Disguise Adversarial Networks for Click-through Rate Prediction

Yue Deng, Yilin Shen, Hongxia Jin
Samsung Research America, Mountain View, CA, USA
{y1.deng, yilin.shen, hongxia.jin}@samsung.com

Abstract

We introduced an adversarial learning framework for improving CTR prediction in Ads recommendation. Our approach was motivated by observing the extremely low click-through rate and imbalanced label distribution in the historical Ads impressions. We hence proposed a Disguise-Adversarial-Networks (DAN) to improve the accuracy of supervised learning with limited positive-class information. In the context of CTR prediction, the rationality behind DAN could be intuitively understood as "non-clicked Ads makeup". DAN disguises the disliked Ads impressions (non-clicks) to be interesting ones and encourages a discriminator to classify these disguised Ads as positive recommendations. In an adversarial aspect, the discriminator should be sober-minded which is optimized to allocate these disguised Ads to their inherent classes according to an unsupervised information theoretic assignment strategy. We applied DAN to two Ads datasets including both mobile and display Ads for CTR prediction. The results showed that our DAN approach significantly outperformed other supervised learning and generative adversarial networks (GAN) in CTR prediction.

1 Introduction

Improving users' click-through rate prediction on Advertisement (Ads) is a long-term research topic in the AI community [Richardson *et al.*, 2007; Chen and Yan, 2012]. In practical advertising industry, the "click pay cost per click" business model allows the recommender accumulating high amount of revenues if they could precisely serve the right Ads to the right user. In existing works, such CTR prediction problem is conventionally tackled by supervised learning [König *et al.*, 2009], in which a classifier/discriminator is trained to map contents (feature) to the practical click/non-click behaviors (label) in historic Ads impressions.

The challenges in this prediction problem stem from the severe imbalances between clicks and non-clicks in the "traffic log" of Ads impressions. For instance, in a mobile Ads dataset [Ava, 2015], only 17% Ads impressions are clicked by users while the other 83% are all non-clicks, yielding to an extremely imbalanced label distribution with quite limited positive samples. There are in fact a number of algorithms designed to handle classification problems with imbalanced labels. According to [Japkowicz, 2000], these methods could be generally profiled into down-sampling and up-sampling categories.

Down-sampling approaches reduce the major class size and make a balanced subgroup for classification. While those methods could alleviate the label imbalance issue to some extent, they do not notice the essential pitfalls of the imbalanced classification—the information about minor class is still insufficient. In the context of Ads recommendation, we know that most Ads impressions are annoying and human may naturally dislike them. Therefore, it is more meaningful for a classifier to get more chances to know interesting Ads contents (minor class up-sampling), rather than losing opportunities to see disliked Ads (major class down-sampling).

Oversampling method tries to generate more samples in the minor group. The first prevalent over-sampling strategy is to parametrically fit a probabilistic distribution (*a.k.a.* generative model) from limited observed samples in the minor class. Then, new data points can be sampled out from such a fitted distribution [Bao *et al.*, 2017b]. Such oversampling method is only suitable for well structured data, *e.g.* gene [Bao *et al.*, 2017a]. Otherwise, it is really hard to find a reasonable parametric generative model to well depict the data generation mechanism. Unfortunately, like most real world data, the Ads feature does not exhibit obvious inherent structure. Alternatively, up-sampling methods may also synthesize new data points by combining a set of observed data points in the minor class [Chawla *et al.*, 2002; Han *et al.*, 2005]. Such strategy was built on the assumption that existing data are sufficient to span the complete space of the minor class, which is not true in practice. The Ads contents and other real world data could be highly heterogeneous. There are great chances that neither the data nor its related items has ever been observed within these limited samples. Accordingly, the assumption that new data could all be produced by combing existing observations does not make too much sense here.

While we admit the rationality of minor class augmentation, existing strategies almost suffer restrictions in handling practical problems. At least, they are not suitable for the CTR prediction problem. We noted that an ideal minor class aug-

mentation algorithm should exhibit two desired properties 1) the data generation function of it should be general and robust 2) the seeds used in it for new data generation should not solely come from limited samples in the minor class.

