Implicit Look-Alike Modelling in Display Ads

Transfer Collaborative Filtering to CTR Estimation

Weinan Zhang[1,2](✉), Lingxi Chen[1], and Jun Wang[1,2]

[1] University College London, London, UK
{w.zhang,lingxi.chen,j.wang}@cs.ucl.ac.uk
[2] MediaGamma Limited, London, UK

Abstract. User behaviour targeting is essential in online advertising. Compared with sponsored search keyword targeting and contextual advertising page content targeting, user behaviour targeting builds users' interest profiles via tracking their online behaviour and then delivers the relevant ads according to each user's interest, which leads to higher targeting accuracy and thus more improved advertising performance. The current user profiling methods include building keywords and topic tags or mapping users onto a hierarchical taxonomy. However, to our knowledge, there is no previous work that explicitly investigates the user online visits similarity and incorporates such similarity into their ad response prediction. In this work, we propose a general framework which learns the user profiles based on their online browsing behaviour, and transfers the learned knowledge onto prediction of their ad response. Technically, we propose a transfer learning model based on the probabilistic latent factor graphic models, where the users' ad response profiles are generated from their online browsing profiles. The large-scale experiments based on real-world data demonstrate significant improvement of our solution over some strong baselines.

1 Introduction

Targeting technologies have been widely adopted in various online advertising paradigms during the recent decade. According to the Internet advertising revenue report from IAB in 2014 [22], 51 % online advertising budget is spent on sponsored search (search keywords targeting) and contextual advertising (page content targeting), while 39 % is spent on display advertising (user demographics and behaviour targeting), and the left 10 % is spent on other ad formats like classifieds. With the rise of ad exchanges [19] and mobile advertising, user behaviour targeting has now become essential in online advertising.

Compared with sponsored search or contextual advertising, user behaviour targeting *explicitly* builds the user profiles and detects their interest segments via tracking their online behaviour, such as browsing history, search keywords and ad clicks etc. Based on user profiles, the advertisers can detect the users with similar interests to the known customers and then deliver the relevant ads to them. Such technology is referred as *look-alike modelling* [17], which efficiently provides

N. Ferro et al. (Eds.): ECIR 2016, LNCS 9626, pp. 589–601, 2016.
DOI: 10.1007/978-3-319-30671-1_43

higher targeting accuracy and thus brings more customers to the advertisers [29]. The current user profiling methods include building keyword and topic distributions [1] or clustering users onto a (hierarchical) taxonomy [29]. Normally, these inferred user interest segments are then used as target restriction rules or as features leveraged in predicting users' ad response [32].

However, the two-stage profiling-and-targeting mechanism is not optimal (despite its advantages of explainability). First, there is no flexible relationship between the inferred tags or categories. Two potentially correlated interest segments are regarded as separated and independent ones. For example, the users who like cars tend to love sports as well, but these two segments are totally separated in the user targeting system. Second, the first stage, i.e., the user interest segments building, is performed independently and with little attention of its latter use of ad response prediction [7,29], which is suboptimal. Third, the effective tag system or taxonomy structure could evolve over time, which makes it much difficult to update them.

In this paper, we propose a novel framework to *implicitly* and *jointly* learn the users' profiles on both the general web browsing behaviours and the ad response behaviours. Specifically, (i) Instead of building explicit and fixed tag system or taxonomy, we propose to directly map each user, webpage and ad into a latent space where the shape of the mapping is automatically learned. (ii) The users' profiles on general browsing and ad response behaviour are jointly learned based on the heterogeneous data from these two scenarios (or tasks). (iii) With a maximum a posteriori framework, the knowledge from the user browsing behaviour similarity can be naturally transferred to their ad response behaviour modelling, which in turn makes an improvement over the prediction of the users' ad response. For instance, our model could automatically discover that the users with the common behaviour on www.bbc.co.uk/sport will tend to click automobile ads. Due to its implicit nature, we call the proposed model *implicit look-alike modelling*.

Comprehensive experiments on a real-world large-scale dataset from a commercial display ad platform demonstrate the effectiveness of our proposed model and its superiority over other strong baselines. Additionally, with our model, it is straightforward to analyse the relationship between different features and which features are critical and cost-effective when performing transfer learning.

