Representative Negative Instance Generation for Online Ad Targeting

Yuhan Quan ∗, Jingtao Ding ∗, Depeng Jin ∗, Jianbo Yang †, Xing Zhou †, Yong Li ∗
∗ Beijing National Research Center for Information Science and Technology (BNRist),
Department of Electronic Engineering, Tsinghua University, Beijing, China, 100084
† Tencent
{quanyh19,dingjt15}@mails.tsinghua.edu.cn,jindp@tsinghua.edu.cn
{jianboyang,leostarzhou}@tencent.com,liyong07@tsinghua.edu.cn

ABSTRACT

Online ad targeting can be formulated as a problem of learning the relevance ranking among possible audiences for a given ad. It has to deal with the massive number of negative, *i.e.*, non-interacted, instances in impression data due to the nature of this service, and thus suffers from data imbalance problem. In this work, we tackle this problem by improving the quality of negative instances used in training the targeting model. We propose to enhance the generalization capability by introducing unobserved data as possible negative instances, and extract more reliable negative instances from the observed negatives in impression data. However, this idea is non-trivial to implement because of the limited learning signal and existing noise signal. To this end, we design a novel RNIG method (short for *Representative Negative Instance Generator*) to leverage feature matching technique. It aims to generate reliable negative instances that are similar to the observed negatives and further improves the representativeness of generated negatives by matching the most important feature. Extensive experiments on the real-world ad targeting dataset show that our RNIG model has achieved a relative improvement of more than 5%.

CCS CONCEPTS

• **Information systems → Computational advertising**; • **Computing methodologies → Neural networks**.

KEYWORDS

Negative sampling; Ad targeting; Adversarial learning; Feature matching

ACM Reference Format:
Yuhan Quan ∗, Jingtao Ding ∗, Depeng Jin ∗, Jianbo Yang †, Xing Zhou †, Yong Li ∗. 2020. Representative Negative Instance Generation for Online Ad Targeting. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340531.3412114

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412114

1 INTRODUCTION

With the development of machine learning algorithms and deep neural networks, many advertising companies have provided a new tool of *ad auto-targeting* for advertisers [3, 7], which can automatically match possible audiences for ads according to the relevance learned from historical data. It can be formulated as a problem of learning the relevance ranking among possible audiences for each ad. In the actual online system, the candidate ads are first filtered for each user visit based on the learned relevance score between ads and users, then they will be reranked and displayed to the specific user.

For an ad system, in the impression data of each ad, the interacted users are labelled as positive and other non-interacted users are lablled as negative. However, in practice, positive instances are often a minority, and the vast majority of instances are negative. Therefore, ad targeting model suffers from a serious data imbalance problem (such as 1:100). Adopting random sampling or down sampling on the negative instances can alleviate above data imbalance issue to some extent, yet cause the non-informativeness of leveraged negative instances. For recommender system, advanced negative sampling methods have been proposed to enhance the quality of negative instances during training process, like dynamic negative sampling that oversamples the hard negative instances [12]. Recently, generative adversarial net (GAN) has also been adopted for generating adversarial instances to train better recommendation models, like IRGAN [11] and KBGAN [2]. However, above models focus on item recommendation scenario where negative instances are generated from unobserved data, while in ad targeting there already exist huge amounts of observed negative instances in impression data. Thus the more important problem is to generate representative negative instances based on impression data, which has not been tackled in previous works.

Nevertheless, it is non-trivial to improve the quality of negative instances in ad targeting because of the following two challenges:

- **Generating high-quality negative instances is difficult with the limited information in impression data.** Although the number of negative instances in impression data is large, every ad shown to the user means the ad is relevant to the user to some extent, which leads to lots of redundant information and few high-quality negative instances.
- **Learning only from impression data can be inaccurate due to the selection bias.** Since in a real advertising platform, the impression data are generated by the current system selecting the most suitable ads for users. The selection

process is not random or uniform, which leads to the the selection bias.

In this work, we design a novel generator model named **R**epresentative **N**egative **I**nstance **G**enerator (RNIG) that learns to generate high-quality negative instances and avoid the selection bias. We summarize the contributions of the paper as follows.