To fully cover the aforementioned two properties, we proposed a disguise adversarial network (DAN) for minor class augmentation (up-sampling). Our DAN was inspired by the recent progress of generative adversarial learning [Goodfellow *et al.*, 2014] and was especially designed to solve the CTR prediction problem with imbalanced Ads' labels. The proposed DAN incorporates a disguise neural network to generate more samples to enrich the minor class by disguising negative samples. The purpose of the disguise neural network is to cheat a discriminator to believe all these disguised samples are all positive. On the other hand, a discriminator neural network is also implemented to clearly assign these disguised samples into their inherent classes via an information theoretic discriminative clustering strategy.

The DAN framework was applied and tested on CTR prediction problems including both mobile and display Ads. Compared with traditional imbalanced classification algorithms and other generative adversarial networks, the proposed DAN improves in both recommendation frequency and accuracy. More impressively, our approach is also very effective when less training data are available. The DAN approach can maintain reasonable good performances even though reducing the size of training samples to 10%, which shows a promising direction for algorithm speeding up.

2 Preliminaries

DAN is inspired by the generative adversarial network (GAN) [Goodfellow *et al.*, 2014] that showed great promises in the computer vision society [Denton *et al.*, 2015]. While there are different innovations on GAN, the central concept of it can be well interpreted as a gambling process involving a generator (G) and a discriminator (D). They could be both implemented by deep neural networks (DNN). The generator maps a randomly sampled vector $z_i \sim P_z(z)$ as an image $y_i = G(z_i)$. The discriminator is designed to identify fake image z_i from real world image $x_i \sim P_{data}(x)$. In a nutshell, the gambling process could be formulated with the following min-max optimization:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x_i \sim P_{data}(x)}[\log D(x)] + \mathbb{E}_{z_i \sim P_z(z)}[\log D(G(z))] \quad (1)$$

Conditional GAN [Gauthier, 2014] is an extension of the basic GAN that takes the label information into consideration. In detail, the generator in conditional GAN takes two streams of information as the input $y_i = G(z_i, l_i)$, where l_i is the assigned categorical label of the ith generated sample. There were different implementations of this conditional GAN by either concatenating l and z as a long vector or taking them as two inputs of a multi-modal DNN [Ren *et al.*, 2016]. InfoGan is another prevalent approach that [Chen *et al.*, 2016] generalizes the categorical information into the generator by adding a mutual information term. It views the categorical information and other control information (*e.g.* the angle of the image) as side information and quantizes them as a long vector l. InfoGAN then maximizes the mutual information between the generated image y and its corresponding side information vector l.

There were also improvements on the discriminator side, *e.g.* SGAN [Odena, 2016]. The discriminator in SGAN assigns data into N classes of real data and one extra class of fake data. In the adversarial gambling game of SGAN, the generator tries to put all fake data into N real data classes but the discriminator is optimized to allocate fake data into the fake category, *i.e.* the $(N+1)th$ category. As SGAN, the main purpose of DAN is also set to enhance classification performances of the discriminator rather than distinguishing fake data from real ones.

3 Disguise Adversarial Networks

3.1 Motivation

We motivate the algorithm of Disguise Adversarial Networks (DAN) in the context of CTR prediction. As stated above, one significant problem in Ads data is the imbalanced label distribution, *i.e.* limited positive samples (clicks) *v.s.* abundant negative samples (non-clicks). A natural solution to this problem is to enrich the information of the minor positive group. However, it is perhaps impossible to directly seek for more positive samples from users in real world. Inspired by GAN, we consider an alternative approach to generate more positive samples via a generative deep neural network.

We consider a way called "Ads Makeup". The general assumption is that if we could slightly change some properties of the disliked Ads, it may have a chance to become an interesting one. We discuss the intuition behind this Ads makeup approach by taking a mobile Ads "super bowl game" as an example. We consider that the Ads feature vector contains one entry denoting the show time of the Ads. If the "super bowl game" Ads impression was pushed to the user at 10:00 AM (with the Ads' time feature denoted as 'morning'), this Ads may not be clicked because morning time is always the business hour. Accordingly, a non-click record about "super bowl Ads" is accumulated in the training data. But it does not necessarily mean the "super bowl Ads" itself is bad and is not liked by the user. Alternatively, if we fix all other features in this Ads the same but only change the Ad's show time feature from 'morning' to 'night', this disliked Ads may become a popular one and gain a click.

