Domain Adaptation in Display Advertising: An Application for Partner Cold-Start

Karan Aggarwal
University of Minnesota
Minneapolis, MN
aggar081@umn.edu

Pranjul Yadav
Criteo AI Labs
Palo Alto, CA
p.yadav@criteo.com

S. Sathiya Keerthi
Criteo AI Labs
Palo Alto, CA
s.selvaraj@criteo.com

ABSTRACT

Digital advertisements connects partners (sellers) to potentially interested online users. Within the digital advertisement domain, there are multiple platforms, *e.g.,* user re-targeting and prospecting. Partners usually start with re-targeting campaigns and later employ prospecting campaigns to reach out to untapped customer base. There are two major challenges involved with prospecting. The first challenge is successful on-boarding of a new partner on the prospecting platform, referred to as partner cold-start problem. The second challenge revolves around the ability to leverage large amounts of re-targeting data for partner cold-start problem.

In this work, we study domain adaptation for the partner cold-start problem. To this end, we propose two domain adaptation techniques, SDA-DANN and SDA-Ranking. SDA-DANN and SDA-Ranking extend domain adaptation techniques for partner cold-start by incorporating sub-domain similarities (product category level information). Through rigorous experiments, we demonstrate that our method SDA-DANN outperforms baseline domain adaptation techniques on real-world dataset, obtained from a major online advertiser. Furthermore, we show that our proposed technique SDA-Ranking outperforms baseline methods for low CTR partners.

CCS CONCEPTS

• **Information systems → Display advertising**.

KEYWORDS

Domain adaptation; Cold start; Digital advertisement; Retargeting; Prospecting

ACM Reference Format:
Karan Aggarwal, Pranjul Yadav, and S. Sathiya Keerthi. 2019. Domain Adaptation in Display Advertising: An Application for Partner Cold-Start. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3298689.3347004

1 INTRODUCTION

Digital advertisement industry aims to match product related advertisements to potentially interested internet users. It is a multi-billion dollar industry growing at a rapid pace [6]. Digital advertising can broadly be classified into two major categories: display-based advertising [1, 14] and search-based advertising [19, 22, 43]. In display-based advertising, the goal is to display relevant product advertisements to users when they visit websites. In search-based advertising the aim is to display the product advertisements after users have expressed their intent via a search query. The sellers who want to target users for their products are referred to as partners. The websites on which the product advertisements of the partners are served are referred to as publishers.

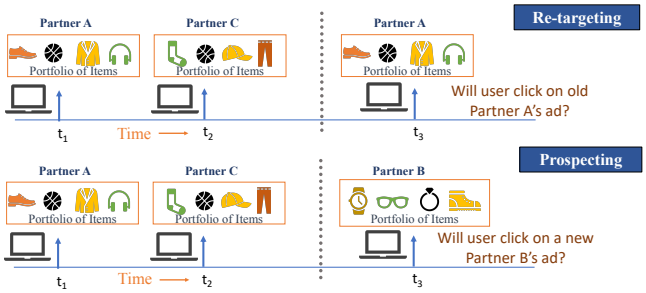


Figure 1: Schematic of user re-targeting and prospecting. For the re-targeting, the problem is to predict whether the user will click an advertisement from partner A, who they have been recently exposed. For prospecting, the problem is to predict whether the user will click on an advertisement from partner B whom they have not been recently exposed.

Display-based advertisements can be further classified into two major types: 1) Re-targeting and 2) Prospecting. Re-targeting [23], refers to a scenario when product advertisements of the partner are displayed to users who have also been shown that partner's product advertisement recently. Prospecting, on the other hand, refers to a scenario for advertisement displays to users who have never been exposed to the partner's advertisements in the past. Figure 1 shows the schematic of the two platforms. This paper is focused on improving the prospecting solution.

In order to maximize online sales, partners should ideally utilize both the platforms, *i.e.,* re-targeting and prospecting. However, in reality, partners usually start with re-targeting campaigns to maximize sales from old and known users and explore prospecting campaigns later to identify new users. This creates a regular need for onboarding new (cold-start) partners on the prospecting platform. A major requisite for a successful prospecting campaign is to identify new users who have a high propensity to engage with partner's products. Several metrics can be used to quantify the engagement of a user with partner advertisements. In this paper, we quantify this engagement using the propensity of the user to click on a product

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09…$15.00
https://doi.org/10.1145/3298689.3347004

advertisement, also known as Click-Through-Rate (CTR). CTR is a commonly used metric in digital advertisement literature [7, 20, 25] to compare models.

For any prospecting platform, recommending several thousands of users from a pool of billion of users to a partner specific campaign is a challenging problem. There are two significant challenges involved. The first challenge revolves around the regular need for successful on-boarding of a new partner on the prospecting platform. This challenge is hereby referred to as the **partner cold-start problem**. Cold-start is a well-studied recommendation systems problem with a primary focus on recommendations for a new user or item [3, 39, 40]. Particularly, in the area of the digital advertisement, it has been studied in the context of new product recommendation [41], new users [34][1], and click-to-sale conversion [11]. However, these are fundamentally different problems than the partner cold-start problem. The difference lies in the fact that a partner has a portfolio of items that can range in thousands or more, spanning different price ranges and categories. *Hence, partner cold-start is not a traditional problem of user-item affinity but user affinity to a partner selling a portfolio of items.* It is important to study the problem at this granularity since the partners approach the platform not with a pre-defined quota of individual items but with a set of items. This is the first paper to study the partner cold-start problem, instead of items or users.

The second challenge revolves around the ability to leverage large amounts of re-targeting data for prospecting partner cold-start. This challenge arises since, in the re-targeting scenario, a user has already interacted with the partner while in the prospecting scenario, the user has not interacted with the partner. Hence, re-targeting and prospecting are quite different from the perspective of user psyche. In other words, in re-targeting, we have some history about the user's interest towards the partner, whereas in prospecting we do not have a history of the user's interest towards the partner. The challenge of using a much larger re-targeting data (source domain) for CTR prediction in prospecting (target domain) is a domain adaptation problem. Domain adaptation, a sub-topic of transfer learning, has recently garnered substantial attention in the recommendation community [2, 21].

