who report in a larger scale will be larger and more model capacity will be devoted to these advertisers. Large gradients affect the model capacity as it will take many examples from smaller labels to compensate for the one large gradient from a large label. While labels for different clicks do represent their relative value for a given advertiser (i.e. they are consistent), across different advertisers they don't necessarily correlate with dollar spent. In particular, advertisers who spend a small amount but report large labels will contribute disproportionately to the loss. As a result, the model will allocate more capacity to predict their labels correctly. Such a system is not

incentive compatible as each advertiser now has an incentive to scale up their labels to a very large scale.

A naive method to resolve the above issue would be to try and pick a loss function whose gradients are scale invariant. Unfortunately, this introduces a different issue. If gradients are scale invariant then the model will take a large number of steps to learn the mean prediction for advertisers with large scale, and not converge (i.e. bounce around) for advertisers who report labels on a small scale.

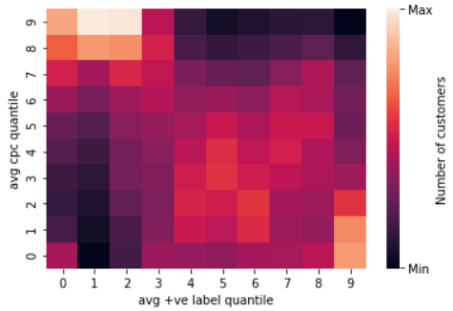

Figure 1: A heatmap of avg label per customer vs avg cpc for that customer.

We solve the above issues by using label normalization, where we multiplicatively normalize the label for each advertiser to 1. To do this, we compute a normalization constant for each advertiser $\eta_A = E[\ell|\ell > 0, A]$. Then the final normalized label is $\ell_n = \ell/\eta_A$. Our final prediction will be normalization constant times the model prediction, i.e $\eta_A \cdot E[\ell_n|X, A]$.

Note that the above design fixes both issue of model capacity allocation as well as learning the mean for each advertiser. Since we do a per advertiser normalization, the average label $\mathbb{E}[\ell|\ell > 0, A]$ is easily computed online. Since the normalized label for all advertisers in the same range, the model converges in a few steps for all advertisers. As a result, it is easy to see why the final prediction of $E[\ell|\ell > 0, A] \cdot E[\ell_n|X, A]$ becomes calibrated for each advertiser within a few gradient steps. Furthermore, we can see that the gradients are now scale invariant and as a result the model capacity is allocated evenly to all advertisers.

We note that there are several alternatives to using multiplicative normalization. We discuss a few of them below:

- We note that another way to normalize the labels is to use an additive normalizer rather than a multiplicative normalizer. However, this approach doesn't yield favorable results for two reasons. Advertisers use the relative values of the labels to indicate the relative value of conversions. Secondly, the wide range which causes poor predictions is still an issue.
- Another way is to try to weight each sample by $\frac{1}{\mathbb{E}[\ell|\ell>0,A]}$. This approach leads to numerical issues as the gradient can be arbitrarily large. This is because even if the label is small (say $\ell \approx 0$), the prediction could be a constant and the resulting gradient would be enormous.
- Another idea is to simply not try to normalize the label.

We show that the above performs poorly in the experimental section.

5 LEARNING PROPERTIES OF THE DISTRIBUTION

In this section, we construct a model to predict the normalized label as accurately as possible. We notice that these distributions have unique properties and we show ways of exploiting them in the subsections below.

5.1 Outlier handling via Winsorized mean and Multi task learning

Even after normalization of the mean, the relative value of the label compared to its mean can take on large values. While this could be due to multiple effects which may be unique to each advertiser, the most salient hypothesis is that these are caused by outliers. There is a rich literature on the ability to handle outliers in machine learning models. However, many techniques such as "median-of-means" is hard to implement in online systems for two reasons.

(1) Simple techniques such as partitioning the input into smaller buckets does make the resulting median quite robust, however the accuracy suffers due to the reduced batch size in each partition.
(2) More sophisticated techniques such as [9, 18] are not very efficient to implement at scale in a distributed manner and make it difficult to estimate the median in an online fashion.

