Entire Space Multi-Task Model: An Effective Approach for Estimating Post-Click Conversion Rate

Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, Kun Gai
Alibaba Inc.
{maxiao.mx, liqin.zlq, bingyi.wz, xiaoqiang.zxq, jingshi.gk}@alibaba-inc.com

ABSTRACT

Estimating post-click conversion rate (CVR) accurately is crucial for ranking systems in industrial applications such as recommendation and advertising. Conventional CVR modeling applies popular deep learning methods and achieves state-of-the-art performance. However it encounters several task-specific problems in practice, making CVR modeling challenging. For example, conventional CVR models are trained with samples of clicked impressions while utilized to make inference on the entire space with samples of all impressions. This causes a *sample selection bias* problem. Besides, there exists an extreme *data sparsity* problem, making the model fitting rather difficult. In this paper, we model CVR in a brand-new perspective by making good use of sequential pattern of user actions, i.e., *impression* → *click* → *conversion*. The proposed **E**ntire **S**pace **M**ulti-task **M**odel (ESMM) can eliminate the two problems simultaneously by i) modeling CVR directly over the entire space, ii) employing a feature representation transfer learning strategy. Experiments on dataset gathered from traffic logs of Taobao's recommender system demonstrate that ESMM significantly outperforms competitive methods. We also release a sampling version of this dataset to enable future research. To the best of our knowledge, this is the first public dataset which contains samples with sequential dependence of click and conversion labels for CVR modeling.

KEYWORDS

post-click conversion rate, multi-task learning, sample selection bias, data sparsity, entire-space modeling

1 INTRODUCTION

Conversion rate (CVR) prediction is an essential task for ranking system in industrial applications, such as online advertising and recommendation etc. For example, predicted CVR is used in OCPC (optimized cost-per-click) advertising to adjust bid price per click to achieve a win-win of both platform and advertisers [4]. It is also an important factor in recommender systems to balance users' click preference and purchase preference.

In this paper, we focus on the task of post-click CVR estimation. To simplify the discussion, we take the CVR modeling in recommender system in e-commerce site as an example. Given recommended items, users might click interested ones and further buy some of them. In other words, user actions follow a sequential pattern of *impression* → *click* → *conversion*. In this way, CVR modeling refers to the task of estimating the post-click conversion rate, i.e., $pCVR = p(conversion|click, impression)$.

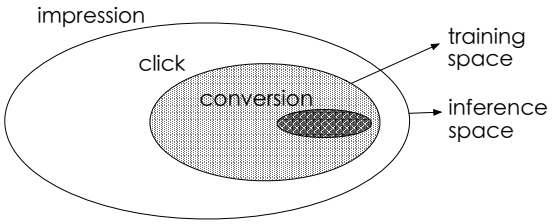

Figure 1: Illustration of sample selection bias problem in conventional CVR modeling. Training space is composed of samples with clicked impressions. It is only part of the inference space which is composed of all impressions.

In general, conventional CVR modeling methods employ similar techniques developed in click-through rate (CTR) prediction task, for example, recently popular deep networks [2, 3]. However, there exist several task-specific problems, making CVR modeling challenging. Among them, we report two critical ones encountered in our real practice: i) *sample selection bias (**SSB**)* problem [12]. As illustrated in Fig.1, conventional CVR models are trained on dataset composed of clicked impressions, while are utilized to make inference on the entire space with samples of all impressions. *SSB* problem will hurt the generalization performance of trained models. ii) *data sparsity (**DS**)* problem. In practice, data gathered for training CVR model is generally much less than CTR task. Sparsity of training data makes CVR model fitting rather difficult.