To overcome the challenges mentioned above arising due to domain adaptation for partner cold-start, we propose first set of methods that utilize deep learning based domain adaptation techniques. In particular, we propose two methods: Supervised Sub-domain Domain Adaptation Neural Network (SDA-DANN) and Ranking loss based Supervised Domain Adaption (SDA-Ranking). SDA-DANN modifies the domain invariance loss of the original DANN [13] method, in a class-wise fashion to utilize the class labels. We also incorporate similarity between the cold-start partner and other partners (referred to as sub-domains) in SDA-DANN. The proposed SDA-Ranking uses a ranking based loss for domain adaptation. The key contributions of this paper are as follows:

- We present the first set of methods to exploit domain adaptation for partner cold-start by incorporating sub-domain similarities (product category level information). We refer to this model as SDA-DANN.
- We introduce a supervised domain adaptation approach termed as SDA-Ranking by leveraging a ranking loss for the domain adaptation task.
- We perform the first study on partner level cold-start problem using domain adaptation while the prior literature focuses exclusively on the item or user cold-start. We demonstrate the effectiveness of our proposed approach on a real-world data-set.

[1] Although this paper also concerns user recommendation, it is different from blank state cold-start [34] with no additional knowledge, unlike our case where with a lot of related information about users from re-targeting.

2 RELATED WORK

In this section, we review the related literature to the focus of this work. We categorize it into three themes: i) CTR prediction, ii) Domain Adaptation, and iii) Cold-start methods.

CTR Prediction. Predicting clicks on a digital advertisement display is one of the central problems in the online advertisement domain [25, 36]. CTR is a popular way of evaluating the display based advertisement models. CTR is a well studied problem with extensive modeling techniques proposed in the literature, including logistic regression [7, 25], log-linear models [1] and decision trees [18]. Recently, several non-linear models such as Field-aware factorization machines [20] or based on deep learning have been proposed for the CTR prediction task showing state-of-the-art results. In particular, Juan et al. [20] proposed Field aware Factorization Machines for classifying large sparse data showing that second-order interaction terms between the features make a considerable difference in the prediction accuracy. The deep learning methods for CTR can be categorized into these themes: models with higher order interaction terms using deep networks [10, 15, 24, 45, 48], sequential models [29, 46, 47], and multimedia content based models [8, 44]. Zhang et al. [46] modeled the user sequences for the CTR prediction using a recurrent neural network (RNN). The multimedia-based models exploit newer sources of structured data like images and text in addition to traditional features, the areas where deep learning has shown the most impact. The other significant areas of research have been with using the keyword queries for optimization of the CTR in search and retrieval settings [9, 12, 35]. In this work, we use a deep neural architecture suited for the domain since the focus of this work is not proposing new deep learning architectures but utilizing domain adaptation techniques for partner cold-start problem.

Domain Adaptation Methods. Domain adaptation is a technique for transferring the knowledge from a domain with abundant data (source) to another data starved related domain with zero or a few labels (target) to improve the prediction on the target domain [31]. Domain adaption is a particular case of transfer learning when the prediction task across the domains is the same, while transfer learning refers to general purpose knowledge transfer across domains and tasks. Transfer learning has shown tremendous results in various domains, especially in the areas of vision and natural language processing [13, 27, 28, 37]. Domain adaptation can be categorized into supervised, semi-supervised, and unsupervised domain adaptation [31]. Domain adaptation has also been deployed for recommender systems as well as CTR prediction. Bickel et al. [4] provide a general purpose framework for transfer learning in the

Table 1: Notation Table

Notation	Explanation
$\mathcal{D}^s$	Source domain data
$\mathcal{D}^t$	Target domain data
f	mapping from input x to y
g	encoder mapping from x to a latent space
x	Input features
y	Output label
$p(x)$	Probability distribution of x
$p_u(x)$	Marginal distribution p with respect to variable u
P	Set of all partners
P_m	Partner missing in the target (prospecting) domain
$\neg P_m$	Set of partners except the missing partner P_m
$\mathcal{L}_c$	Classification loss
$\mathcal{L}_U$	Unsupervised domain adaptation loss
$\mathcal{L}_{\text{DI}}$	Domain invariance loss
$\mathcal{L}_S$	Supervised domain adaptation loss
$\mathcal{L}_D$	Dissimilarity loss for supervised Transfer learning loss
Pros($\neg P_m$)	Prospecting data for all partners except P_m
Pros(P)	Prospecting data for all partners
Ret(P)	Re-targeting data for all partners
K	Number of classes for the classification task of interest (2 in our case)
$\mathbf{d}(.,.)$	Distance metric like $\mathcal{H}$ divergence between two distributions
$\mathbf{s}(.,.)$	Similarity loss between two distributions

advertisement domain. Dalessandro et al. [11] used transfer learning using the data from the user's advertisement campaign visit on user's post-campaign conversion - hence, across different tasks. They use the prior learned from the source task as a regularizer for the logistic regression on the target task. Su et al. [41] used transfer learning for improving the click prediction from the data-rich product to a data scarce target product. In the recommender systems, there have been a few works using transfer learning which focus on the imputing the missing features for new users in target domain [17, 30, 32], for diverse recommendations [33], or across domains [2, 26]. We demonstrate the effectiveness of domain adaptation by incorporating sub-domain similarities (product category level information).

Cold-start. Cold-start is the problem of being able to make recommendations/predictions in the absence of data from the item or user of interest. There is a vast literature in the recommender systems [3, 38–40, 42] for the cold-start - both item and user cold-start. However, these methods are focused entirely on cold-start with-in the same domain. As previously mentioned, Dalessandro et al. [11] show the application of transfer learning for new domain task of post-campaign conversion prediction. However, in this work, our focus is not on the item or user cold-start but for a partner cold start for a different target domain. Hence, we instead focus on the scenarios where the partner data is not available for the cold-start, rather than focusing on pure domain adaptation.

3 PROBLEM STATEMENT

We have been given re-targeting data (source) $\mathcal{D}^s = (x, y)^s$ and prospecting data (target) $\mathcal{D}^t = (x, y)^t$. The task is to learn a mapping $f : x \rightarrow y$, with $y \in \{\text{Click}, \text{NoClick}\}$. However, the source mapping $f^s(\cdot)$ and the target mapping $f^t(\cdot)$ would be different owing to the difference in the domains. The re-targeting data covers all partners. On the other hand, the prospecting data covers all partners except one. There is a partner of interest, $p = P_m$ who is not covered by the prospecting data, and this is the partner for whom we want to cold-start.

With the above terminology, we formulate the domain adaptation problem for the **cold-start for one prospecting partner**. As the data $\mathcal{D}^t_{\{p=P_m\}}$ for a partner of interest $p = P_m$, grows from 0 to n, the classifier error for the instances of partner $p = P_m$, would decrease. However, the *main focus of this paper is on the cold-start scenario*, where we do not have any data for the partner in the target (prospecting) domain, *i.e.*, $|\mathcal{D}^t_{\{p=P_m\}}| = 0$. Note that $\mathcal{D}^s_{\{p=P_m\}}$ is non-zero and this allows domain adaptation to transfer useful knowledge to the prospecting solution for $p = P_m$. The aim of this paper is to find effective methods for doing this.