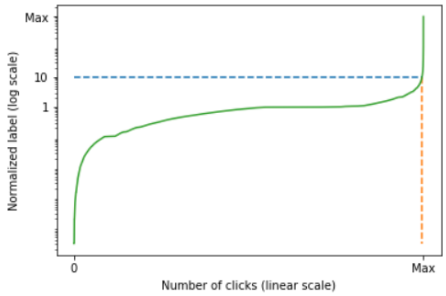

Figure 2: Distribution of normalized label.

Instead we leverage a paradigm from statistics commonly referred to as the Winsorized Mean which is known to perform well in the precence of outliers(see [5, 13]). To compute the mean of a random variable, we first winsorize the sample set – replace the extremes of the sample space with truncated values. Observe that this is quite difficult to do with extremes in the distribution directly, as each advertiser might have a different extreme due to the large range in their values. By normalizing the mean of each advertiser, we can truncate the relative value (the ratio of the label to the mean) for all advertisers simultaneously.

Simply predicting a truncated advertiser can result in a very poor poisson logloss. This is because some advertisers may indeed have a higher normalized label and may not be outliers. However, it is not easy to distinguish these advertisers than those for whom large labels are outliers. Instead, we take a two-pronged approach.

(1) Create a separate objective (by introducing a new head) that tries to minimize the winsorized relative label.
(2) We down-weight the objective for the un-truncated value which is used for prediction.

The success of the above approach can be interpreted in two different ways. First, by having a capped label is that it more evenly allocates the model capacity. Secondly, we can view the winsorized mean objective is used to regularize the objective value and forces the model to pick an equilibrium that can better allocate model capacity for all advertisers.

5.2 Handling zero inflation

One common problem faced in modelling advertiser reported values is that most of the labels are zero. This is a common problem that is faced in many real-world datasets. One technique that is used to handle this issue is to use a model this distribution using a "Zero-Inflated Model". This suggests that observed phenomenon arise as the composition of two separate processes: the first chooses the probability of being zero and the second arise from some natural probabilistic process, in our case a Poisson Model.

Perhaps the most natural way to capture this in a machine learning model to split the label generating process as $\mathbb{E}[\ell] = \Pr[\ell > 0] \cdot \mathbb{E}[\ell \mid \ell > 0]$. Surprisingly the resulting model has *poorer* performance. We find that it is better to have the model predict the value directly and have a separate head predict $\Pr[\ell > 0]$. This is due to two compounding effects. The first is that it is inherently harder to optimize the product of two labels as both models have to be accurate. The second is that the model can allocate its own capacity between these objectives as it sees fit rather than having it artificially decide weight both objectives equally. Lastly, the additional head can serve as a regularizer for the serving objective.

Remark 1. *Note that the splitting of the label (as mentioned above) is also compatible with the other ideas in our paper. For example, we can easily incorporate label normalization by predicting* $\mathbb{E}[\ell] = \Pr[\ell > 0|X] \cdot \mathbb{E}\left[\frac{\ell}{\eta_A} \mid \ell > 0, X\right] \cdot \eta_A$ *where* $\eta_A = \mathbb{E}[\ell \mid A, \ell > 0]$.

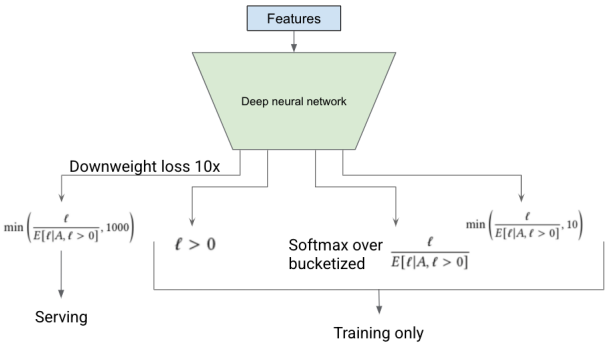

Figure 3: A pictorial representation of DNN along with all the ideas in this paper.

As we show in our experiments, the addition of a new head that predicts this probability results in an improvement in the Poisson log loss (see table 3).