1 We propose a RNIG model that can generate representative negative instances from both observed impression data and unobserved data to avoid the selection bias, collaborating with the ad targeting model to achieve better performance.
2 We design a feature matching scheme to learn ad-user negative relevance from the observed negative instances in impression data, and further propose to generate high-quality negative instances by matching the important features.
3 We conduct extensive experiments on a real-world ad targeting dataset to demonstrate the effectiveness of RNIG and verify the utility of feature matching in generating representative negative instances.

2 PROPOSED NEGATIVE INSTANCE GENERATOR

We start by introducing the basic notations. For an ad-user impression matrix $\mathbf{I} \in \mathbb{R}^{M\times N}$, M and N denote the number of ads and users, respectively. As for the ad-user interactions we aim to predict, there is another interaction matrix $\mathbf{R} \in \mathbb{R}^{M\times N}$. Here for brevity we only maintain those displayed but not interacted ad-user pairs as non-zero entries in $\mathbf{I}$, as other interacted ones already exist in $\mathbf{R}$. For a specific ad a, $\mathcal{R}_a$ denotes the set of users that have interacted with a, while $\mathcal{I}_a$ refers to those non-interacted users within a's display history. We use symbols σ and $\odot$ to denote the sigmoid function and element-wise production, respectively.

Our proposed negative instance generator can collaborate with a discriminative targeting model showing in Figure 1. The generator (G) calculates a probability distribution over a set of candidate negative instances, then samples one of them as the output. With the generated negative instance and interacted instance, the discriminator (D) maximizes the margin between their scores by minimizing the BPR loss [9]. After receiving the reward from D, G is encouraged to generate representative negative instances. During training process, D can benefit more from the better quality negative instances and thus perform better on predicting ad-user relevance.

2.1 Generator Network

Our proposed generator can produce the probability distribution for sampling negative instances, *i.e.*, $\hat{\Psi}_G(u|a)$. It is the softmax probability of the score of ad-user pair (a, u), denoted as $\hat{g}_{au}(\Theta)$, where $u \in \mathcal{N}_a$. $\mathcal{N}_a$ denotes a's candidate set for the generated negative instances, which is constructed by both users from a's displayed userset ($u \in \mathcal{I}_a$) and other non-displayed users ($u' \notin \mathcal{I}_a$). Next we elaborate the design of $\hat{g}(\Theta)$ layer by layer.

The first is the Embedding layer. It is a fully connected layer that projects each input feature to a dense vector representation. Then the embedding sets are fed into the encoder to obtain the ad(user) vector $\mathbf{p}_a$ ($\mathbf{q}_u$). Ad and user encoders are two independent multilayer perceptrons (MLPs) where the bottom layer is the widest and each successive layer has a smaller number of neurons [3]. Above

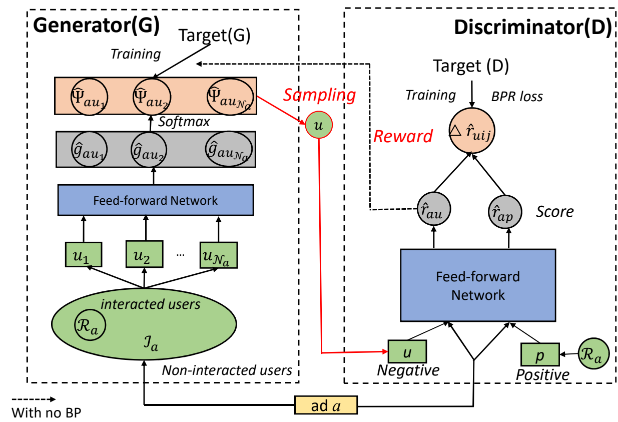

Figure 1: Our proposed *G-D* framework for improving online ad targeting.

the encoding layers is another MLP with H layers, which is capable of learning higher-order interactions between ad and user vectors. Here we adopt the constant size for each layer. Finally, the output vector of the last hidden layer is transformed to the score of representativeness as negative instance $\hat{g}_{au} = \mathbf{h}^T f_{1:H}([\mathbf{p}_a, \mathbf{q}_u])$,where vector $\mathbf{h}$ denotes the neuron weights for scoring layer.