Following the rationality discussed above, we believe there should be a huge amount of non-clicks in the historic data that could be converted to interesting ones with slight modifications. We design a "Disguise Neural Network" (Fig 1) to transform and makeup non-clicks. However, there is still a lack of the metric to evaluate the quality of such an Ads disguise approach. We hence define a "Discriminator Neural Network" to mimic real user's behavior on Ads clicking. From the aspect of the Disguise Neural Network, its objective is to ultimately disguise the Ads and encourage the discriminator to classify these disguised Ads as positive. In an adversarial view, the discriminator should avoid being cheated by the disguise network and come up with a "smart" objective to identify these disguised Ads.

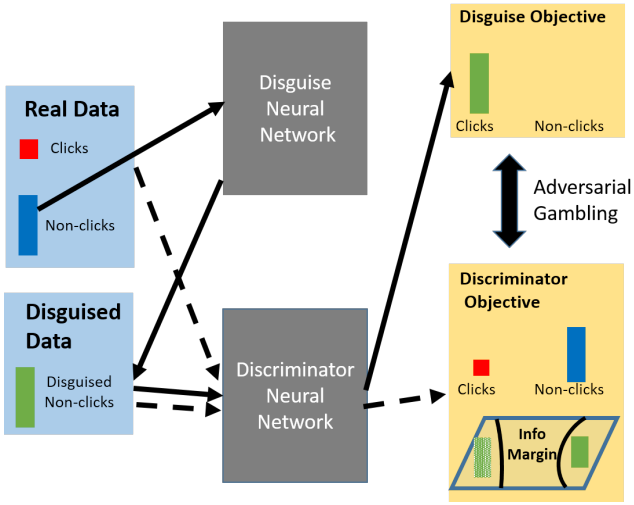

Figure 1: An overview of Disguise Adversarial Network (DAN). The solid (*resp.* dotted) line indicates the learning processes of the Disguise (*resp.* Discriminator) Neural Network.

In detail, we discuss two possible objectives for the discriminator. The first option is the "tough objective" which is set up to assign all disguised Ads to the negative class (non-clicks). However, such objective is too strict to be reasonable. There should be a number of successfully disguised Ads, such as the super bowl Ads, that could be assigned to the positive class after makeup. If treating all disguised Ads as non-clicks, these successfully disguised Ads and their corresponding favorite feature content may not be correctly fed to the positive side of the classifier. Therefore, we consider a more reasonable "mild objective" for the discriminator. It should assign the disguised data into two classes (both clicks and non-clicks) according to their inherent properties. To achieve this, we design a clustering objective for the discriminator to encourage a large margin between two classes. We will explain how to mathematically design the discriminative objective in Section 3.3. But please keep in mind here that the "mild objective" could smartly treat both successfully and unsuccessfully disguised samples without any biased pre-judgment.

Although we have motivated the Disguise Adversarial Networks (DAN) in the context of Ads CTR prediction, we still remind readers that all subsequent discussions about DAN model are also seamlessly adapted to general classification problems. In the next two subsections, we will elaborately illustrate formulations of these two adversarial objectives in DAN.

3.2 Disguise Objective

The disguise learning part was linked by solid arrows in Fig.1. There are M positive samples (clicks, red bar in Fig.1) $x^+ \sim P_{\Omega^+}(x)$ and N negative samples (non-clicks, blue bar in Fig.1) $x^- \sim P_{\Omega^-}(x)$, $M < N$ for imbalanced cases. Ω^+ (*resp.* Ω^-) represents the positive samples' (*resp.* negative samples') space. We introduce a Disguise Neural Network $T(\cdot)$ to map the negative samples to $z = T(x^-) \sim P_{\hat{\Omega}}(z)$, where $\hat{\Omega}$ is space spanned by those disguised samples z (green bar in Fig.1). z shares the same dimension as the real world data x^+ and x^-.