5.3 Teaching label distribution

A novel aspect that we introduce in this work is to improve the regression task with the use of additional objectives which recast the problem as a classification task. In particular, we believe that this provides an additional regularization technique that helps to sort out the outlier and focus the model's capacity into more useful regions of the latent space. While training, most models aim to predict just the mean of the distribution. However, one can make the model learn various aspects of the distribution. We show that learning the quantiles of the distribution can help the overall accuracy of the mean prediction.

To do this, we add an additional objective that asks the model to predict the quantile of the label of the normalized label. The number of quantiles to predict, is a hyperparameter that can be tuned. We find that the addition of these heads improves the model performance especially as the model size increases. In particular, a larger model with these additional heads outperforms a larger model without these additional heads.

One reason why this model improves performance is that the additional head is further able to distinguish between extreme outliers. Secondly, deep models are very good at classification tasks as evidenced by their performance on a number of classification tasks (for e.g. [19]). By breaking the regression problem into smaller classification tasks, we can leverage the improved performance in classification.

6 EXPERIMENTS

We have performed both live experiments and experiments on offline historical data. We report only the latter due to the propriety nature of live experiments.

Data-set. We train the models on data from a commercial search engine's logs. Each example is a click and the label is total value of conversions for that click as reported by the advertisers. Like [7], we assume last click attribution where conversions are attributed to the most recent click. We train on XX-Billion training examples.

Features and Model. Similar to most machine learning models in advertising, we use several categorical features. In particular we embed these features, concatenate the embeddings for each of these features and then pass it through several layers of a fully connected feed forward deep neural network.

Optimizer and Training Model. For each of the objectives suggested, we append a final layer that is attached to the appropriate objective. For the classification objective, we use a simple softmax function. For predicting whether the $\Pr[\ell > 0]$, we use a sigmoid-loss function.

All models use the Ada-Grad optimizer [11] with the same hyper-parameters (including the same learning rate). With the exception of the down-weighting on the final objective, all other objectives are weighted uniformly. Lastly, these models are trained in an online fashion [25]. In online training you start training on the oldest examples and then train on examples in the order of time that they occurred. This allows the models to continuously capture any drifts in the distribution in either the mean or the actual value reported by the advertisers. Online training is quite standard in the industry and is widely used to train a wide variety of machine learning models. Note that in online training, each example is evaluated and the loss is noted. As a result, there is no need for a separate test set to evaluate the models.

Evaluation and metrics. We primarily consider negative Poisson log likelihood (NPLL) as the evaluation metric. This is also the same metric used in the loss function. Typically for a machine learning model trained in batch setting we evaluate either using a hold out set or via cross validation. But in the case of online training we can instead evaluate the model on the example before training on that example. That way we get exactly the performance that you would get at serving time. We will use this form of evaluation in reporting all our metrics. While we train over several months of data and show plots over the complete period we report aggregate numbers in the table over the final 3 months.

6.1 Models Trained

We considered the following models and evaluate NPLL over a period of X months.

- **Baseline Model (BN)** We start a baseline model that contains a simple model that trains on the normalized label. We also have a simple counting model to compute running average of the labels seen for each advertiser for computing the normalization constant, i.e. $\eta_A = \mathbb{E}[\ell \mid A, \ell > 0]$.
- **Simple - (S)** We consider a model that contains no additional heads and directly tries to predict the final label. We re-weight each example by the same number for all events. This ensures that the learning rate is comparable across the baseline model and the simple model (due to difference in global average label).
- **Median of Means - (MM)** We also take 3 copies of the baseline model, with each model training on a third of the data. We output the median prediction of the three towers. Observe that this model is 3 times more expensive as it contains a 3 copies of the baseline model.
- **Plain Model with Weighted Training - (PW)** We also consider a model where each event is weighted by $1/\eta_A$ instead of normalizing the label.
- **Full Model - (F)**: A model with label normalization and all improvements from learning the distribution. These include an additional head to predict the winsorized label, an additional head to predict if the $\Pr[\ell > 0]$ and a softmax head to predict the quantile of the normalized label.
- **Full model without $\ell > 0$ head - (FP)** A model with the same configuration as **F** except the head predicting the probability $\Pr[\ell > 0]$.
- **Full model without Softmax Head - (FS)** A model with the same configuration as **F** except the softmax head which predicts the quantile.
- **Full model without Winsorized Label Head - (FW1)** A model with the same configuration as **F** except the capped head and head predicting the original value is weighted normally.
- **Full model without Winsorized Label Head - (FW2)** A model with the same configuration as **F** except the capped head and head predicting the original value is downweighted by 10x.
- **Zero Inflated FullModel - (ZI)** A model with the same configuration as **F** except the following change. We split the normalized label into a product of two normalized labels: $\mathbb{E}[\ell_\eta] = \Pr[\ell_\eta > 0] \cdot \mathbb{E}[\ell_\eta \mid \ell_\eta > 0]$. We have two heads predicting each component. The first component uses sigmoid loss function and the second one uses a poisson log loss.