There are several studies trying to tackle these challenges. In [5], hierarchical estimators on different features are built and combined with a logistic regression model to solve *DS* problem. However, it relies on a priori knowledge to construct hierarchical structures, which is difficult to be applied in recommender systems with tens of millions of users and items. Oversampling method [11] copies rare class examples which helps lighten sparsity of data but is sensitive to sampling rates. All Missing As Negative (AMAN) applies random sampling strategy to select un-clicked impressions as negative examples [6]. It can eliminate the *SSB* problem to some degree by introducing unobserved examples, but results in a consistently underestimated prediction. Unbiased method [10] addresses *SSB* problem in CTR modeling by fitting the truly underlying distribution from observations via rejection sampling. However, it might encounter numerical instability when weighting samples by division of rejection probability. In all, neither *SSB* nor *DS* problem has

SIGIR'18, July 8-12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07.
https://doi.org/10.1145/3209978.3210104

been well addressed in the scenario of CVR modeling, and none of above methods exploits the information of sequential actions.

In this paper, by making good use of sequential pattern of user actions, we propose a novel approach named Entire Space Multi-task Model (ESMM), which is able to eliminate the *SSB* and *DS* problems simultaneously. In ESMM, two auxiliary tasks of predicting the post-view click-through rate (CTR) and post-view click-through&conversion rate (CTCVR) are introduced. Instead of training CVR model directly with samples of clicked impressions, ESMM treats *pCVR* as an intermediate variable which multiplied by *pCTR* equals to *pCTCVR*. Both *pCTCVR* and *pCTR* are estimated over the entire space with samples of all impressions, thus the derived *pCVR* is also applicable over the entire space. It indicates that *SSB* problem is eliminated. Besides, parameters of feature representation of CVR network is shared with CTR network. The latter one is trained with much richer samples. This kind of parameter transfer learning [7] helps to alleviate the *DS* trouble remarkablely.

For this work, we collect traffic logs from Taobao's recommender system. The full dataset consists of 8.9 billions samples with sequential labels of click and conversion. Careful experiments are conducted. ESMM consistently outperforms competitive models, which demonstrate the effectiveness of the proposed approach. We also release our dataset[1] for future research in this area.

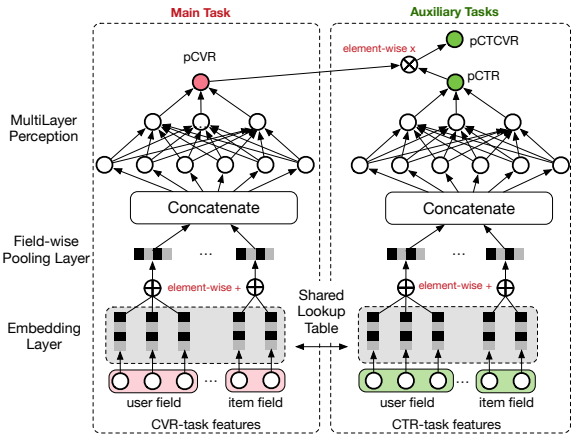

Figure 2: Architecture overview of ESMM for CVR modeling. In ESMM, two auxiliary tasks of CTR and CTCVR are introduced which: i) help to model CVR over entire input space, ii) provide feature representation transfer learning. ESMM mainly consists of two sub-networks: CVR network illustrated in the left part of this figure and CTR network in the right part. Embedding parameters of CTR and CVR network are shared. CTCVR takes the product of outputs from CTR and CVR network as the output.

2 THE PROPOSED APPROACH

2.1 Notation

We assume the observed dataset to be $\mathcal{S} = \{(\boldsymbol{x}_i, y_i \rightarrow z_i)\}|_{i=1}^{N}$, with sample $(\boldsymbol{x}, y \rightarrow z)$ drawn from a distribution D with domain $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, where $\mathcal{X}$ is feature space, $\mathcal{Y}$ and $\mathcal{Z}$ are label spaces, and N is the total number of impressions. $\boldsymbol{x}$ represents feature vector of observed impression, which is usually a high dimensional sparse vector with multi-fields [8], such as user field, item field etc. y and z are binary labels with $y = 1$ or $z = 1$ indicating whether click or conversion event occurs respectively. $y \rightarrow z$ reveals the sequential dependence of click and conversion labels that there is always a preceding click when conversion event occurs.