2 Related Work

Ad Response Prediction aims at predicting the probability that a specific user will respond (e.g., click) to an ad in a given context [4,18]. Such context can be either a search keyword [8], webpage content [2], or other kinds of real-time information related to the underlying user [31]. From the modelling perspective, many user response prediction solutions are based on linear models, such as logistic regression [14,24] and Bayesian probit regression [8]. Despite the advantage of high learning efficiency, these linear models suffer from the lack of feature interaction and combination [9]. Thus non-linear models such as tree models [9] and latent vector models [20,30] are proposed to catch the data non-linearity and interactions between features. Recently the authors in [12] proposed to first learn combination features from gradient boosting decision trees (GBDT) and, based on the tree leaves as features, learn a factorisation machine (FM) [23] to build feature interactions to improve ad click prediction performance.

Collaborative Filtering (CF) on the other hand is a technique for personalised recommendation [26]. Instead of exploring content features, it learns the user or/and item similarity based on their interactions. Besides the user(item)-based approaches [25,28], latent factor models, such as probabilistic latent semantic analysis [10], matrix factorisation [13] and factorisation machines [23], are widely used model-based approaches. The key idea of the latent factor models is to learn a low-dimensional vector representation of each user and item to catch the observed user-item interaction patterns. Such latent factors have good generalisation and can be leveraged to predict the users' preference on unobserved items [13]. In this paper, we explore latent models of collaborative filtering to model user browsing patterns and use them to infer users' ad click behaviour.

Transfer Learning deals with the learning problem where the learning data of the target task is expensive to get, or easily outdated, via transferring the knowledge learned from other tasks [21]. It has been proven to work on a variety of problems such as classification [6], regression [16] and collaborative filtering [15]. Different from multi-task learning, where the data from different tasks are assumed to drawn from the same distribution [27], transfer learning methods may allow for arbitrary source and target tasks. In online advertising field, the authors in a recent work [7] proposed a transfer learning scheme based on logistic regression prediction models, where the parameters of ad click prediction model were restricted with a regularisation term from the ones of user web browsing prediction model. In this paper, we consider it as one of the baselines.

3 Implicit Look-Alike Modelling

In performance-driven online advertising, we commonly have two types of observations about underlying user behaviours: one from their browsing behaviours (the interaction with webpages) and one from their ad responses, e.g., conversions or clicks, towards display ads (the interactions with the ads) [7]. There are two predictions tasks for understanding the users:

- **Web Browsing Prediction (CF Task).** Each user's online browsing behaviour is logged as a list containing previously visited publishers (domains or URLs). A common task of using the data is to leverage collaborative filtering (CF) [23,28] to infer the user's profile, which is then used to predict whether the user is interested in visiting any given new publisher. Formally, we denote the dataset for CF as D^{c} and an observation is denoted as $(\boldsymbol{x}^{\mathrm{c}}, y^{\mathrm{c}}) \in D^{\mathrm{c}}$, where $\boldsymbol{x}^{\mathrm{c}}$ is a feature vector containing the attributes from the user and the publisher and y^{c} is the binary label indicating whether the user visits the publisher or not.

- **Ad Response Prediction (CTR Task).** Each user's online ad feedback behaviour is logged as a list of pairs of ad impression events and their corresponding feedbacks (e.g., click or not). The task is to build a click-through rate (CTR) prediction model [5] to estimate how likely it is that the user will click a specific ad impression in the future. Each ad impression event consists of various information, such as user data (cookie ID, location, time, device, browser, OS etc.), publisher data (domain, URL, ad slot position etc.), and advertiser data (ad creative, creative size, campaign etc.). Mathematically, we denote the ad CTR dataset as D^{r} and its data instance as $(\boldsymbol{x}^{\mathrm{r}}, y^{\mathrm{r}})$, where $\boldsymbol{x}^{\mathrm{r}}$ is a feature vector and y^{r} is the binary label indicating whether the user clicks a given ad or not.

This paper focuses on the latter task: ad response prediction. We, however, observe that although they are different prediction tasks, the two tasks share a large proportion of users, publishers and their features. We can thus build a user-publisher interest model jointly from the two tasks. Typically we have a large number of observations about user browsing behaviours and we can use the knowledge learned from publisher CF recommendation to help infer display advertising CTR estimation.