The learning purpose of disguise neural network is to encourage the equivalence of two distributions $P_{\Omega^+}(x)$ and $P_{\hat{\Omega}}(x)$. From the "disguise objective panel" in Fig.1, it is apparent that the disguise objective intends to assign all disguised data to the positive class with the discriminator neural network $D(\cdot)$. The last layer in neural network D is connected with a sigmoid output to indicate the probability that the input sample is positive. Therefore, we could mathematically write the disguise learning objective:

$$\mathcal{L}_1(T, D) = -\mathbb{E}_{x^- \sim P_{\Omega^-}}[\log D(T(x^-))] + \lambda \|T(x^-) - x^-\|_1 \quad (2)$$

The first term in Eq.2 is the KL divergence between the output probability and the positive distribution. It is a part of the cross-entropy term. The second term uses a ℓ_1 distance to restrict the disguise process could only 'slightly' change the original data.

We would like to highlight the conceptual differences between this disguise neural network and the generator neural network in GAN [Goodfellow *et al.*, 2014]. GAN is designed to generate a brand new sample that never exists in real world. It thus requires sampling a random seed (vector) to feed in the generator and to produce a fabricate output. In our DAN, everything is from real world data and the inputs to the disguise neural network are practical negative samples. It just adjusts negative data to make them comply with the distribution in the positive class. Therefore, we prefer to use the word "disguise" rather than "generator" in our approach. Such characteristic is apparent in Fig.1 that no random vector sampling function was observed in DAN's infrastructure.

3.3 Discriminator Objective

As indicated by dotted arrows in Fig.1, the discriminator neural network takes both real and disguised data as inputs. Therefore, both of these two types of data contribute to the final loss in the discriminator neural network. The real data come with the user's clicking labels, so there is no difficulty to define a supervised loss for this part of data via the cross-entropy loss. The second part of the loss is the attitude of the discriminator neural network about disguised data. We will elaborately discuss it in the following part.

As stated in Section 3.1, there were two possible ways for the discriminator neural network to handle disguised data via either the "tough loss" or the "mild loss". Here, we adopt the more reasonable "mild loss" that allows some negative data to be converted as positive ones. However, the difficulty is that we have no idea about which part of the disguised data behave like positive samples and which part are still negative. To address such a problem, we follow an existing work to maximize the information theoretic margin between positively and negatively disguised samples [Krause *et al.*, 2010]. Unlike other margins in supervised learning, such information theoretic margin is absolutely unsupervised. This approach is also termed as discriminative clustering in multiple early works [Deng *et al.*, 2016a][Shi and Sha, 2012].

We assume there are N unlabeled points. When assigning these N points to 2 classes ($l = 1$ or $l = 0$) by a discrimi-

nator neural network $D(\cdot)$, the assignment confidence of the discriminator could be well characterized by the following additive conditional entropy $\mathcal{M}_D(x)$,

$$\begin{aligned}\mathcal{M}_D(x) &= \tfrac{1}{N}\sum_i H(l|x_i) \\ &= -\tfrac{1}{N}\sum_i \{D(x_i)\log[D(x_i)] + (1-D(x_i))\log[1-D(x_i)]\}\end{aligned} \tag{3}$$

As indicated in [Krause *et al.*, 2010], the conditional entropy captures the discriminative clustering margin between two classes and hence we call $\mathcal{M}_D(x)$ as the information theoretic margin in our approach. This term should be minimized to encourage a large margin between the clustering results.

We combine the aforementioned two parts of losses altogether and form the final training objective for the discriminator neural network:

$$\begin{aligned}&\mathcal{L}_2(T, D) = -\mathbb{E}_{x^-\sim P_{\Omega^-}}[\log(1-D(x^-)] \\ &-\mathbb{E}_{x^+\sim P_{\Omega^+}}[\log(D(x^+)] + \eta\mathcal{M}_D[T(x^-)]\end{aligned} \tag{4}$$

where the first two terms come from the cross-entropy of the real world labeled data and the last term penalizes the margin of disguised data.

To note, our approach is quite different from traditional GAN in which the discriminator is only set up to classify whether a sample is real or fake. Our discriminator objective function shares a very similar framework as semi-supervised classification. In traditional semi-supervised learning, the unlabeled data used for training are pre-fixed. But unsupervised samples in DAN are produced by a disguise neural network. Therefore, in our approach, the discriminator neural network may get a chance to access to different versions of disguised samples in multiple iterations. More importantly, the disguise neural network also evolves and could produce more difficult samples for the discriminator along with training iterations going on. Therefore, compared with traditional semi-supervised learning, DAN could be optimized with more diverse and difficult unsupervised samples in the training phase. This is the exact reason why we believe DAN can achieve much better performance than traditional semi-supervised methods that only adopt pre-fixed unlabeled samples as assistance.