Post-click CVR modeling is to estimate the probability of $pCVR = p(z = 1|y = 1, \boldsymbol{x})$. Two associated probabilities are: post-view click-through rate (CTR) with $pCTR = p(z = 1|\boldsymbol{x})$ and post-view click&conversion rate (CTCVR) with $pCTCVR = p(y = 1, z = 1|\boldsymbol{x})$. Given impression $\boldsymbol{x}$, these probabilities follow Eq.(1):

$$\underbrace{p(y = 1, z = 1|\boldsymbol{x})}_{pCTCVR} = \underbrace{p(y = 1|\boldsymbol{x})}_{pCTR} \times \underbrace{p(z = 1|y = 1, \boldsymbol{x})}_{pCVR}. \quad (1)$$

2.2 CVR Modeling and Challenges

Recently deep learning based methods have been proposed for CVR modeling, achieving state-of-the-art performance. Most of them follow a similar Embedding&MLP network architecture, as introduced in [3]. The left part of Fig.2 illustrates this kind of architecture, which we refer to as **BASE** model, for the sake of simplicity.

In brief, conventional CVR modeling methods directly estimate the post-click conversion rate $p(z = 1|y = 1, \boldsymbol{x})$. They train models with samples of clicked impressions, i.e., $\mathcal{S}_c = \{(\boldsymbol{x}_j, z_j)|y_j = 1\}|_{j=1}^{M}$. M is the number of clicks over all impressions. Obviously, $\mathcal{S}_c$ is a subset of $\mathcal{S}$. Note that in $\mathcal{S}_c$, (clicked) impressions without conversion are treated as negative samples and impressions with conversion (also clicked) as positive samples. In practice, CVR modeling encounters several task-specific problems, making it challenging.

Sample selection bias (SSB) [12]. In fact, conventional CVR modeling makes an approximation of $p(z = 1|y = 1, \boldsymbol{x}) \approx q(z = 1|\boldsymbol{x}_c)$ by introducing an auxiliary feature space $\mathcal{X}_c$. $\mathcal{X}_c$ represents a *limited*[2] space associated with $\mathcal{S}_c$. $\forall \boldsymbol{x}_c \in \mathcal{X}_c$ there exists a pair $(\boldsymbol{x} = \boldsymbol{x}_c, y_{\boldsymbol{x}} = 1)$ where $\boldsymbol{x} \in \mathcal{X}$ and $y_{\boldsymbol{x}}$ is the click label of $\boldsymbol{x}$. In this way, $q(z = 1|\boldsymbol{x}_c)$ is trained over space $\mathcal{X}_c$ with clicked samples of $\mathcal{S}_c$. At inference stage, the prediction of $p(z = 1|y = 1, \boldsymbol{x})$ over entire space $\mathcal{X}$ is calculated as $q(z = 1|\boldsymbol{x})$ under the assumption that for any pair of $(\boldsymbol{x}, y_{\boldsymbol{x}} = 1)$ where $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{x}$ belongs to $\mathcal{X}_c$. This assumption would be violated with a large probability as $\mathcal{X}_c$ is just a small part of entire space $\mathcal{X}$. It is affected heavily by the randomness of rarely occurred click event, whose probability varies over regions in space $\mathcal{X}$. Moreover, without enough observations in practice, space $\mathcal{X}_c$ may be quite different from $\mathcal{X}$. This would bring the drift of distribution of training samples from truly underling distribution and hurt the generalization performance for CVR modeling.

Data sparsity (DS). Conventional methods train CVR model with clicked samples of $\mathcal{S}_c$. The rare occurrence of click event causes training data for CVR modeling to be extremely sparse. Intuitively, it is generally 1-3 orders of magnitude less than the associated CTR task, which is trained on dataset of $\mathcal{S}$ with all impressions. Table 1 shows the statistics of our experimental datasets, where number of samples for CVR task is just 4% of that for CTR task.

[1] https://tianchi.aliyun.com/datalab/dataSet.html?dataId=408

[2] space $\mathcal{X}_c$ equals to $\mathcal{X}$ under the condition that $\forall X \in \mathcal{X}$, $p(y = 1|\boldsymbol{x}) > 0$ and the number of observed impressions is large enough. Otherwise, space $\mathcal{X}_c$ is part of $\mathcal{X}$.