4 BACKGROUND AND PRELIMINARIES

In this section, we present the background of domain adaptation techniques. Domain adaptation literature can be broadly divided into two categories: unsupervised and supervised. We describe these two types in more detail along with the formulation of an instantiation of each type: DANN (unsupervised) and SDA-CCSA (supervised).

4.1 Unsupervised Domain Adaptation

For the unsupervised case, assumption is that source domain has prediction task labels, but target domain samples do not have associated labels. The idea here is to align the f^s and f^t with each other, using transformation $g(.)$. g is a transformation of the original feature space to a space that is invariant to the domains, which is learned. Recently, unsupervised domain adaptation has been deployed by several works [2, 5, 13]. Let us now describe one of the approaches that combine classification loss with an adversarial loss for unsupervised domain adaptation [13]. The unsupervised loss consists of two terms, classification loss $\mathcal{L}_c$ and adversarial loss $\mathcal{L}_U$. The classification loss is given by:

$$\mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{\phi}|x, y) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[l(f(g(x)), y)] \tag{1}$$

where the $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are parameters of function f and g respectively, and l is a classification loss such as cross-entropy. The adversarial loss [13] can be written as follows:

$$\mathcal{L}_{\text{U}}(\boldsymbol{\phi}|x) = \mathbf{d}(p_{x\sim\mathcal{D}^s}(g(x)), p_{x\sim\mathcal{D}^t}(g(x))) \tag{2}$$

where, $\mathbf{d}$ is a distance metric such as $\mathcal{H}$-divergence between the source and target domains. The total loss is:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}|x, y) = \mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{\phi}|x, y) + \lambda\mathcal{L}_{\text{U}}(\boldsymbol{\phi}|x) \tag{3}$$

The adversarial loss, $\mathcal{L}_{\text{U}}$ is a metric between two distributions, and needs to be instantiated. The standard approach as proposed by Ganin et al. [13] is to deal with the minimization of the $\mathcal{H}$-divergence by maximizing the discriminator loss of a binary classifier that tries to separate the two domains:

$$p(d|x) = \frac{\exp(\mathbf{u}_d^\top \mathbf{g}(\mathbf{x}))}{\sum_{d'} \exp(\mathbf{u}_{d'}^\top \mathbf{g}(\mathbf{x}))} \tag{4}$$

where $d \in \{s, t\}$ is the domain class label and $\mathbf{u}$ consists of the softmax weights. Given that there are only two domains, we could

have avoided the softmax and simply used a sigmoid to output the probabilities of belonging to the two domains. The domain adaptation loss can now be written as the following adversarial loss:

$$\mathcal{L}_{\mathrm{U}} = \mathcal{L}_{\mathrm{DI}}(\boldsymbol{\phi}, \mathbf{u}|x) = \sum_{d \in \{s,t\}} \mathbb{I}(D = d) \log p(D = d|x) \tag{5}$$

Note, there is no negative sign in front of the $\mathcal{L}_{\mathrm{DI}}$ loss since we want to play a min-max game, minimizing the classification loss $\mathcal{L}_c$ (label decoder loss) and maximizing the discriminator loss $\mathcal{L}_{\mathrm{DI}}$. This ensures that the encoder $g(\cdot)$ transforms the source and target domains into a space that contains no domain information, hence making it easier for the classifier to predict the labels trained on the joint corpus. The total min-max loss optimized by DANN is:

$$\mathcal{L}_{\mathrm{DANN}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{u}|x, y) = \mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{\phi}|x, y) + \lambda \mathcal{L}_{\mathrm{DI}}(\boldsymbol{\phi}, \mathbf{u}|x) \tag{6}$$

Note that we are writing loss terms only for a given example and the total loss would be summation of losses over all examples, in addition to the regularization terms on the weights. We will use a similar style of formulation for rest of the paper.

4.2 Supervised Domain Adaptation

Supervised domain adaptation [28] is a way of using prediction task labels in the domain adaptation process. Similar to the unsupervised case above, supervised loss consists of a classification loss and an adversarial domain adaptation loss - the difference being that the adversarial loss uses the task label information. Since prospecting data does not cover $p = P_m$, we can write the adversarial loss as:

$$\mathcal{L}_{\mathrm{S}}(\boldsymbol{\phi}|x, y) = \sum_{y=1}^{K} \mathbf{d}(p^{y}_{x \sim \mathcal{D}^s_{\{p \neq P_m\}}}(g(x)), p^{y}_{x \sim \mathcal{D}^t_{\{p \neq P_m\}}}(g(x))) \tag{7}$$

where, K is the number of classes for the supervised learning task - two in our case - click or not-click. Motiian et al. [27, 28] incorporate class dissimilarity along with the above class similarity by adding the following loss with the pairs (i, j) of different classes from source domain and target domain:

$$\mathcal{L}_{\mathrm{D}}(\boldsymbol{\phi}|x, y) = \sum_{i,j \in K, i \neq j} \mathbf{s}(p^{i}_{x \sim \mathcal{D}^s_{\{p \neq P_m\}}}(g(x)), p^{j}_{x \sim \mathcal{D}^t_{\{p \neq P_m\}}}(g(x))) \tag{8}$$

where s is a similarity based loss, and aim is to minimize the similarity between the source sample from class i and target sample from class j. Canonical Contrastive Supervised Alignment (**SDA-CCSA**) method by Motiian et al. [28] instantiates the distance and similarity based loss by:

$$d(x_i, x_j) = \frac{1}{2} ||g(x_i) - g(x_j)||^2 \tag{9}$$

$$s(x_i, x_j) = \frac{1}{2} max(0, 1 - d(x_i, x_j))^2 \tag{10}$$

where, $d(x_i, x_j)$ is the normalized distance between the embedding of $x_i \in \mathcal{D}^s, x_j \in \mathcal{D}^t$. The above loss essentially takes the form of constrastive loss [16], and we refer to this method as SDA-CCSA. With ρ being the set of paired examples (x_i, x_j, y_i, y_j), total loss becomes:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}|x, y) = \mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{\phi}|x, y) + \lambda_1 \mathcal{L}_{\mathrm{S}}(\boldsymbol{\phi}|\rho) + \lambda_2 \mathcal{L}_{\mathrm{D}}(\boldsymbol{\phi}|\rho) \tag{11}$$

5 PROPOSED METHODOLOGY

In the previous section, we gave an overview of the domain adaptation techniques and described two popular techniques in unsupervised and supervised domain adaptation. In this section, we describe our proposed methods: Supervised Sub-Domain Adaptation-DANN (SDA-DANN) and Supervised Domain Adaptation based on Ranking (SDA-Ranking) for our problem.