3.1 The Joint Conditional Likelihood

In our solution, the prediction models on CF task and CTR task are learned jointly. Specifically, we build a joint data discrimination framework. We denote Θ as the parameter set of the joint model with prior $P(\Theta)$, and the *conditional* likelihood of an observed data instance is the probability of predicting the correct binary label given the features $P(y|\boldsymbol{x};\Theta)$. As such, the conditional likelihood of the two datasets are $\prod_{(\boldsymbol{x}^{\mathrm{c}},y^{\mathrm{c}})\in D^{\mathrm{c}}} P(y^{\mathrm{c}}|\boldsymbol{x}^{\mathrm{c}};\Theta)$ and $\prod_{(\boldsymbol{x}^{\mathrm{r}},y^{\mathrm{r}})\in D^{\mathrm{r}}} P(y^{\mathrm{r}}|\boldsymbol{x}^{\mathrm{r}};\Theta)$. Maximising a posteriori (MAP) estimation gives

$$\hat{\Theta} = \max_{\Theta} P(\Theta) \prod_{(\boldsymbol{x}^{\mathrm{c}},y^{\mathrm{c}})\in D^{\mathrm{c}}} P(y^{\mathrm{c}}|\boldsymbol{x}^{\mathrm{c}};\Theta) \prod_{(\boldsymbol{x}^{\mathrm{r}},y^{\mathrm{r}})\in D^{\mathrm{r}}} P(y^{\mathrm{r}}|\boldsymbol{x}^{\mathrm{r}};\Theta). \tag{1}$$

Just like most solutions on CF recommendation [10,13] and CTR estimation [14,24], in this discriminative framework, Θ is only concerned with the mapping from the features to the labels (the conditional probabilities) rather than modelling the prior distribution of features [11].

The details of the conditional likelihood $P(y^{\mathrm{c}}|\boldsymbol{x}^{\mathrm{c}};\Theta)$, $P(y^{\mathrm{r}}|\boldsymbol{x}^{\mathrm{r}};\Theta)$ and the parameter prior $P(\Theta)$ will be discussed in the latter subsections.

3.2 CF Prediction

For the CF task, we use a factorisation machine [23] as our prediction model. We further define the features $\boldsymbol{x}^{\mathrm{c}} \equiv (\boldsymbol{x}^{u}, \boldsymbol{x}^{p})$, where $\boldsymbol{x}^{u} \equiv \{x_i^u\}$ is the set of features for a user and $\boldsymbol{x}^{p} \equiv \{x_j^p\}$ is the set of features for a publisher[1]. The parameter

[1] All the features studied in our work are one-hot encoded binary features.

$\mathit{\Theta} \equiv (w_0^{\mathrm{c}}, \boldsymbol{w}^{\mathrm{c}}, \boldsymbol{V}^{\mathrm{c}})$, where $w_0^{\mathrm{c}} \in \mathbb{R}$ is the global bias term and $\boldsymbol{w}^{\mathrm{c}} \in \mathbb{R}^{I^{\mathrm{c}}+J^{\mathrm{c}}}$ is the weight vector of the I^{c}-dimensional user features and J^{c}-dimensional publisher features. Each user feature x_i^u or publisher feature x_j^p is associated with a K-dimensional latent vector $\boldsymbol{v}_i^{\mathrm{c}}$ or $\boldsymbol{v}_j^{\mathrm{c}}$. Thus $\boldsymbol{V}^{\mathrm{c}} \in \mathbb{R}^{(I^{\mathrm{c}}+J^{\mathrm{c}})\times K}$.