It is worth mentioning that there exists other challenges for CVR modeling, e.g. *delayed feedback* [1]. This work does not focus on it. One reason is that the degree of conversion delay in our system is slightly acceptable. The other is that our approach can be combined with previous work [1] to handle it.

2.3 Entire Space Multi-Task Model

The proposed **ESMM** is illustrated in Fig.2, which makes good use of the sequential pattern of user actions. Borrowing the idea from multi-task learning [9], ESMM introduces two auxiliary tasks of CTR and CTCVR and eliminates the aforementioned problems for CVR modeling simultaneously.

On the whole, ESMM simultaneously outputs *pCTR*, *pCVR* as well as *pCTCVR* w.r.t. a given impression. It mainly consists of two sub-networks: CVR network illustrated in the left part of Fig.2 and CTR network in the right part. Both CVR and CTR networks adopt the same structure as **BASE** model. CTCVR takes the product of outputs from CVR and CTR network as the output. There are some highlights in ESMM, which have notable effects on CVR modeling and distinguish ESMM from conventional methods.

Modeling over entire space. Eq.(1) gives us hints, which can be transformed into Eq.(2).

$$p(z = 1|y = 1, \boldsymbol{x}) = \frac{p(y = 1, z = 1|\boldsymbol{x})}{p(y = 1|\boldsymbol{x})} \tag{2}$$

Here $p(y = 1, z = 1|\boldsymbol{x})$ and $p(y = 1|\boldsymbol{x})$ are modeled on dataset of $\mathcal{S}$ with all impressions. Eq.(2) tells us that with estimation of *pCTCVR* and *pCTR*, *pCVR* can be derived over the entire input space $\mathcal{X}$, which addresses the *sample selection bias* problem directly. This seems easy by estimating *pCTR* and *pCTCVR* with individually trained models separately and obtaining *pCVR* by Eq.(2), which we refer to as **DIVISION** for simplicity. However, *pCTR* is a small number practically, divided by which would arise numerical instability. ESMM avoids this with the multiplication form. In ESMM, *pCVR* is just an intermediate variable which is constrained by the equation of Eq.(1). *pCTR* and *pCTCVR* are the main factors ESMM actually estimated over entire space. The multiplication form enables the three associated and co-trained estimators to exploit the sequential patten of data and communicate information with each other during training. Besides, it ensures the value of estimated *pCVR* to be in range of [0,1], which in DIVISION method might exceed 1.

The loss function of ESMM is defined as Eq.(3). It consists of two loss terms from CTR and CTCVR tasks which are calculated over samples of all impressions, without using the loss of CVR task.

$$\begin{aligned} L(\theta_{cvr}, \theta_{ctr}) = &\sum_{i=1}^{N} l\left(y_i, f(\boldsymbol{x}_i; \theta_{ctr})\right) \\ &+ \sum_{i=1}^{N} l\left(y_i \& z_i, f(\boldsymbol{x}_i; \theta_{ctr}) \times f(\boldsymbol{x}_i; \theta_{cvr})\right), \end{aligned} \tag{3}$$

where θ_{ctr} and θ_{cvr} are the parameters of CTR and CVR networks and $l(\cdot)$ is cross-entropy loss function. Mathematically, Eq.(3) decomposes $y \rightarrow z$ into two parts[3]: y and $y\&z$, which in fact makes use of the sequential dependence of click and conversion labels.

Feature representation transfer. As introduced in section 2.2, embedding layer maps large scale sparse inputs into low dimensional representation vectors. It contributes most of the parameters of deep network and learning of which needs huge volume of training samples. In ESMM, embedding dictionary of CVR network is shared with that of CTR network. It follows a feature representation transfer learning paradigm. Training samples with all impressions for CTR task is relatively much richer than CVR task. This parameter sharing mechanism enables CVR network in ESMM to learn from un-clicked impressions and provides great help for alleviating the *data sparsity* trouble.

Note that the sub-network in ESMM can be substituted with some recently developed models [2, 3], which might get better performance. Due to limited space, we omit it and focus on tackling challenges encountered in real practice for CVR modeling.