5.1 Supervised Sub-Domain Adaptation-DANN

SDA-CCSA described previously, uses a pairwise formulation to minimize the adversarial loss in Eq. (7). However, this formulation is not attractive given the computation complexity involved in pairwise training. Instead, we can use a domain invariance loss based adversarial learning. This is different from DANN, since we perform domain invariance in a class-wise fashion to leverage the prediction label information. For class-wise based adversarial loss, we can formulate the discriminator loss as follows.

$$p(d|Y, x) = \frac{\exp(\mathbf{u}_d^\top \mathbf{g}(\mathbf{x}))}{\sum_{d'} \exp(\mathbf{u}_{d'}^\top \mathbf{g}(\mathbf{x}))} \tag{12}$$

$$\mathcal{L}_S(\boldsymbol{\phi}, \mathbf{u}|x, y) = \sum_{Y=1}^{K} \mathbb{I}(Y = y) \sum_{d \in \{s,t\}} \mathbb{I}(D = d) \log p(D = d|Y = y, x) \tag{13}$$

where $d \in \{s, t\}$ is domain class and $Y \in \{Click, noClick\}$ is class label.

Sub-domain Level Information: Adding Partner Level Affinities. The preceding formulation treats every existing partner equally for cold-starting a new partner. However, in practice partners deal with a range of products that can be wildly different from one another. Hence, when developing a model for the cold-start partner, partners dealing with similar products might be more relevant. We can incorporate the partner *affinities* by the products they are dealing with. In order to do so, we propose to use a weighing factor α_{P_m, P_i} derived from the similarity of catalogues of partner products. For example, a partner who deals with electronics will be more relevant for on-boarding a cold-start partner that deals in the electronics domain, rather than a partner that deals in apparels. Thus the class-wise loss from Eq. (13) can be modified as:

$$\mathcal{L}_{S'}(\boldsymbol{\phi}, \mathbf{u}|x, y) = \alpha_{P_m, P_i} \mathcal{L}_S(\boldsymbol{\phi}, \mathbf{u}|x, y) \tag{14}$$

where, P_m is the partner being cold-started and example $x \in P_i$.

To calculate α_{P_m, P_i} we can use the product categories partners deal with. We have a vector of category weights for a partner P_i given by $\mathbf{c_i}$ where $\mathbf{c_i}$ is a vector consisting of weights for each of the categories. The catalogue contains the counts of each category the partner deals with. The similarity between the partners can be defined as cosine similarity $\alpha_{P_m, P_i} = \mathrm{cosine\_sim}(\mathbf{c_m}, \mathbf{c_i})$. Hence, the total loss for the method we refer to as Supervised Sub-Domain Adaptation-DANN (SDA-DANN) is:

$$\mathcal{L}_{\mathrm{SDA-DANN}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{u}|x, y) = \mathcal{L}_{\mathrm{c}}(\boldsymbol{\theta}, \boldsymbol{\phi}|x, y) + \lambda \mathcal{L}_{S'}(\boldsymbol{\phi}, \mathbf{u}|x, y) \tag{15}$$

5.2 Ranking Loss Based Supervised Domain Adaptation

The goal of the supervised domain adaptation methods is to align the source and target domains together. The loss functions used by

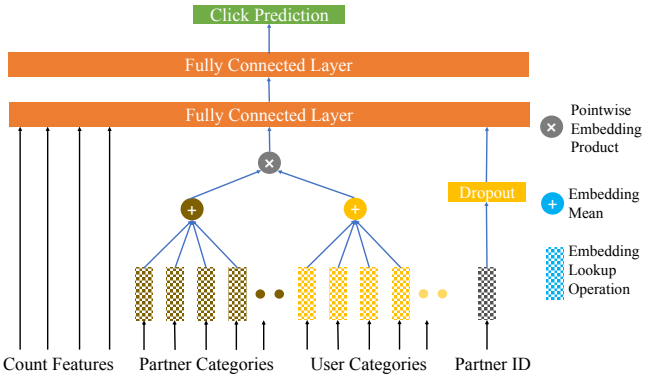


Figure 2: Proposed architecture of Deep Neural Network

SDA-CCSA [28] imposes class wise similarity and dissimilarities during the domain alignment phase. In essence, these methods try to separate the different classes from each other while aligning the same classes together in source-target domains. It imposes stringent constraints like $l2$ distance between the embedding of source-target pairs while giving different treatment to pairwise similarity and pairwise dissimilarity. This in spirit is similar to the triplet loss or ranking loss. In our problem, we only have two classes: clicks and no-clicks. An alternate to the method proposed in Section 5.1, is to directly use a ranking loss between the source-target pairs, with clicks (both source and target) ranked higher than the non-clicks in the shared deep network structure.

Let the source and target pair be (x_i, x_j) with $y_i \neq y_j$, where x_i and x_j can be from either domains. The encoder output from the source and target pair, (x_i, x_j) is respectively $g(x_i)$ and $g(x_j)$. The source target pair ranking loss $\mathcal{L}_R$ can be given by:

$$\Delta_{ij} = g(x_i) - g(x_j) \tag{16}$$

$$\sigma(\Delta_{ij}) = \frac{1}{1 + e^{-\mathbf{w}^\top \Delta_{ij}}} \tag{17}$$

$$\mathcal{L}_R(w|\rho) = -\sum_{(i,j)\in\rho} \mathbb{I}_{ij} log\sigma(\Delta_{ij}) + (1 - \mathbb{I}_{ij}) log(1 - \sigma(\Delta_{ij})) \tag{18}$$

where, $\mathbb{I}_{ij}$ is the indicator function with $\mathbb{I}_{ij} = 1$ if y_i, y_j is a (+,-) pair and $\mathbb{I}_{ij} = 0$ for a (-,+) pair, ρ is the set of paired examples (x_i, x_j, y_i, y_j), and $\mathbf{w}$ is set of weights. Using this ranking loss, we promote the intra-class alignment, along with the inter-class separation in a single loss function.

Apart from the ranking loss, we also use the classification loss $\mathcal{L}_c$ given by Eq. (1) since it helped improve the performance. Hence, the total loss for the method we refer to as Supervised Domain Adaptation-Ranking (SDA-Ranking) is:

$$\mathcal{L}_{\text{SDA-Ranking}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{w}|\rho) = \mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{\phi}|\rho) + \lambda_r \mathcal{L}_R(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{w}|\rho) \tag{19}$$

where λ_r is the weight for domain adaptation using ranking loss.