With such setting, the conditional probability for CF in Eq. (1) can be reformulated as:

$$\prod_{(\boldsymbol{x}^{\mathrm{c}}, y^{\mathrm{c}}) \in D^{\mathrm{c}}} P(y^{\mathrm{c}} | \boldsymbol{x}^{\mathrm{c}}; \mathit{\Theta}) = \prod_{(\boldsymbol{x}^{u}, \boldsymbol{x}^{p}, y^{\mathrm{c}}) \in D^{\mathrm{c}}} P(y^{\mathrm{c}} | \boldsymbol{x}^{u}, \boldsymbol{x}^{p}; w_0^{\mathrm{c}}, \boldsymbol{w}^{\mathrm{c}}, \boldsymbol{V}^{\mathrm{c}}). \tag{2}$$

Let $\hat{y}_{u,p}^{\mathrm{c}}$ be the predicted probability of whether user u will be interested in visiting publisher p. With the FM model, the likelihood of observing the label y^{c} given the features $(\boldsymbol{x}^u, \boldsymbol{x}^p)$ and parameters is

$$P(y^{\mathrm{c}} | \boldsymbol{x}^{u}, \boldsymbol{x}^{p}; w_0^{\mathrm{c}}, \boldsymbol{w}^{\mathrm{c}}, \boldsymbol{V}^{\mathrm{c}}) = (\hat{y}_{u,p}^{\mathrm{c}})^{y^{\mathrm{c}}} \cdot (1 - \hat{y}_{u,p}^{\mathrm{c}})^{(1-y^{\mathrm{c}})}, \tag{3}$$

where the prediction $\hat{y}_{u,p}^{\mathrm{c}}$ is given by an FM with a logistic function:

$$\hat{y}_{u,p}^{\mathrm{c}} = \sigma\Big(w_0^{\mathrm{c}} + \sum_i w_i^{\mathrm{c}} x_i^u + \sum_j w_j^{\mathrm{c}} x_j^p + \sum_i \sum_j \langle \boldsymbol{v}_i^{\mathrm{c}}, \boldsymbol{v}_j^{\mathrm{c}} \rangle x_i^u x_j^p\Big), \tag{4}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function and $\langle \cdot, \cdot \rangle$ is the inner product of two vectors: $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle \equiv \sum_{f=1}^{K} v_{i,f} \cdot v_{j,f}$, which models the interaction between a user feature i and a publisher feature j.

3.3 CTR Task Prediction Model

For a data instance $(\boldsymbol{x}^{\mathrm{r}}, y^{\mathrm{r}})$ in ad CTR task dataset D^{r}, its features $\boldsymbol{x}^{\mathrm{r}} \equiv (\boldsymbol{x}^u, \boldsymbol{x}^p, \boldsymbol{x}^a)$ can be divided into three categories: the user features $\boldsymbol{x}^u$ (cookie, location, time, device, browser, OS, etc.), the publisher features $\boldsymbol{x}^p$ (domain, URL etc.), and the ad features $\boldsymbol{x}^a$ (ad slot position, ad creative, creative size, campaign, etc.). Each feature has potential influence to another one in a different category. For example, a mobile phone user might prefer square-sized ads instead of banner ads; users would like to click the ad on the sport websites during the afternoon etc.

By the same token as CF prediction, we leverage factorisation machine and the model parameter thus is $\mathit{\Theta} \equiv (w_0^{\mathrm{r}}, \boldsymbol{w}^{\mathrm{r}}, \boldsymbol{V}^{\mathrm{r}})$. Specifically, x_l^a is one of the L^{r}-dimensional ad features $\boldsymbol{x}^a$, w_l^{r} is the corresponding bias weight for the feature, and the feature is also associated with a K-dimensional latent vector $\boldsymbol{v}_l^{\mathrm{r}}$. Thus $\boldsymbol{V}^{\mathrm{r}} \in \mathbb{R}^{(I^{\mathrm{r}}+J^{\mathrm{r}}+L^{\mathrm{r}})\times K}$. Similar to CF task, the CTR data likelihood is:

$$\prod_{(\boldsymbol{x}^{\mathrm{r}}, y^{\mathrm{r}}) \in D^{\mathrm{r}}} P(y^{\mathrm{r}} | \boldsymbol{x}^{\mathrm{r}}; \mathit{\Theta}) = \prod_{(\boldsymbol{x}^{u}, \boldsymbol{x}^{p}, \boldsymbol{x}^{a}, y^{\mathrm{r}}) \in D^{\mathrm{r}}} P(y^{\mathrm{r}} | \boldsymbol{x}^{u}, \boldsymbol{x}^{p}, \boldsymbol{x}^{a}; w_0^{\mathrm{r}}, \boldsymbol{w}^{\mathrm{r}}, \boldsymbol{V}^{\mathrm{r}}). \tag{5}$$