3 EXPERIMENTS

3.1 Experimental Setup

Datasets. During our survey, no public datasets with sequential labels of click and conversion are found in CVR modeling area. To evaluate the proposed approach, we collect traffic logs from Taobao's recommender system and release a 1% random sampling version of the whole dataset, whose size still reaches 38GB (without compression). In the rest of the paper, we refer to the released dataset as ***Public Dataset*** and the whole one as ***Product Dataset***. Table 1 summarizes the statistics of the two datasets. Detailed descriptions can be found in the website of Public Dataset[1].

Competitors. We conduct experiments with several competitive methods on CVR modeling. (1) **BASE** is the baseline model introduced in section 2.2. (2) **AMAN** [6] applies negative sampling strategy and best results are reported with sampling rate searched in {10%, 20%, 50%, 100%}. (3) **OVERSAMPLING** [11] copies positive examples to reduce difficulty of training with sparse data, with sampling rate searched in $\{2, 3, 5, 10\}$. (4) **UNBIAS** follows [10] to fit the truly underlying distribution from observations via rejection sampling. *pCTR* is taken as the rejection probability. (5) **DIVISION** estimates *pCTR* and *pCTCVR* with individually trained CTR and CTCVR networks and calculates *pCVR* by Eq.(2). (6) **ESMM-NS** is a lite version of ESMM without sharing of embedding parameters.

The first four methods are different variations to model CVR directly based on state-of-the-art deep network. DIVISION, ESMM-NS and ESMM share the same idea to model CVR over entire space which involve three networks of CVR, CTR and CTCVR. ESMM-NS and ESMM co-train the three networks and take the output from CVR network for model comparison. To be fair, all competitors including ESMM share the same network structure and hyper parameters with BASE model, which i) uses ReLU activation function, ii) sets the dimension of embedding vector to be 18, iii) sets dimensions of each layers in MLP network to be $360 \times 200 \times 80 \times 2$, iv) uses adam solver with parameter $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$.

Metric. The comparisons are made on two different tasks: (1) conventional CVR prediction task which estimates *pCVR* on dataset with clicked impressions, (2) CTCVR prediction task which estimates *pCTCVR* on dataset with all impressions. Task (2) aims to

[3] Corresponding to labels of CTR and CTCVR tasks, which construct training datasets as follows: i) samples are composed of all impressions, ii) for CTR task, clicked impressions are labeled $y = 1$, otherwise $y = 0$, iii) for CTCVR task, impressions with click and conversion events occurred simultaneously are labeled $y\&z = 1$, otherwise $y\&z = 0$

Table 1: Statistics of experimental datasets.

dataset	#user	#item	#impression	#click	#conversion
Public Dataset	0.4M	4.3M	84M	3.4M	18k
Product Dataset	48M	23.5M	8950M	324M	1774k

Table 2: Comparison of different models on Public Dataset.

Model	AUC(mean ± std) on CVR task	AUC(mean ± std) on CTCVR task
BASE	66.00 ± 0.37	62.07 ± 0.45
AMAN	65.21 ± 0.59	63.53 ± 0.57
OVERSAMPLING	67.18 ± 0.32	63.05 ± 0.48
UNBIAS	66.65 ± 0.28	63.56 ± 0.70
DIVISION	67.56 ± 0.48	63.62 ± 0.09
ESMM-NS	**68.25 ± 0.44**	**64.44 ± 0.62**
ESMM	**68.56 ± 0.37**	**65.32 ± 0.49**

compare different CVR modeling methods over entire input space, which reflects the model performance corresponding to *SSB* problem. In CTCVR task, all models calculate *pCTCVR* by $pCTR \times pCVR$, where: i) *pCVR* is estimated by each model respectively, ii) *pCTR* is estimated with a same independently trained CTR network (same structure and hyper parameters as BASE model). Both of the two tasks split the first 1/2 data in the time sequence to be training set while the rest to be test set. Area under the ROC curve (AUC) is adopted as performance metrics. All experiments are repeated 10 times and averaged results are reported.