Note: In terms of training computation complexity, the baseline method by Dalessandro et al. [11] is the simplest, followed by DANN and SDA-DANN, since they deploy adversarial discriminator that takes more time to train the model. Pairwise ranking methods, SDA-Ranking and SDA-CCSA take more training time in that order owing to pairwise training regime. However, at inference time, all the methods are identical since we use the same base classifier (deep neural network) for the CTR prediction on prospecting domain.

Table 2: Description of partners in our test dataset.

Partner	Category	#CLicks	#NoClicks	#Total	CTR (%)
1	Home and Garden/Misc	3604	127985	131589	2.74
2	Apparel	4661	79219	83880	5.56
3	Electronics	17812	154313	172125	10.35
4	Apparel	8716	711077	719793	1.21
5	Home and Beauty	33638	409040	442678	7.60
6	Apparel	1795	50960	52755	3.40
7	Apparel	1294	54443	55737	2.32
8	Apparel	1923	94006	95929	2.00
9	Home and Beauty	4773	110534	115307	4.14
10	Apparel	4295	79155	83450	5.15
11	Electronics	3367	102336	105703	3.19
12	Apparel	2270	48648	50918	4.46
13	Home and Beauty	936	66580	67516	1.39
14	Home and Garden/Misc	3362	128801	132163	2.54
15	Home and Garden/Misc	3104	102063	105167	2.95

6 EXPERIMENTAL SETTINGS

In this section, we describe our experimental setup: data used, deep neural network architecture, metrics used, baselines, and hyper-parameter tuning.

6.1 Data

We use data from Criteo, a major digital advertiser. We have three sets of input features: user count features summarizing the display and click statistics of the user, partner features (partner categories and ID), and user categories along with click or no-click information about the display made. We use a universal set of categories from Google[2]. Even though there are thousands of categories, we take top-125 categories in our data for reduced complexity. We use a day of data from January 2019 as our training set and use the following day of data as our test set for the partner cold-start problem. Table 2 shows the statistics of the prospecting test data for each of the partners. Our training data consists of 352,081 samples for prospecting and 771,194 samples for re-targeting across 15 partners.

6.2 Deep Neural Network Architecture

Figure 2 shows the architecture of our deep neural network. To handle the category data, we represent the categories using an embedding matrix $\mathbf{E_c} \in \mathbb{R}^{125xd}$, where d is the dimension of the representation and 125 is the number of categories. Embedding matrix $\mathbf{E_c}$ is initialized with random values and updated during the training to represent each of the 125 categories. Since each partner (user) can have multiple categories, we take a mean of the embeddings of partner categories given by $\mathbf{e}_p$ (or user categories $\mathbf{e}_u$). To denote the compatibility of the users and partner, we take a point-wise product of the two to get a user-partner embedding.

$$\mathbf{e}_{up} = \mathbf{e}_p \odot \mathbf{e}_u \tag{20}$$

Next, we concatenate the other input features: count features (x_{count}) and partner embedding (z_p) with $\mathbf{e}_{up}$ as $a = (x_{count}; z_p; \mathbf{e}_{up})$. The concatenated vector, a is then passed on to a fully connected

[2]https://www.google.com/basepages/producttype/taxonomy.en-US.txt

layered network that finally predicts whether the user would click on the partner advertisement or not. We keep this base neural network architecture the same across our experiments since we want to compare their performance on the domain adaptation task.

6.3 Metrics

We use the following metrics for evaluating our methods:

- **AUC-ROC**: We calculate Area under the Curve (AUC) for the Receiver Operator Curve (ROC) for the cold-start experiments. We refer to AUC-ROC as AUC for brevity throughout this paper. For a classifier $f : x \rightarrow y$, AUC is essentially a ranking metric that can be defined as:

$$\text{AUC} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbb{I}(f(x_i^+) > f(x_j^-)) \tag{21}$$

  where, the m is the number of positive (clicks) examples, and n is the number of negative (non-clicks) examples. Hence, this metric measures how well the positive instances are ranked higher than the negative instances.
- **Mean-AUC**: The mean-AUC metric is defined as the mean of AUCs across all partners for the CTR prediction task.
- **Median-AUC**: Since the previous metric is prone to be influenced by the outliers, we also report the median-AUC for better representation of the AUC values. We use median-AUC as the primary metric for evaluating our models.
- **Precision@K**: Precision @K metric represents the ratio of true positives in the predicted top-K positives. While it has not been used conventionally in the CTR literature, this is an important metric to consider than plain AUC. The difference lies in the fact that AUC performs a micro-averaging across all users, whereas Precision@K targets top-K users. Usually, a partner would come to the digital advertiser with a predefined budget or number of users a partner wants to target. A model would return the top-K users inferred are the most likely ones to click.

6.4 Baselines

We define the following baselines used to compare with:

- **Prospecting**: We use only prospecting data without the re-targeting data.
- **Direct transfer (Baseline)**: We use Dalessandro et al. [11]'s method as our primary baseline. From here on, direct transfer is referred to as **baseline**, interchangeably. We follow their re-training approach on target data using the neural network trained on the source data.
- **DANN**: We use DANN [13] as described in Section 4.1.
- **SDA-CCSA**: We use SDA-CCSA [28] as described in Section 4.2.

6.5 Hyper-parameter Tuning

We use 80% data for training and 20% for validation across our experiments. For cold-start experiments, we remove the prospecting data of the partner we want to cold-start during the training phase. We use early stopping criteria for model selection, *i.e.,* we stop the model training if there is no improvement in the AUC values for last 25 epochs. The best hyper-parameter combination is selected based on the performance on the validation set. For the deep neural network, we use a fully connected layer with k hidden units, selected from the set of $\{10, 20, 30, 40, 50, 60, 70, 80\}$. For the models based on DANN, we use the same setup of the gradient coefficient schema for λ in Eq. (6) as the Ganin et al. [13]. For the SDA-CCSA, we use $\lambda_1 = 0.25$ and $\lambda_2 = 0.75$ in Eq. (11).

7 EXPERIMENTS AND RESULTS

In this section, we present the findings from our experiments. First, we present the results on CTR prediction task for the partner cold-start problem using domain adaptation methods. Next, we present the results on CTR prediction task for the partner steady state problem using domain adaptation methods. Lastly, we show three ablation studies.

7.1 Cold-Start Results

In this section, we evaluate the performance of our domain adaptation methods in the cold-start setting.