Then the factorisation machine with logistic activation function $\sigma(\cdot)$ is adopted to model the click probability over a specific ad impression:

$$P(y^{\mathrm{r}} | \boldsymbol{x}^{u}, \boldsymbol{x}^{p}, \boldsymbol{x}^{a}; w_0^{\mathrm{r}}, \boldsymbol{w}^{\mathrm{r}}, \boldsymbol{V}^{\mathrm{r}}) = (\hat{y}_{u,p,a}^{\mathrm{r}})^{y^{\mathrm{r}}} \cdot (1 - \hat{y}_{u,p,a}^{\mathrm{r}})^{(1-y^{\mathrm{r}})}, \tag{6}$$

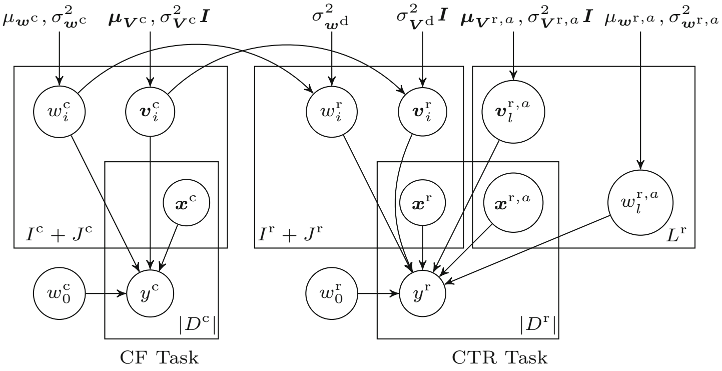
$\mu_{\boldsymbol{w}^{\mathrm{c}}}, \sigma^2_{\boldsymbol{w}^{\mathrm{c}}}$
$\boldsymbol{\mu}_{\boldsymbol{V}^{\mathrm{c}}}, \sigma^2_{\boldsymbol{V}^{\mathrm{c}}}\boldsymbol{I}$
$\sigma^2_{\boldsymbol{w}^{\mathrm{d}}}$
$\sigma^2_{\boldsymbol{V}^{\mathrm{d}}}\boldsymbol{I}$
$\boldsymbol{\mu}_{\boldsymbol{V}^{\mathrm{r},a}}, \sigma^2_{\boldsymbol{V}^{\mathrm{r},a}}\boldsymbol{I}$
$\mu_{\boldsymbol{w}^{\mathrm{r},a}}, \sigma^2_{\boldsymbol{w}^{\mathrm{r},a}}$
w_i^{c}
$\boldsymbol{v}_i^{\mathrm{c}}$
$\boldsymbol{x}^{\mathrm{c}}$
$I^{\mathrm{c}} + J^{\mathrm{c}}$
w_0^{c}
y^{c}
$|D^{\mathrm{c}}|$
w_i^{r}
$\boldsymbol{v}_i^{\mathrm{r}}$
$\boldsymbol{v}_l^{\mathrm{r},a}$
$\boldsymbol{x}^{\mathrm{r}}$
$\boldsymbol{x}^{\mathrm{r},a}$
$w_l^{\mathrm{r},a}$
$I^{\mathrm{r}} + J^{\mathrm{r}}$
L^{r}
w_0^{r}
y^{r}
$|D^{\mathrm{r}}|$
CF Task
CTR Task

Fig. 1. Graphic model of transferred factorisation machines.

where $\hat{y}^{\mathrm{r}}_{u,p,a}$ is modelled by interactions among 3-side features

$$\hat{y}^{\mathrm{r}}_{u,p,a} = \sigma\Big(w_0^{\mathrm{r}} + \sum_i w_i^{\mathrm{r}} x_i^u + \sum_j w_j^{\mathrm{r}} x_j^p + \sum_l w_l^{\mathrm{r}} x_l^a + \sum_i \sum_j \langle \boldsymbol{v}_i^{\mathrm{r}}, \boldsymbol{v}_j^{\mathrm{r}} \rangle x_i^u x_j^p + \sum_i \sum_l \langle \boldsymbol{v}_i^{\mathrm{r}}, \boldsymbol{v}_l^{\mathrm{r}} \rangle x_i^u x_l^a + \sum_j \sum_l \langle \boldsymbol{v}_j^{\mathrm{r}}, \boldsymbol{v}_l^{\mathrm{r}} \rangle x_j^p x_l^a \Big). \tag{7}$$