3.2 Results on Public Dataset

Table 2 shows results of different models on public dataset. (1) Among all the three variations of BASE model, only AMAN performs a little worse on CVR task, which may be due to the sensitive of random sampling. OVERSAMPLING and UNBIAS show improvement over BASE model on both CVR and CTCVR tasks. (2) Both DIVISION and ESMM-NS estimate *pCVR* over entire space and achieve remarkable promotions over BASE model. Due to the avoidance of numerical instability, ESMM-NS performs better than DIVISION. (3) ESMM further improves ESMM-NS. By exploiting the sequential patten of user actions and learning from un-clicked data with transfer mechanism, ESMM provides an elegant solution for CVR modeling to eliminate *SSB* and *DS* problems simultaneously and beats all the competitors. Compared with BASE model, ESMM achieves absolute AUC gain of 2.56% on CVR task, which indicates its good generalization performance even for biased samples. On CTCVR task with full samples, it brings 3.25% AUC gain. These results validate the effectiveness of our modeling method.

3.3 Results on Product Dataset

We further evaluate ESMM on our product dataset with 8.9 billions of samples, two orders of magnitude larger than public one. To verify the impact of the volume of the training dataset, we conduct careful comparisons on this large scale datasets w.r.t. different sampling rates, as illustrated in Fig.3. First, all methods show improvement with the growth of volume of training samples. This indicates the influence of data sparsity. In all cases except AMAN on 1% sampling CVR task, BASE model is defeated. Second, ESMM-NS and ESMM outperform all competitors consistently w.r.t. different

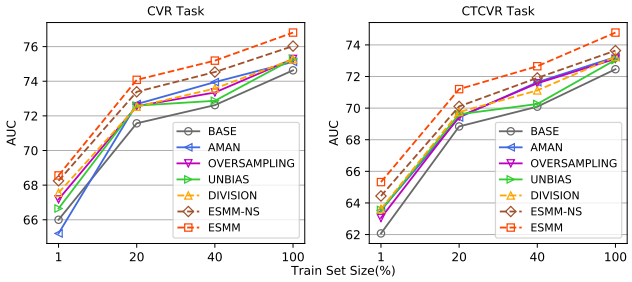

Figure 3: Comparison of different models w.r.t. different sampling rates on Product Dataset.

sampling rates. In particular, ESMM maintains a large margin of AUC promotion over all competitors on both CVR and CTCVR tasks. BASE model is the latest version which serves the main traffic in our real system. Trained with the whole dataset, ESMM achieves absolute AUC gain of 2.18% on CVR task and 2.32% on CTCVR task over BASE model. This is a significant improvement for industrial applications where 0.1% AUC gain is remarkable.

4 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel approach ESMM for CVR modeling task. ESMM makes good use of sequential patten of user actions. With the help of two auxiliary tasks of CTR and CTCVR, ESMM elegantly tackles challenges of *sample selection bias* and *data sparsity* for CVR modeling encountered in real practice. Experiments on real dataset demonstrate the superior performance of the proposed ESMM. This method can be easily generalized to user action prediction in scenario with sequential dependence. In the future, we intend to design global optimization models in applications with multi-stage actions like $request \rightarrow impression \rightarrow click \rightarrow conversion$.

REFERENCES

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

[2] Heng-Tze Cheng and Levent Koc. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.

[3] Zhou G., Song C., et al. 2017. Deep Interest Network for Click-Through Rate Prediction. *arXiv preprint arXiv:1706.06978* (2017).

[4] Zhu H., Jin J., et al. 2017. Optimized cost per click in taobao display advertising. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2191–2200.

[5] Lee K., Orten B., et al. 2012. 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*. ACM.

[6] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 502–511.

[7] Sinno Jialin Pan and Q. Yang. 2010. A Survey on Transfer Learning. In *IEEE Transactions on Knowledge and Data Engineering*. 1345–1359.

[8] Steffen Rendle. 2010. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 995–1000.

[9] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

[10] Zhang W., Zhou T., et al. 2016. Bid-aware gradient descent for unbiased learning with censored data in display advertising. In *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining*. ACM.

[11] Gary M Weiss. 2004. Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter* 6, 1 (2004), 7–19.

[12] Bianca Zadrozny. 2004. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the 21th international conference on Machine learning*. ACM.