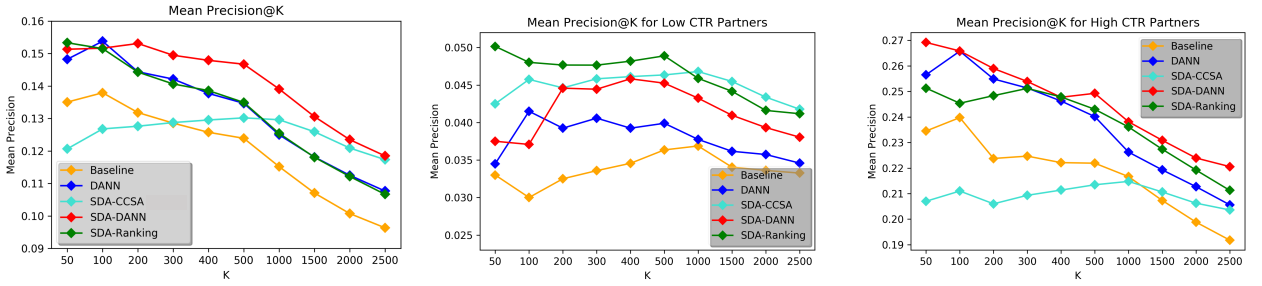
Median-AUC. Table 3 shows the results for our methods and baselines. The proposed SDA-DANN method performs better than all other methods, improving over baseline by 4.69% in median-AUC. This increased performance can be attributed to the addition of class-wise domain in-variance and partner affinities in the training regime. SDA-Ranking method gives a significant improvement over the baseline methods.

DANN and SDA-CCSA methods perform better as compared to the direct transfer baseline method. The direct transfer baseline method merely adapts the classifier trained on the source domain to the target domain. The domain adaptation methods, on the other hand, align the two domains by projecting them into a shared space. Hence, the domain adaptation classifiers can exploit more extensive data available in the source domain more effectively during the training procedure compared to the baseline.

The supervised domain adaptation methods have been shown to perform better [28] than unsupervised domain adaptation methods as clearly demonstrated in our experiments. As evident, modifying DANN to a supervised setting improves the performance over the baseline DANN by 2.37% for median-AUC. *Over 4% improvement in AUC numbers is quite significant in the advertising domain given the advertising revenues ranging in billions of dollars.*

Table 3: Results for the cold-start scenario: mean values of mean-AUC and median-AUC across each of 15 partners over 10 runs. $\neg P_m$ refers to the set of partners except the partner for which perform the cold-start. P is the set of all partners. % Improv is the relative improvement of the methods with respect the baseline direct transfer method's median-AUC.

Method	Data-set	Mean AUC	Median AUC	% Improv.
Prospecting	Pros($\neg P_m$)	0.573	0.568	-1.04%
Direct Transfer	Pros($\neg P_m$)+Ret(P)	0.574	0.576	0.00%
DANN	Pros($\neg P_m$)+Ret(P)	0.593	0.589	+2.26%
SDA-CCSA	Pros($\neg P_m$)+Ret(P)	0.601	0.589	+2.26%
SDA-DANN	Pros($\neg P_m$)+Ret(P)	**0.606**	**0.603**	**+4.69%**
SDA-Ranking	Pros($\neg P_m$)+Ret(P)	0.592	0.593	+2.95%



(a) Precision@K for all partners

(b) Mean Precision@K for low CTR partners

(c) Mean Precision@K for high CTR partners

Figure 3: Precision@K metric for $K = \{50, 100, 200, 300, 400, 500, 1000, 1500, 2000, 2500\}$ for all the partners (left), low CTR partners (center), and high CTR partners (right).

Precision@K. Precision@K measures how likely the top-K users represents the ratio of true positives in the predicted top-K positives. This is in contrast to the AUC metric that performs a micro-averaging over all users and not top K users only. The range of K depends on the respective partner. We use $K \in \{50, 100, 200, 300, 400, 500, 1000, 1500, 2000, 2500\}$ in our experiments.

Figure 3a shows the mean precision@K, averaged over the 15 partners. As evident, SDA-DANN outperforms all the other methods. However, we observe that the mean plot of performance across partners can be misleading. We further analyze the partners based on their CTRs to illustrate this point.

Fig 3b shows the mean precision@K for low CTR (CTR $< 2.5\%$) partners. As illustrated, we observe that ranking methods, *i.e.,* SDA-Ranking and SDA-CCSA perform better for lower CTR partners. This observation can be explained by the fact that ranking based methods use a pairwise loss to create a differentiation between clicks and non-clicks. This allows for the creation of more contrasting pairs, pushing the top-ranked clicks (+ve's) relatively higher than non-clicks (-ve's) while classification based methods make no such effort.

Fig 3c shows the mean precision@K for high CTR (CTR $> 5\%$) partners. As evident, SDA-DANN outperforms all the other methods. This can be attributed to the sub-domain information, *i.e.,* partner affinities being used by the SDA-DANN method, that shows its superiority over other methods.

Table 4: Results for the steady state scenario: mean values of mean-AUC and median-AUC across each of 15 partners over 10 runs. P is the set of all partners. % Improv is the relative improvement of the methods with respect the baseline direct transfer method's median-AUC.

Method	Data-set	Mean AUC	Median AUC	% Improv.
Prospecting	Pros(P)	0.588	0.585	+1.03%
Direct Transfer	Pros(P)+Ret(P)	0.577	0.579	0.00%
DANN	Pros(P)+Ret(P)	0.595	0.589	+1.73%
SDA-CCSA	Pros(P)+Ret(P)	0.604	0.591	+2.07%
SDA-DANN	Pros(P)+Ret(P)	**0.605**	**0.603**	**+4.14%**
SDA-Ranking	Pros(P)+Ret(P)	0.591	0.592	+2.24%

7.2 Steady State Results

Cold-start was defined as the scenario when we have data about other partners available in the source domain as well as the target domain. Further, the data for the partner of interest was only available in the source domain. In contrast, steady-state is defined as the scenario when not only we have data about other partners available in source and target domain, but also the data for the partner of interest is available in both source and target domains. Table 4 shows the AUC metrics for the CTR prediction task in prospecting domain during steady state.

As depicted, when we add the partner's prospecting data, the performance of the Prospecting baseline model improves substantially to 0.585 from 0.568 from the cold-start results (Table 3). This can be attributed to the addition of extra information from the partner of interest's target domain samples.

For the other transfer learning methods, adding the partner of interest's target domain samples does not make a noticeable difference in performance compared to cold-start. This can be attributed to the fact that prospecting is predominantly about transferring re-targeting users across partners. We can infer that there is already enough knowledge transfer from similar cross-partner data from the re-targeting domain.