3.4 Dual-Task Bridge

To model the dependency between the two tasks, the weights of the user features and publisher features in CTR task are assumed to be generated from the counterparts in CF task (as a prior):

$$\boldsymbol{w}^{\mathrm{r}} \sim \mathcal{N}(\boldsymbol{w}^{\mathrm{c}}, \sigma^2_{\boldsymbol{w}^{\mathrm{d}}}\boldsymbol{I}), \tag{8}$$

where $\sigma^2_{\boldsymbol{w}^{\mathrm{d}}}$ is the assumed variance of the Gaussian generation process between each pair of feature weights of CF and CTR tasks and the weight generation is assumed to be independent across features. Similarly, the latent vectors of CTR task are assumed to be generated from the counterparts of CF task:

$$\boldsymbol{v}_i^{\mathrm{r}} \sim \mathcal{N}(\boldsymbol{v}_i^{\mathrm{c}}, \sigma^2_{\boldsymbol{V}^{\mathrm{d}}}\boldsymbol{I}) \tag{9}$$

where i is the index of a user or publisher feature; $\sigma^2_{\boldsymbol{V}^{\mathrm{d}}}$ is defined similarly.

The rational behind the above bridging model is that the users' interest towards webpage content is relatively general and the displayed ad can be regarded as a special kind of webpage content. One can infer user interests from their browsing behaviours, while their interests on commercial ads can be regarded as a modification or derivative from the learned general interests.

The graphic representation for the proposed *transferred factorisation machines* is depicted in Fig. 1. It illustrates the relationship among model parameters and observed data. The left part is for the CF task: $\boldsymbol{x}^{\mathrm{c}}$, w_0^{c}, $\boldsymbol{w}^{\mathrm{c}}$ and $\boldsymbol{V}^{\mathrm{c}}$

work together to infer our CF task target y^{c}, i.e., whether the user would visit a specific publisher or not. The right part illustrates the CTR task. Corresponding to CF task, $\boldsymbol{w}^{\mathrm{r}}$ and $\boldsymbol{V}^{\mathrm{r}}$ here represent user and publisher features' weights and latent vectors, while $\boldsymbol{w}^{\mathrm{r},a}$ and $\boldsymbol{V}^{\mathrm{r},a}$ are separately depicted to represent ad features' weights and latent vectors. All these factors work together to predict CTR task target y^{r}, i.e., whether the user would click the ad or not. On top of that, for each (user or publisher) feature i of the CF task, its weight w_i^{c} and latent vector $\boldsymbol{v}_i^{\mathrm{c}}$ act as a prior of the counterparts w_i^{r} and $\boldsymbol{v}_i^{\mathrm{r}}$ in CTR task while learning the model.

Considering the datasets of the two tasks might be seriously unbalanced, we choose to focus on the *averaged* log-likelihood of generating each data instance from the two tasks. In addition, we add a hyperparameter α for balancing the task relative importance. As such, the joint conditional likelihood in Eq. (1) is written as

$$\Big[\prod_{(\boldsymbol{x}^{\mathrm{c}},y^{\mathrm{c}})\in D^{\mathrm{c}}} P(y^{\mathrm{c}}|\boldsymbol{x}^{\mathrm{c}};\Theta)\Big]^{\frac{\alpha}{|D^{\mathrm{c}}|}} \cdot \Big[\prod_{(\boldsymbol{x}^{\mathrm{r}},y^{\mathrm{r}})\in D^{\mathrm{r}}} P(y^{\mathrm{r}}|\boldsymbol{x}^{\mathrm{r}};\Theta)\Big]^{\frac{1-\alpha}{|D^{\mathrm{r}}|}} \tag{10}$$

and its log form is

$$\begin{aligned}\frac{\alpha}{|D^{\mathrm{c}}|}&\sum_{(\boldsymbol{x}^{\mathrm{c}},y^{\mathrm{c}})\in D^{\mathrm{c}}}\Big[y^{\mathrm{c}}\log\hat{y}^{\mathrm{c}}_{u,p}+(1-y^{\mathrm{c}})\log(1-\hat{y}^{\mathrm{c}}_{u,p})\Big]\\&+\frac{1-\alpha}{|D^{\mathrm{r}}|}\sum_{(\boldsymbol{x}^{\mathrm{r}},y^{\mathrm{r}})\in D^{\mathrm{r}}}\Big[y^{\mathrm{r}}\log\hat{y}^{\mathrm{r}}_{u,p,a}+(1-y^{\mathrm{r}})\log(1-\hat{y}^{\mathrm{r}}_{u,p,a})\Big].\end{aligned} \tag{11}$$