6.2 Results For Label Normalization

In this subsection, we compare the models **S**, **BN** and **PW**. We begin by noting that **PW** doesn't even train and has severe numerical issues. The reason is because the gradient for this formulation is $(\ell - e^\theta)/E[\ell|A, \ell > 0])$ and while $\ell/E[\ell|A, \ell > 0]$ is numerically stable we find that $e^\theta/E[\ell|A, \ell > 0]$ cannot be numerically stabilized. We see that this quantity explodes when we consider an advertiser for whom $E[\ell|A, \ell > 0]$ is very very small.

Models	Relative NPLL for un-normalized labels	Relative NPLL for normalized labels
S	0.0%	0.0%
BN	+1.53%	-38.02%

Table 1: Poisson log likelihood for label changes

Increasing value of bucketized $E[\ell\|A, \ell > 0]$	**S** un-normalized label	**BN** un-normalized label	**S** normalized label	**BN** normalized label
Bucket0	2.36	1.04	17.94	0.99
Bucket1	1.29	0.85	1.75	0.99
Bucket2	1.03	1.0	1.03	1.0
Bucket3	1.02	1.0	1.02	1.0
Bucket4	1.01	1.01	1.01	1.0
Bucket5	1.03	1.10	1.11	1.04
Bucket6	1.0	1.04	1.01	1.01

Table 2: Avg Prediction/Avg Label

Now we compare **S** and **BN**. As we can see from table 1, **S** does better on PLL with respect to un-normalized label since it directly trains on un-normalized labels. But when we consider the results with respect to normalized labels it does terribly. To further showcase the issue we look at bias of both the models on data sliced by bucketized values of avg per advertiser label, i.e $E[\ell|A, \ell > 0]$. If we look at table 2 we see that **S** has terrible overprediction for smaller values of $E[\ell|A, \ell > 0]$. This is true even though we have advertiser as a feature in the model.

6.3 Learning Properties of the Distribution.

In this section, we will evaluate the performance of each of the improvements that we added. The plot from 4 shows how the relative accuracy of each model with respect to **BN** changes over time. To compare the total accuracy improvement of all the improvements that we added we compare baseline model **BN** to the fullmodel **F** and we see an overall improvement of 0.87% NPLL which is significant.

The next step in the evaluation is to show that each of the improvements we added is necessary. To do so we run ablation by removing 1 change at a time and compare **F** with the models **FP**, **FS**, **FW1**, **FW2**. We can see in table 3 that the model is indeed worse if we exclude any of the improvements that we add.

We further show why our results are better than what is in the literature. In table 3 we see that **MM** is neutral with respect to **BN** showing that median of means doesn't seem to solve the outlier problem. In addition, we see that splitting the prediction as $Pr(\ell_n > 0)E[\ell|\ell_n > 0]$ in **ZI** is stricly worse than directly predicting $E[\ell_n]$ and adding an additional head for $Pr(\ell_n > 0)$ in **F**.

On top of the above aggregate analysis we also look at metrics on an interesting slice of the dataset. More specifically, we look at examples which belong to advertisers who have more than 2% of positive labels which are winsorized. We see that our models tend to do better on these set of advertisers.