The training of DAN is involved in a bilevel optimization [Bard, 2013] that requires minimizing the disguise and the discriminator losses in turn. Bilevel optimization was used in a number of practical learning problems including GAN [Radford *et al.*, 2015], sparse learning [Deng *et al.*, 2013] and reinforcement learning [Lillicrap *et al.*, 2015; Deng *et al.*, 2017a]. We divide all training samples into multiple mini-batches and iteratively feed these mini-batches to train DAN. Algorithm 1 summarized our detailed training steps.

Algorithm 1: Training DAN

Input : Ads Features X and their labels Y
Initialization: Randomly Initialize all parameters Θ_T of disguise neural network T and Θ_D of discriminator Neural Network D; Find all negative samples X^- in X

```
for epoch=1...K do
    for all mini-batches in Training Data do
        Minimize the the disguise loss in Eq.2 with X^- and update Θ_T in T
        Feed all negative samples X^- through disguise neural network with current Θ_T and get the the disguised samples Z
        Feed the discriminator with both supervised data (X) and unlabeled disguised data Z = T(X^-); treat Z as unsupervised samples
        Minimize the discriminator loss in Eq.4 and update Θ_D in D
    end
end
```

Output : Discriminator Neural Network D with parameter Θ_D

4 Experiments

4.1 Experimental Setup

We evaluate the performance of GAN in CTR prediction on two datasets including both display and mobile Ads. Display Ads dataset record the Ads impressions from Criteo in an one-week time period that include various undisclosed features along with the click labels [Cri, 2015]. The feature for each display Ads is composed of 13 integer and 26 categorical features. The mobile Ads dataset come from [Ava, 2015] that cover mobile Ads impressions in 10-days period. All provided attributes in this dataset are anonymous categorical features such as "device_type", "app_id" etc. Because both of these two datasets contain categorical features, we tried to convert each categorical feature as a binary vector indicating which category the certain item belongs to. A brief summary about these two datasets were provided in Table 1. The last column "Dim" in the table reports feature dimensions of each Ads dataset after converting the categorical feature to numerical values. These numerical features are used in our DAN and other competing methods.

Table 1: Summaries about two CTR datasets

Datasets	Total logs	Period	CTR	Dim
Display	46 million	7 days	0.26	228
Mobile	40 million	10 days	0.17	100

It is also noted that Ads impressions in both datasets were accumulated in an ascending order over time. We obey the time order and uniformly divide each dataset as 100 bulks. Each bulk contains 1% Ads impressions in a certain period and different bulks are consecutive in time. It is important to note the time-varying effects on Ads impressions that one specific Ads could be quite hot in a short period but quickly losing its popularity afterwards. To fully take this time-varying effect into consideration, we train our model with the last 20 bulks of Ads impressions in the history and predict the CTR in the next 5 bulks. The whole process was incrementally moved forward following the time order of bulks.

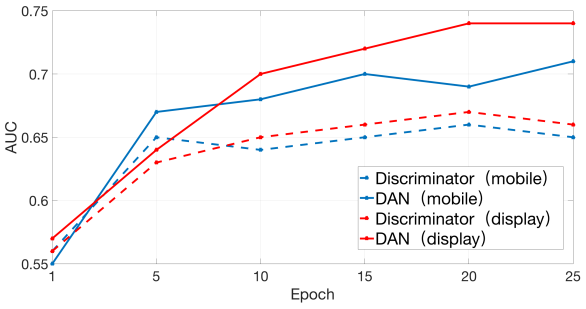

Figure 2: Comparisons of the same discriminator trained with and without disguise adversarial strategies.