2.2 Generating Representative Negatives via Feature Matching

We introduce the specific design of our proposed RNIG model, which is learned by matching the feature distribution between generated negative instances and those observed negatives in impression data. The generated negative instance (a, u) is paired with a positive instance (a, p) and then sent to the discriminative targeting model D for learning their pairwise ranking relation.

2.2.1 Maximizing the Overlap. As both ad-user interaction and non-interaction are explicitly recorded in the impression data, it is reasonable to generate reliable negative instances based on the displayed but not interacted instances. Therefore, our proposed RNIG learns a probability distribution to match negative signal in the impression data. Given a's displayed userset $\mathcal{I}_a$, we consider the following overlap-based objective:

$$L_G^o = \sum_{(a,p)\in\mathcal{R}} \mathbb{E}_{u\sim\hat{\Psi}_G(u|a)}\left[O_{\mathcal{I}_a}(a,u)\right], \tag{1}$$

where the binary indicator function $O_{\mathcal{I}_a} = 1$ if $u \in \mathcal{I}_a$.

However, if we solely rely on L_G^o to generate negative instances, for the set of generated negative instances, denoted as $\mathcal{G}_a$, and the set of observed negative instances, *i.e.*, $\mathcal{I}_a$, the sampler trained with this objective may tend to choose exactly the same instances from the impression data, which harms the performance as not all of the impression instances are beneficial.

2.2.2 Matching the Feature Distribution. Therefore, we further measure the similarity in the latent feature space and propose a feature matching technique to generate representative negative instance. As the generated negative instance (a, u) is sent to D for learning ad-user relevance, the representation of (a, u) in the latent feature space should be designed in terms of D. Regardless of the specific model structure in D, a final prediction layer, *i.e.*,

$\hat{r}(a,u) = \mathbf{h}_D^T \mathbf{f}_{au}$, is required to calculate the predicted relevance score between ad a and user u. Thus it is natural to choose $\mathbf{f}_{au}$, which is the output of last hidden layer, as the feature vector in latent space. Consider a subset of generated negative instances $\mathcal{G}_s$, for each $(a,u) \in \mathcal{G}_s$, we sample a (a,v) from $\mathcal{I}_a$ and group all these sampled instances into a subset denoted as $\mathcal{I}_s$. Then, we use the maximum mean discrepancy (MMD) between the empirical distribution of feature vectors in these two subsets to measure the distance between $\mathcal{G}_s$ and $\mathcal{I}_s$, denoted as L_G^m. By maximizing it, we are able to encourage G to generate $\{(a,u)\}$ that are similar to those observed negative signal from impression data in the latent feature embedding perspective.

From the above considerations, we design the negative instance generator with a weighted sum of overlap-based objective and MMD-based objective, which can be formulated as maximizing $L_G = (1-\alpha)L_G^o + \alpha L_G^m$.

2.2.3 Improving the Feature Matching. Above feature matching scheme can better leverage the limited learning signal and thus improve the generation quality. However, since ads are displayed to users via a certain relevance-matching mechanism, some of non-interacted instances in the impression data could be the noisy data that does not necessarily indicate the negative ad-user relevance. To overcome this shortcoming, we propose to improve the feature matching. Specifically, for each feature i, we calculate the Jensen-Shannon divergence (JSD) j_i between its two distributions among positive instances and observed negative instances. Then, we select K features with the top K largest JSD values as the important feature set $\mathcal{F}$. Then, we add a mask operation on the input user feature vector $\mathbf{x}$ of the G, such that all the other features $\{j \notin \mathcal{F}\}$ have no impact on the generation of negative instances. Therefore, we are able to guide the generator towards generating representative negative instances that "look" like observed negatives in the most important and essential aspects.

2.3 Collaborating with Targeting Model

For the discriminative targeting model D, we adopt the Bayesian Personalized Ranking (BPR) method for optimizing the model, which assumes that a positive instance should be predicted with a much higher score over the negative one [9]. Minimizing BPR loss is equivalent to maximizing the margin between $\hat{r}_{ap}$ and $\hat{r}_{au}$, which encourages D to learn the pairwise ranking relation of ad relevance between p and u.