Moreover, from the graphic model, the prior of model parameters can be specified as

$$P(\Theta) = P(\boldsymbol{w}^{\mathrm{c}})P(\boldsymbol{V}^{\mathrm{c}})P(\boldsymbol{w}^{\mathrm{r}}|\boldsymbol{w}^{\mathrm{c}})P(\boldsymbol{V}^{\mathrm{r}}|\boldsymbol{V}^{\mathrm{c}})P(\boldsymbol{w}^{\mathrm{r},a})P(\boldsymbol{V}^{\mathrm{r},a}) \tag{12}$$

$$\begin{aligned}\log P(\Theta) =&\sum_i \log\mathcal{N}(w_i^{\mathrm{c}};\mu_{\boldsymbol{w}^{\mathrm{c}}},\sigma^2_{\boldsymbol{w}^{\mathrm{c}}})+\sum_i\log\mathcal{N}(\boldsymbol{v}_i^{\mathrm{c}};\boldsymbol{\mu}_{\boldsymbol{V}^{\mathrm{c}}},\sigma^2_{\boldsymbol{V}^{\mathrm{c}}}\boldsymbol{I})\\&+\sum_i\log\mathcal{N}(w_i^{\mathrm{r}};w_i^{\mathrm{c}},\sigma^2_{\boldsymbol{w}^{\mathrm{d}}})+\sum_i\log\mathcal{N}(\boldsymbol{v}_i^{\mathrm{r}};\boldsymbol{v}_i^{\mathrm{c}},\sigma^2_{\boldsymbol{V}^{\mathrm{d}}}\boldsymbol{I})\\&+\sum_l\log\mathcal{N}(w_l^{\mathrm{r},a};\mu_{\boldsymbol{w}^{\mathrm{r},a}},\sigma^2_{\boldsymbol{w}^{\mathrm{r},a}})+\sum_l\log\mathcal{N}(\boldsymbol{v}_l^{\mathrm{r},a};\boldsymbol{\mu}_{\boldsymbol{V}^{\mathrm{r},a}},\sigma^2_{\boldsymbol{V}^{\mathrm{r},a}}\boldsymbol{I}).\end{aligned} \tag{13}$$

3.5 Learning the Model

Given the detailed implementations of the MAP solution (Eq. (1)) components in Eqs. (11) and (13), for each data instance $(\boldsymbol{x}, y)$, the gradient update of Θ is

$$\Theta \leftarrow \Theta + \eta\Big(\beta\frac{\partial}{\partial\Theta}\log P(y|\boldsymbol{x};\Theta)+\frac{\partial}{\partial\Theta}\log P(\Theta)\Big), \tag{14}$$

where $P(y|\boldsymbol{x};\Theta)$ is as Eqs. (3) and (6) for $(\boldsymbol{x}^{\mathrm{c}}, y^{\mathrm{c}}) \in D^{\mathrm{c}}$ and $(\boldsymbol{x}^{\mathrm{r}}, y^{\mathrm{r}}) \in D^{\mathrm{r}}$, respectively; η is the learning rate; β is the instance weight parameter depending on which task the instance belongs to, as given in Eq. (11). The detailed gradient for each specific parameter can be calculated routinely and thus are omitted here due to the page limit.