Many previous works have indicated that conventional classification accuracy indicators such as precision and recall are not suitable for evaluating the CTR performance because the data themselves are highly imbalanced [Yan *et al.*, 2014]. Therefore, they suggest the use of area-under-curves (AUC) as a robust indicator to validate the performances of different CTR algorithms. AUC should be reported in the range of $(0.5, 1)$, with AUC=0.5 meaning random guessing. A higher AUC implies a better classification result. We also prefer to use AUC as a main indicator to report CTR prediction performances.

While the neural networks in DAN could be implemented with arbitrary structures, such as the fancy hierarchical structure in [Deng *et al.*, 2016b], we still prefer to implement them with the simple multi-layer perception (MLP). In detail, both the disguise and discriminator neural networks are configured with 4 layers and each layer contains 32 nodes. The output layer of the disguise neural network shares the same nodes' number as its input layer. The output of the discriminator neural network is a sigmoid function indicating the clicking probability. We do not carefully tune the network structure because this general setting is already good enough to produce reasonable results as reported below.

4.2 Overall Evaluation

To visualize the effectiveness of adversarial training in DAN, we first compare the DAN's performance with a discriminator neural network. For fair comparisons, both the DAN and the discriminator neural network share the same network structure as stated above. Mathematically, the discriminator was trained with the supervised loss while DAN was optimized with the adversarial loss defined in Eq.4. We train both the DAN and the discriminator neural network for 25 epochs. At the end of every $5th$ epoch, the corresponding CTR performances on testing data were calculated and reported in Fig.2.

In the figure, we used dotted and solid lines to represent the supervised discriminator and DAN, respectively. Different colors correspond to results on different datasets. At the beginning of training, both supervised and adversarial discriminators achieved similar performances. But after the $5th$ epoch, the adversarial training strategy leads to an obvious improvement in AUC on both datasets. Besides, the trend of curves also empirically imply the evolutions of DAN in the gambling process. It achieved relatively higher performance than ever with more iterations going on.

We also compare our DAN with other competing methods. In general, these competing methods could be divided as four types as listed in the first column of Table 2. In detail, Deep Neural network (DNN) and Support Vector Machine (SVM) were chosen as the supervised algorithms. We further chose cluster centroid algorithm (Centroid) [Ganganwar, 2012], adaptive synthetic sampling (ADASYN) [He *et al.*, 2008], synthetic minority over-sampling technique (SMOTE) [Chawla *et al.*, 2002] as representative imbalanced classification techniques, in which the first one is a down-sampling strategy and the later two are up-sampling approaches.

We also consider other GAN methods as competitors. However, except for SGAN [Odena, 2016], most other GANs are not designed for classification purpose. Therefore, we slightly modified existing GAN (Modified GAN) to make it applied to the CTR problem. In the implementation, we first ran a traditional GAN to generate many fake Ads. After the GAN training, the generated fake Ads that could successfully pass the authenticity verification step (*i.e.* by the discriminator in traditional GAN) were used as unsupervised samples altogether with supervised CTR records for semi-supervised training as in [Deng *et al.*, 2016a]. Finally, different DAN implementations were also discussed. In addition to the DAN model presented in the paper, we alternatively consider two other variations. The first variation is the tough-loss DAN (TL-DAN) as stated in Section 3.1. The other variation is the Noise seeded DAN (Noise GAN) that is designed to use a random generator to generate more samples for the discriminator. In Noise DAN, the discriminator neural network is the same as in DAN but fake samples are all generated from random noise.

We report the performances of these methods in Table.2. In addition to the AUC indicator, we also report two other indicators that may intuitively explain how different methods perform in the practical recommendation scenario. The first indicator is the recommendation frequency (Rd-frequency) that reports how frequently a recommendation decision is rendered. In our setting, a recommendation is only made when the CTR prediction model indicates to the positive class. Therefore, the recommendation frequency means the percentage of positive labels predicted by the recommendation method. Further, we investigate the CTR based on the recommended items (Rd-CTR) that calculates the CTR among all recommendations. Such evaluation strategy is similar as the off-line evaluation policy introduced in [Li *et al.*, 2010].