From these results, we can conclude that the *major gains for domain adaptation methods come from additional re-targeting data* containing the samples for the partner being cold-started on the prospecting platform.

7.3 Ablation Studies on Cold-Start

In this section, we will describe three ablation studies performed during the cold-start scenario.

7.3.1 Effect of varying amount of target domain data. In order to motivate the applicability of domain adaptation, we analyze the effect of the amount of target domain (prospecting) data on the partner of interest's cold-start performance. We train the prospecting baseline model with varying proportions of target domain data. As can be seen in Figure 4, as we increase the amount of data, there is an increase in performance. However, given limited target domain data (100% in Figure 4), further, improvement is only possible if we bring additional data from re-targeting. Note that in re-targeting, not only do we have the availability of more data, but also data for

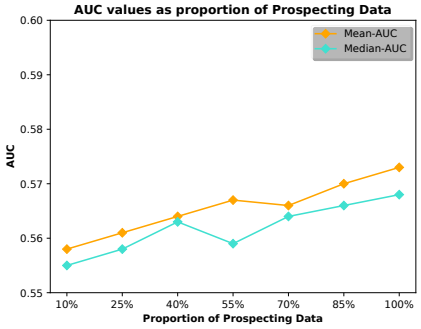


Figure 4: AUC metrics on cold-start using increasing amount of prospecting data. As we can observe adding more data helps increase the performance.

the partner being cold-started. This *motivates our intuition for using a domain adaptation approach* by utilizing re-targeting data for the prospecting task.

7.3.2 Effect of varying amount of source domain data. Figure 5 shows the performance as we change the amount of the cold-start partner P_m's samples from the source (re-targeting) domain. There is no noticeable change in the performance of our methods as we increase the amount of source data of the cold-start partner. This can be attributed to the fact that prospecting is all about transferring re-targeting users from other partners to the partner of interest. Hence, the cross-partner data, especially from the re-targeting domain becomes very useful. In conclusion, cross-partner data is more important as algorithms are able to extract signals that are required for the prospecting cold-start prediction task.

7.3.3 Effect of Partner Embedding Dropout. We study the effect of using dropout on partner embedding (see Fig 2) as proposed by Volkovs et al. [42]. They demonstrate that using dropout on the user or item embedding layer helps in traditional cold-start. Figure 6 shows the effect of dropout on the baseline Prospecting only model. As observed, there is not much difference in the performance of the model as we increase dropout. Availability of more informative features explains the non-applicability of Volkovs et al.'s dropout method in our scenario. The partner level information

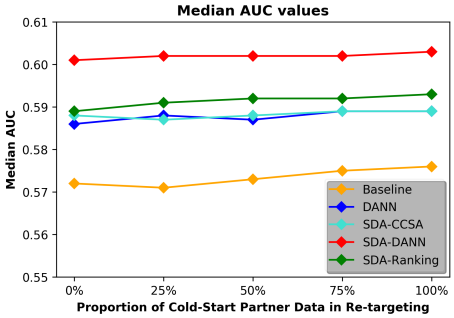


Figure 5: Cold-start results as a function of amount of re-targeting data for the cold-started partner.

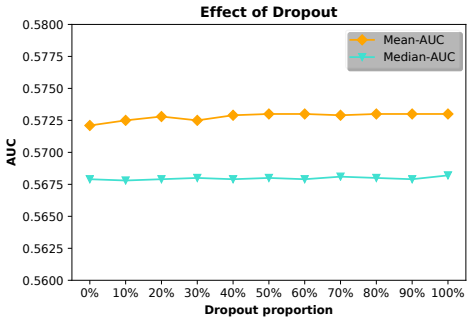


Figure 6: AUC metrics on cold-start for the Prospecting method Pros($\neg P_m$). Dropout does not exhibit an appreciable effect.

can be obtained from category information as a part of partner category embeddings.

8 CONCLUSIONS AND FUTURE WORK

We present the first study on partner cold-start problem, with an application on user prospecting platform using re-targeting data. Towards that, we propose two domain adaptation techniques to overcome the challenges associated with partner cold-start. In particular, we propose the first set of methods to extend domain adaptation for partner cold-start by incorporating sub-domain similarities (product category level information), also referred to as SDA-DANN. In particular, SDA-DANN leverages the similarities across partners and performs the domain alignment in a class-wise fashion, in contrast to the original DANN method. We also proposed a supervised domain adaptation approach termed as SDA-Ranking by leveraging a ranking loss for the domain adaptation task.

We then demonstrated the effectiveness of our proposed approach on a real-world data-set obtained from Criteo, a major digital advertisement company. In our experiments, we illustrate the superior performance of our proposed methods as compared to the state-of-art baselines. Our experiments reveal that SDA-DANN performs the best among all the methods, whereas SDA-Ranking works better for the low-CTR partners.

In the future, we would like to deploy these models in the online setting to measure their performance in deployment scenarios. Another important aspect we did not consider was the temporal nature of the displays since the users view these advertisements in a sequence. This sequential aspect would be an exciting research avenue to explore with domain adaptation techniques.

ACKNOWLEDGMENTS

We would like to thank Matthieu Kirchmeyer and Amit Goyal for the insightful discussions, and Zhengming Xing for the help with computing resources.

REFERENCES

[1] Agarwal, D., Agrawal, R., Khanna, R., and Kota, N. Estimating rates of rare events with multiple hierarchies through scalable log-linear models. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (2010), ACM, pp. 213–222.

[2] Azarbonyad, H., Sim, R., and White, R. W. Domain adaptation for commitment detection in email. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (2019), ACM, pp. 672–680.

[3] Barjasteh, I., Forsati, R., Masrour, F., Esfahanian, A.-H., and Radha, H. Cold-start item and user recommendation with decoupled completion and transduction. In *Proceedings of the 9th ACM Conference on Recommender Systems* (2015), ACM, pp. 91–98.

[4] Bickel, S., Sawade, C., and Scheffer, T. Transfer learning by distribution matching for targeted advertising. In *Advances in neural information processing systems* (2009), pp. 145–152.

[5] Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., and Erhan, D. Domain separation networks. In *Advances in Neural Information Processing Systems* (2016), pp. 343–351.

[6] Bureau, I. A. IAB internet advertising revenue report. https://www.iab.com/wp-content/uploads/2018/11/REPORT-IAB-Internet-Advertising-Revenue-Report-HY-2018.pdf, 2018. [Online; accessed 27-March-2018].