As for the structure of targeting model D, we maintain the same structure as the online serving model in Tencent Ad system, which is similar to G without the hidden layers. The final prediction score is calculated by inner product between encoded ad and user vectors, which can utilize efficient approximate kNN search so as to be applied in a large-scale industrial ad targeting system.

However, unlike the optimization of D that can be achieved by the stochastic gradient descent (SGD), training G has following two problems. First, it involves a discrete sampling step, which makes simple differentiation infeasible. Second, the indicator function $\mathbf{O}_{\mathcal{I}_a}$ in overlap-based objective L_G^o is non-differentiable. Therefore, for G with model parameters Θ, we use the policy gradient based RL [10] to derive its gradient as follows,

$$\nabla_\Theta L_G \simeq \sum_{(a,p)\in\mathcal{R}} \frac{1}{T} \sum_{u_t \sim \hat{\Psi}_G(u|a), t\le T} \left[d_{au_t} \nabla_\Phi \log \hat{\Psi}_G(u_t|a) \right], \quad (2)$$

where $d_{au} = (1-\beta)O_{\mathcal{I}_a}(a,u) + \beta M(a,u)$.

The learning process is carried out in mini-batch mode, where D and G alternatively update their parameters. First, for each interaction (a,p), G generates a negative instance u for D to do pairwise leaning. Then, after receiving the reward, G optimizes its parameters so as to generate better quality negative instance. Note that in order to match the feature distribution of generated instance to the observed negative instance in impression data, we sample a subset $\mathcal{I}_s$ to compute the MMD-based reward.

3 EXPERIMENTS

3.1 Experimental Settings

3.1.1 Datasets. We collect a real-world dataset from Tencent Ads[1], which is one of the largest Chinese online display advertising platform. All ads of the dataset belong to CPC (Cost per click) ads according to the different billing mode.

The features of the dataset are composed of 13 user features and 7 ad features. Besides, it additionally includes some statistical features for users, mainly reflecting the feedback behavior (like click) of individual users in history. All value is hashed. The train set is was collected from 2019/06/03 to 2019/06/05, it contains 2,453,966 impressions, 11,270 ads and 1,041,991 users. The test set is was collected in 2019/06/06, it contains 2,205,432 impressions, 5,080 ads and 1,032,202 users.

3.1.2 Performance Methodology. We use GAUC and RelaImpr [14] as the metrics. GAUC is a weighted average of the AUCs for different ads based on the number of samples an ad contains in the test set. For AUC, when all parameters are randomly initialized, the value should be 0.5, so we subtract the value of AUC by 0.5 and the relative improvement is the result of RelaImpr.

3.1.3 Baselines and Implementation. We compare our proposed method with three groups of baselines. First is the BaseModel, *i.e.*, currently used targeting model, which is based on DSSM [7] and YouTube DNN [3]. Then we consider two negative sampling baselines: KBGAN [2] and IBPR [4]. Finally we compare with the following NN-based ranking models: NFM [6], DeepFM [5], xDeepFM [8], DIN [14] and DIEN [13].

For the RNIG and negative sampling based models, the embedding size of all features is set among 3-10. For other five ranking models, the embedding size of all features is hyper-parameter by searching in {6,8,12,16,24,32}. For all the experiments, we use Adam optimizer, the mini-batch size is 512. We use optuna [1] to search hyper-parameters including learning rate in [10^{-6},10^{-2}], L_2 regularizer in [10^{-5},10^{-2}], drop out in [0,1] and the number of MLP layers in {1,2,3,4}. Besides, our proposed RNIG also contains the size of negative candidate set $\mathcal{N}_a$, the ratio γ of displayed users in $\mathcal{N}_a$ and the weight α in L_G, which are searched in {16,32,64,80,128,160}, [0,1] and [0,1], respectively. For each method, We conducted experiments up to 100 times with different hyper-parameter settings.

[1] https://e.qq.com

3.2 Performance Comparison

Table 1 displays the performance on *Tencent ad* dataset, where we perform paired t-test between RNIG and each of baselines over 10-round results. From the results, we observe that RNIG significantly improves the ad targeting performance by training with more representative negative instances compared with BaseModel trained with uniform random negatives, with RelaImpr up to 5.65%. Besides, compared with adversarial instances, negative instances that are similar with observed negatives in several important aspects are more useful for training the targeting model. We can observe that KBGAN and BaseModel perform the same on Mean GAUC, but 0.0008 lower than *IBPR* that randomly generates negative instances from both observed and unobserved data according to a fixed probability. In addition, Mean GAUC of NN-based ranking models are all lower than BaseModel, which seems reasonable given that they are all designed for predicting user preference among different items, instead of the ad relevance among different users.