We observed that all machine learning models render pretty similar but relatively low recommendation frequency. Even though based on the best performed DAN model, its recommendation frequency is only 11% and 5% on two datasets. But, the DAN improves Rd-CTR for a large amount. From the AUC score, it is observed that our DAN significantly outperforms other learning methods. Only the DAN model could achieve the AUC score higher than 0.7 on two datasets. By comparing two datasets, performances on Mobile dataset is obviously worse than on Display dataset. This is partially because the Criteo display data were already up-sampled by the data provider to maintain some balance level and the mobile data is much closer to real-world data. In addition, we

Table 2: Performances of different algorithms for CTR prediction and recommendation

		Display Ads			Mobile Ads		
		Rd-frequency	Rd-CTR	AUC	Rd-frequency	Rd-CTR	AUC
Supervised	DNN	0.07	0.53	0.69	0.02	0.51	0.67
	SVM	0.07	0.50	0.62	0.03	0.47	0.63
Imbalance	Centroid	0.08	0.49	0.66	0.06	0.51	0.64
	ADASYN	0.08	0.62	0.70	0.04	0.55	0.66
	SMOTE	0.09	0.57	0.67	0.05	0.53	0.65
GAN	Modified-GAN	0.09	0.57	0.69	0.04	0.52	0.63
	SGAN	0.09	0.61	0.68	0.04	0.51	0.67
DAN	TL-DAN	0.04	0.68	0.69	0.01	0.58	0.64
	Noise-DAN	0.08	0.62	0.71	0.02	0.53	0.69
	DAN	0.11	0.66	0.75	0.05	0.57	0.73

Table 3: The AUCs by training DAN with less samples

	100%	50%	20%	10%
Display	0.75	0.73	0.71	0.70
Mobile	0.73	0.71	0.70	0.68

have not observed obvious advantages of existing imbalanced classification algorithms when compared with deep learning approaches. Such claim is apparent by comparing all methods in the "Imbalanced" category with the DNN result in the "Supervised" category in Table 2. It implies the deep learning model perhaps already owns some inherent properties to robustly treat samples with imbalanced label distributions. But the adversarial deep learning (such as our DAN) models further improve traditional supervised deep learning.

4.3 Training Complexity

Deep learning framework always gains the reputation of "heavy to train". Among all reported competing methods, the DAN requires the heaviest training complexity. It is conceivable because DAN involves two (deep) neural networks. Even worse, the learning objectives of these two neural networks are designed to go against each other. The gambling essence of adversarial training inevitably adds complexity to the optimization. Our practical training always requires 3 hours to finish 25 epochs on 9 million historic data with 4 GPUs parallelized. When using the same data to train a supervised neural network, the consumed time is only about 1 hour. Rather than hardware improvement, we also consider alternative approach to improve the training speed by reducing total training samples as in Table 3. The AUC results for DAN are calculated on out-of-sample data.

The experimental results here comply with most findings in deep learning that a larger training size may lead to a higher classification performance. But when taking training complexity into consideration, decreasing the sample size in DAN may be a good trade-off. It is observed that AUCs on testing data do not drop significantly even though decreasing the training sample size to 10%. The AUC of DAN is still better than most approaches in Table 2 with 10% training data. Such plausible property may be partially due to the self-data-augmentation mechanism encoded in the adversarial learning framework. Therefore, it is reasonable to reduce the total training size while maintaining a good performance.

5 Discussions

While our DAN was motivated and innovated in Ads CTR prediction, we should remind readers that it is meanwhile flexibly applied to other classification problems involving imbalanced data. While the algorithm and experiment in this paper were mainly designed on the two-class classification problem, it is still possible to extend this framework to handling multi-class tasks. One intuitive extension is just to implement DAN in an one vs others manner [Deng *et al.*, 2017b]. But it requires training at least C (the class number) different DANs which is impractical when C is large. We will consider more efficient approaches in our future works. One downside that have been observed from experiment is the extremely low recommendation frequency as listed in Table 2. We may consider other approaches, *e.g.* training multiple machines with different initializations or configurations, to improve the recommendation frequency in an ensemble manner.

References

[Ava, 2015] Avazu mobile ads ctr dataset. https://www.kaggle.com/c/avazu-ctr-prediction/data, 2015.

[Bao *et al.*, 2017a] Feng Bao, Yue Deng, Mulong Du, Zhiquan Ren, Qingzhao Zhang, Yanyu Zhao, Jinli Suo, Zhengdong Zhang, Meilin Wang, and Qionghai Dai. Probabilistic natural mapping of gene-level tests for genome-wide association studies. *Brief Bioinform*, (bbx002), 2017.