[7] Chapelle, O., Manavoglu, E., and Rosales, R. Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology (TIST) 5*, 4 (2015), 61.

[8] Chen, J., Sun, B., Li, H., Lu, H., and Hua, X.-S. Deep ctr prediction in display advertising. In *Proceedings of the 2016 ACM on Multimedia Conference* (2016), ACM, pp. 811–820.

[9] Cheng, H., and Cantú-Paz, E. Personalized click prediction in sponsored search. In *Proceedings of the third ACM international conference on Web search and data mining* (2010), ACM, pp. 351–360.

[10] Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* (2016), ACM, pp. 7–10.

[11] Dalessandro, B., Chen, D., Raeder, T., Perlich, C., Han Williams, M., and Provost, F. Scalable hands-free transfer learning for online advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (2014), ACM, pp. 1573–1582.

[12] Edizel, B., Mantrach, A., and Bai, X. Deep character-level click-through rate prediction for sponsored search. *arXiv preprint arXiv:1707.02158* (2017).

[13] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research 17*, 1 (2016), 2096–2030.

[14] Goldfarb, A., and Tucker, C. Online display advertising: Targeting and obtrusiveness. *Marketing Science 30*, 3 (2011), 389–404.

[15] Guo, H., Tang, R., Ye, Y., Li, Z., and He, X. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247* (2017).

[16] Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (2006), vol. 2, IEEE, pp. 1735–1742.

[17] He, M., Zhang, J., Yang, P., and Yao, K. Robust transfer learning for cross-domain collaborative filtering using multiple rating patterns approximation. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (2018), ACM, pp. 225–233.

[18] He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising* (2014), ACM, pp. 1–9.

[19] Hillard, D., Schroedl, S., Manavoglu, E., Raghavan, H., and Leggetter, C. Improving ad relevance in sponsored search. In *Proceedings of the third ACM international conference on Web search and data mining* (2010), ACM, pp. 361–370.

[20] Juan, Y., Zhuang, Y., Chin, W.-S., and Lin, C.-J. Field-aware factorization machines for ctr prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems* (2016), ACM, pp. 43–50.

[21] Khan, M. M., Ibrahim, R., and Ghani, I. Cross domain recommender systems: a systematic literature review. *ACM Computing Surveys (CSUR) 50*, 3 (2017), 36.

[22] King, M., Atkins, J., and Schwarz, M. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *The American economic review 97*, 1 (2007), 242–259.

[23] Lambrecht, A., and Tucker, C. When does retargeting work? information specificity in online advertising. *Journal of Marketing Research 50*, 5 (2013), 561–576.

[24] Liu, W., Tang, R., Li, J., Yu, J., Guo, H., He, X., and Zhang, S. Field-aware probabilistic embedding neural network for ctr prediction. In *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), ACM, pp. 412–416.

[25] McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (2013), ACM, pp. 1222–1230.

[26] Moreno, O., Shapira, B., Rokach, L., and Shani, G. Talmud: transfer learning for multiple domains. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (2012), ACM, pp. 425–434.

[27] Motiian, S., Jones, Q., Iranmanesh, S., and Doretto, G. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems* (2017), pp. 6670–6680.

[28] Motiian, S., Piccirilli, M., Adjeroh, D. A., and Doretto, G. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 5715–5725.

[29] Ni, Y., Ou, D., Liu, S., Li, X., Ou, W., Zeng, A., and Si, L. Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks. *arXiv preprint arXiv:1805.10727* (2018).

[30] Oldridge, E. Adapting session based recommendation for features through transfer learning. In *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), ACM, pp. 481–481.

[31] Pan, S. J., and Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering 22*, 10 (2010), 1345–1359.

[32] Pan, W., Xiang, E. W., Liu, N. N., and Yang, Q. Transfer learning in collaborative filtering for sparsity reduction. In *Twenty-fourth AAAI conference on artificial intelligence* (2010).

[33] Pandey, G., Kotkov, D., and Semenov, A. Recommending serendipitous items using transfer learning. In *Proceedings of the 27th ACM international conference on information and knowledge management* (2018), ACM, pp. 1771–1774.

[34] Perlich, C., Dalessandro, B., Raeder, T., Stitelman, O., and Provost, F. Machine learning for targeted display advertising: Transfer learning in action. *Machine learning 95*, 1 (2014), 103–127.

[35] Regelson, M., and Fain, D. Predicting click-through rate using keyword clusters. In *Proceedings of the Second Workshop on Sponsored Search Auctions* (2006), vol. 9623.

[36] Richardson, M., Dominowska, E., and Ragno, R. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 521–530.

[37] Ruder, S. *Neural Transfer Learning for Natural Language Processing*. PhD thesis, National University of Ireland, Galway, 2019.

[38] Sahebi, S., and Brusilovsky, P. Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In *International Conference on User Modeling, Adaptation, and Personalization* (2013), Springer, pp. 289–295.

[39] Saveski, M., and Mantrach, A. Item cold-start recommendations: learning local collective embeddings. In *Proceedings of the 8th ACM Conference on Recommender systems* (2014), ACM, pp. 89–96.

[40] Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (2002), ACM, pp. 253–260.

[41] Su, Y., Jin, Z., Chen, Y., Sun, X., Yang, Y., Qiao, F., Xia, F., and Xu, W. Improving click-through rate prediction accuracy in online advertising by transfer learning. In *Proceedings of the International Conference on Web Intelligence* (2017), ACM, pp. 1018–1025.

[42] Volkovs, M., Yu, G., and Poutanen, T. Dropoutnet: Addressing cold start in recommender systems. In *Advances in Neural Information Processing Systems* (2017), pp. 4957–4966.

[43] Zeff, R. L., and Aronson, B. *Advertising on the Internet*. John Wiley & Sons, Inc., 1999.

[44] Zhai, S., Chang, K.-h., Zhang, R., and Zhang, Z. M. Deepintent: Learning attentions for online advertising with recurrent neural networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (2016), ACM, pp. 1295–1304.

[45] Zhang, W., Du, T., and Wang, J. Deep learning over multi-field categorical data. In *European conference on information retrieval* (2016), Springer, pp. 45–57.

[46] Zhang, Y., Dai, H., Xu, C., Feng, J., Wang, T., Bian, J., Wang, B., and Liu, T.-Y. Sequential click prediction for sponsored search with recurrent neural networks. In *AAAI* (2014), vol. 14, pp. 1369–1375.

[47] Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., and Gai, K. Deep interest evolution network for click-through rate prediction. *arXiv preprint arXiv:1809.03672* (2018).

[48] Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., and Gai, K. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018), ACM, pp. 1059–1068.