Models	Relative NPLL	Relative NPLL for advertisers with > 2% winsorized +ve labels
BN	0.0%	0.0%
MM	+0.04%	0.12%
ZI	-0.47%	-0.75%
F	**-0.87%**	**-1.23%**
FP	-0.67%	-1.19%
FS	-0.79%	-1.07%
FW1	-0.50%	-0.81%
FW2	-0.50%	-0.33%

Table 3: NPLL improvements of various model variants with respect to normalized label

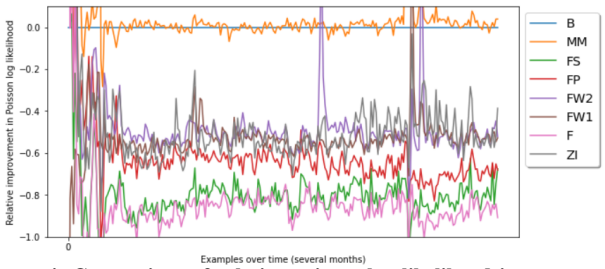

Figure 4: Comparison of relative poisson log likelihood improvement over time.

REFERENCES

[1] [n.d.]. About Target ROAS bidding. https://support.google.com/google-ads/answer/6268637?hl=en

[2] Pravin K. Trivedi Adrian Colin Cameron. 1998. *Regression analysis of count data*. Cambridge University Press.

[3] Deepak Agarwal, Rahul Agrawal, Rajiv Khanna, and Nagaraj Kota. 2010. Estimating Rates of Rare Events with Multiple Hierarchies Through Scalable Log-linear Models *(KDD '10)*. ACM, New York, NY, USA, 213–222. https://doi.org/10.1145/1835804.1835834

[4] Gagan Aggarwal, Ashish Goel, and Rajeev Motwani. 2006. Truthful auctions for pricing search keywords. In *Proceedings 7th ACM Conference on Electronic Commerce (EC-2006), Ann Arbor, Michigan, USA, June 11-15, 2006*, Joan Feigenbaum, John C.-I. Chuang, and David M. Pennock (Eds.). ACM, 1–7. https://doi.org/10.1145/1134707.1134708

[5] N Balakrishnan and N Kannan. 2003. Variance of a Winsorized mean when the sample contains multiple outliers. *Communications in Statistics-Theory and Methods* 32, 1 (2003), 139–149.

[6] G. W. Brier. 1950. VERIFICATION OF FORECASTS EXPRESSED IN TERMS OF PROBABILITY. *Monthly Weather Review* 78 (1950), 1–3.

[7] Olivier Chapelle. 2014. Modeling delayed feedback in display advertising. 1097–1105. https://doi.org/10.1145/2623330.2623634

[8] Olivier Chapelle. 2015. Offline Evaluation of Response Prediction in Online Advertising Auctions. In *WWW (WWW '15 Companion)*. Association for Computing Machinery, New York, NY, USA, 919–922. https://doi.org/10.1145/2740908.2742566

[9] Ilias Diakonikolas, Daniel M Kane, and Ankit Pensia. 2020. Outlier robust mean estimation with subgaussian rates via stability. *arXiv preprint arXiv:2007.15618* (2020).

[10] Eustache Diemert, Julien Meynet, Pierre Galland, and Damien Lefortier. 2017. Attribution Modeling Increases Efficiency of Bidding in Display Advertising. In *ADKDD*. ACM, 2:1–2:6. https://doi.org/10.1145/3124749.3124752

[11] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research* 12, 7 (2011).

[12] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords. *American Economic Review* 97, 1 (March 2007), 242–259. https://doi.org/10.1257/aer.97.1.242

[13] Wayne A Fuller. 1991. Simple estimators for the mean of skewed populations. *Statistica Sinica* (1991), 137–158.

[14] Tilmann Gneiting and Adrian E Raftery. 2007. Strictly Proper Scoring Rules, Prediction, and Estimation. *J. Amer. Statist. Assoc.* 102, 477 (2007), 359–378. https://doi.org/10.1198/016214506000001437

[15] Samuel B Hopkins and Jerry Li. 2019. How Hard is Robust Mean Estimation?. In *Conference on Learning Theory*. PMLR, 1649–1682.