[Bao *et al.*, 2017b] Feng Bao, Yue Deng, Yanyu Zhao, Jinli Suo, and Qionghai Dai. Bosco: boosting corrections for genome-wide association studies with imbalanced samples. *IEEE Transactions on NanoBioscience*, PP(99):1–1, 2017.

[Bard, 2013] Jonathan F Bard. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media, 2013.

[Chawla *et al.*, 2002] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *JAIR*, 16:321–357, 2002.

[Chen and Yan, 2012] Ye Chen and Tak W Yan. Position-normalized click prediction in search advertising. In *ACM SIGKDD*, pages 795–803. ACM, 2012.

[Chen *et al.*, 2016] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Nips*, pages 2172–2180, 2016.

[Cri, 2015] Criteo display ads ctr dataset. https://www.kaggle.com/c/criteo-display-ad-challenge, 2015.

[Deng *et al.*, 2013] Yue Deng, Qionghai Dai, Risheng Liu, Zengke Zhang, and Sanqing Hu. Low-rank structure learning via nonconvex heuristic recovery. *IEEE TNNLS*, 24(3):383–396, March 2013.

[Deng *et al.*, 2016a] Yue Deng, Feng Bao, Xuesong Deng, Ruiping Wang, Youyong Kong, and Qionghai Dai. Deep and structured robust information theoretic learning for image analysis. *IEEE TIP*, 25(9):4209–4221, Sept 2016.

[Deng *et al.*, 2016b] Yue Deng, Zhiquan Ren, Youyong Kong, Feng Bao, and Qionghai Dai. A hierarchical fused fuzzy deep neural network for data classification. *IEEE TFS*, 2016.

[Deng *et al.*, 2017a] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE TNNLS*, 28(3):653–664, March 2017.

[Deng *et al.*, 2017b] Yue Deng, Yanyu Zhao, Zhiquan Ren, Youyong Kong, Feng Bao, and Qionghao Dai. Discriminant kernel assignment for image coding. *IEEE Transactions on Cybernetics*, 47(6):1434–1445, June 2017.

[Denton *et al.*, 2015] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Nips*, pages 1486–1494, 2015.

[Ganganwar, 2012] Vaishali Ganganwar. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4):42–47, 2012.

[Gauthier, 2014] Jon Gauthier. Conditional generative adversarial nets for convolutional face generation. *Technical Report*, 2014:5, 2014.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Nips*, pages 2672–2680, 2014.

[Han *et al.*, 2005] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*, pages 878–887. Springer, 2005.

[He *et al.*, 2008] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *IJCNN*, pages 1322–1328. IEEE, 2008.

[Japkowicz, 2000] Nathalie Japkowicz. Learning from imbalanced data sets: a comparison of various strategies. In *AAAI workshop on learning from imbalanced data sets*, volume 68, pages 10–15. Menlo Park, CA, 2000.

[König *et al.*, 2009] Arnd Christian König, Michael Gamon, and Qiang Wu. Click-through prediction for news queries. In *SIGIR*, pages 347–354. ACM, 2009.

[Krause *et al.*, 2010] Andreas Krause, Pietro Perona, and Ryan G Gomes. Discriminative clustering by regularized information maximization. In *Nips*, pages 775–783, 2010.

[Li *et al.*, 2010] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670. ACM, 2010.

[Lillicrap *et al.*, 2015] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[Odena, 2016] Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

[Radford *et al.*, 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[Ren *et al.*, 2016] Zhiquan Ren, Yue Deng, and Qionghai Dai. Local visual feature fusion via maximum margin multimodal deep neural network. *Neurocomputing*, 175:427–432, 2016.

[Richardson *et al.*, 2007] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In *WWW*, pages 521–530. ACM, 2007.

[Shi and Sha, 2012] Yuan Shi and Fei Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. *arXiv preprint arXiv:1206.6438*, 2012.

[Yan *et al.*, 2014] Ling Yan, Wu-jun Li, Gui-Rong Xue, and Dingyi Han. Coupled group lasso for web-scale ctr prediction in display advertising. In *ICML*, pages 802–810, 2014.