Table 1: Performance comparison between all the methods, where significant test is based on GAUC.

Group	Methods	GAUC		RelaImpr	P-value
		Mean	STD		
Base	BaseModel	0.6850	2.972e-3	-	3.196e-9
NS	KBGAN	0.6850	3.360e-3	0.00%	6.342e-8
	IBPR	0.6858	2.849e-3	0.43%	6.023e-9
Ranking	NFM	0.6719	5.936e-3	-7.08%	2.929e-10
	DeepFM	0.6716	2.890e-3	-7.22%	2.062e-15
	xDeepFM	0.6802	2.070e-3	-2.62%	2.708e-14
	DIN	0.6741	2.582e-3	-5.89%	2.331e-15
	DIEN	0.6804	2.173e-3	-2.50%	7.061e-14
Proposed	**RNIG**	**0.6955**	**8.771e-4**	**5.65%**	-

3.3 Feature Matching Strategy

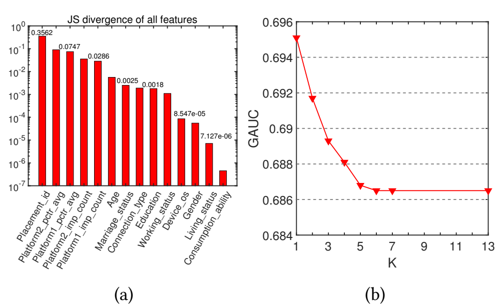

Figure 2: (a) JSD value of each user feature. (b) GAUC performance v.s. number of matched features.

In RNIG, we propose to improve the feature matching by focusing on several important features and guiding the generated negatives to be similar to the observed negatives in terms of these features. Next, we analyze the feature importance in terms of their corresponding JSD values between positive instances and observed negative instances, and then observe the change of performance when varying the number of matched features, *i.e.*, K. The results are shown in Figure 2.From Figure 2(a) we can observe that the *placement_id* feature has the largest JSD value. As *Tencent Ad* system has large number of advertising placements distributed among tens of mobile apps, it is reasonable for placement-related features to have large impact. Other important features are those related to users' historical behaviors. As for the GAUC performance, we observe a clear decrease when number of matched features increases, which indicating that matching several or even just one important feature is vital in generating representative negatives.

4 CONCLUSION

We study the problem of designing better negative instance generator for online ad targeting. To generate representative negative instances, we propose a RNIG model that leverages feature matching technique to generate reliable negative instances and further improves the representativeness of generated negatives by matching the most important feature. In the future, we plan to explore various ad targeting models to investigate the generality of our proposed RNIG model and conduct experiments on more datasets.

ACKNOWLEDGEMENTS

This work was supported in part by The National Key Research and Development Program of China under grant SQ2018YFB180012, the National Nature Science Foundation of China under 61971267, 61972223, 61861136003, and 61621091, Beijing Natural Science Foundation under L182038, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *KDD*. 2623–2631.
[2] Liwei Cai and William Yang Wang. 2018. KBGAN: Adversarial Learning for Knowledge Graph Embeddings. In *NAACL*. 1470–1480.
[3] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *RecSys*. 191–198.
[4] Jingtao Ding, Fuli Feng, Xiangnan He, Guanghui Yu, Yong Li, and Depeng Jin. 2018. An Improved Sampler for Bayesian Personalized Ranking by Leveraging View Data. In *WWW Companion*. 13–14.
[5] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *IJCAI*. 1725–1731.
[6] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*. 355–364.
[7] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*. 2333–2338.
[8] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *KDD*. 1754–1763.
[9] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. 452–461.
[10] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*. 1057–1063.
[11] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *SIGIR*. 515–524.
[12] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR*. 785–788.
[13] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *AAAI*. 5941–5948.
[14] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *KDD*. 1059–1068.