[16] Patrick Hummel and R. Preston McAfee. 2017. Loss Functions for Predicted Click-Through Rates in Auctions for Online Advertising. *Journal of Applied Econometrics* 32 (2017), 1314–1328. http://onlinelibrary.wiley.com/doi/10.1002/jae.2581/full

[17] N Jansakul and JP Hinde. 2002. Score tests for zero-inflated Poisson models. *Computational statistics & data analysis* 40, 1 (2002), 75–96.

[18] Pravesh K Kothari, Jacob Steinhardt, and David Steurer. 2018. Robust moment estimation and improved clustering via sum of squares. In *STOC*. 1035–1046.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012), 1097–1105.

[20] Diane Lambert. 1992. Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing. *Technometrics* 34, 1 (1992), 1–14.

[21] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past performance data. In *SIGKDD*, Qiang Yang, Deepak Agarwal, and Jian Pei (Eds.). ACM, 768–776. https://doi.org/10.1145/2339530.2339651

[22] R. Lewis and J. Wong. 2018. Incrementality Bidding & Attribution. *Microeconomics: Production* (2018).

[23] Gábor Lugosi and Shahar Mendelson. 2019. Mean Estimation and Regression Under Heavy-Tailed Distributions: A Survey. *Found. Comput. Math.* 19, 5 (2019), 1145–1190. https://doi.org/10.1007/s10208-019-09427-x

[24] Andrea Mangani. 2004. Online advertising: Pay-per-view versus pay-per-click. *Journal of Revenue and Pricing Management* 2, 4 (2004), 295–302.

[25] H. Brendan McMahan, Gary Holt, D. Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos, and Jeremy Kubica. 2013. Ad Click Prediction: A View from the Trenches. In *KDD (KDD '13)*. ACM, New York, NY, USA, 1222–1230. https://doi.org/10.1145/2487575.2488200

[26] Aditya Krishna Menon, Krishna Prasad Chitrapura, Sachin Garg, Deepak Agarwal, and Nagaraj Kota. 2011. Response prediction using collaborative filtering with hierarchies and side-information. In *SIGKDD*, Chid Apté, Joydeep Ghosh, and Padhraic Smyth (Eds.). ACM, 141–149. https://doi.org/10.1145/2020408.2020436

[27] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *WWW*, Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy (Eds.). ACM, 521–530. https://doi.org/10.1145/1242572.1242643

[28] Romer Rosales, Haibin Cheng, and Eren Manavoglu. 2012. Post-Click Conversion Modeling and Analysis for Non-Guaranteed Delivery Display Advertising. (2012), 293–302.

[29] Leonard J. Savage. 1971. Elicitation of Personal Probabilities and Expectations. *J. Amer. Statist. Assoc.* 66, 336 (1971), 783–801.

[30] Hal R. Varian. 2007. Position auctions. *International Journal of Industrial Organization* 25, 6 (2007), 1163 – 1178. https://doi.org/10.1016/j.ijindorg.2006.10.002

[31] Flavian Vasile, Damien Lefortier, and Olivier Chapelle. 2017. Cost-sensitive Learning for Utility Optimization in Online Advertising Auctions. In *ADKDD'17*. 8:1–8:6. https://doi.org/10.1145/3124749.3124751

[32] Robert L. Winkler. 1969. Scoring Rules and the Evaluation of Probability Assessors. *J. Amer. Statist. Assoc.* 64, 327 (1969), 1073–1078.

[33] M Xie, B He, and TN Goh. 2001. Zero-inflated Poisson model in statistical process control. *Computational statistics & data analysis* 38, 2 (2001), 191–201.

[34] Zeyuan Allen Zhu, Weizhu Chen, Tom Minka, Chenguang Zhu, and Zheng Chen. 2010. A novel click model and its applications to online advertising. In *WSDM*, Brian D. Davison, Torsten Suel, Nick Craswell, and Bing Liu (Eds.). ACM, 321–330. https://doi.org/10.1145/1718487.1718528