4 Experiments

4.1 Dataset

Our experiments are conducted based on a real-world dataset provided by Adform, a global digital media advertising technology company based in Copenhagen, Denmark. It consists of two weeks of online display ad logs across different campaigns during March 2015. Specifically, there are 42.1M user domain browsing events and 154.0 K ad display/click events. To fit the data into the joint model, we group useful data features into three categories: user features $\boldsymbol{x}^u$ (`user_cookie`, `hour`, `browser`, `os`, `user_agent` and `screen_size`), publisher features $\boldsymbol{x}^p$ (`domain`, `url`, `exchange`, `ad_slot` and `slot_size`), ad features $\boldsymbol{x}^a$ (`advertiser` and `campaign`). Detailed unique value numbers for each attribute are given as below.

Attribute	user_cookie	hour	browser	os	user_agent	screen_size	domain
Unique number	4,180,170	24	71	37	29,488	118	38,495
Attribute	url	exchange	position	size	advertiser	campaign	
Unique number	1,100,523	140	3	55	486	2,665	

In order to perform stable knowledge transfer, we have down-sampled the negative instances to make the ratio of positive over negative instances as 1:5.[2]

4.2 Experiment Protocol

We conduct a two-stage experiment to verify the effectiveness of our proposed models. First, in a very clean setting, we only focus on `user_cookie` and `domain` to check whether the knowledge of users' behaviour on webpage browsing can be transferred to model their behaviour on clicking the ads in these webpages. Second, we start to append various features in the first setting to observe the performance change and check which features lead to better transfer learning. Specifically, we try appending a single side feature into the baseline setting: 1. appending user feature $\boldsymbol{x}^u$, 2. appending publisher feature $\boldsymbol{x}^p$, 3. appending ad feature $\boldsymbol{x}^a$. Finally, all features are added into the model to perform the transfer learning.

For each experiment stage, there are three datasets: CF dataset (D^{c}), CTR dataset (D^{r}) and Joint dataset ($D^{\mathrm{c}}, D^{\mathrm{r}}$). Each dataset is split into two parts: the first week data as training data and the second one as test data.

[2] It is common to perform negative down sampling to balance the labels in ad CTR estimation [9]. Calibration methods [3] are then leveraged to eliminate the model bias.

4.3 Evaluation Metrics

To evaluate the performance of proposed model, area under the ROC curve (AUC) [8] and root mean square error (RMSE) [13] are adopted as performance metrics. As we focus on ad click prediction performance improvement, we only report the performance of the CTR estimation task.

4.4 Compared Models

We implement the following models for experimental comparison.

- Base: This baseline model only considers the ad CTR task, without any transfer learning. The parameters are learned by $\max_{\Theta} \prod_{(\boldsymbol{x}^{\mathrm{r}}, y^{\mathrm{r}}) \in D^{\mathrm{r}}} P(y^{\mathrm{r}} | \boldsymbol{x}^{\mathrm{r}}; \Theta) P(\Theta)$.
- Disjoint: This method performs a knowledge transfer in a disjoint two-stage fashion. First, we train the CF task model to get the parameters $\boldsymbol{w}^{\mathrm{c}}$ and $\boldsymbol{V}^{\mathrm{c}}$ by $\max_{\Theta} \prod_{(\boldsymbol{x}^{\mathrm{c}}, y^{\mathrm{c}}) \in D^{\mathrm{c}}} P(y^{\mathrm{c}} | \boldsymbol{x}^{\mathrm{c}}; \Theta) P(\Theta)$. Second, with the CF task parameters fixed, we train the CTR task using Eqs. (11) and (13). Note that α in Eq. (11) is still a hyperparameter for this method.
- DisjointLR: The transfer learning model proposed in [7] is considered as state-of-the-art transfer learning methods in display advertising. In this work, both source and target tasks adopt logistic regression as a behaviour prediction model, which uses the linear model to minimise the logistic loss from each observation sample:

$$\mathcal{L}_{\boldsymbol{w}}(\boldsymbol{x}, y) = -y \log \sigma(\langle \boldsymbol{w}, \boldsymbol{x} \rangle) - (1-y) \log(1 - \sigma(\langle \boldsymbol{w}, \boldsymbol{x} \rangle)). \tag{15}$$

In our context of regarding the CF task as source task and CTR task as target task, the learning objectives are listed below:

$$\text{CF TASK} : \overset{*}{\boldsymbol{w}}{}^{\mathrm{c}} = \arg\min_{\boldsymbol{w}^{\mathrm{c}}} \sum_{(\boldsymbol{x}^{\mathrm{c}}, y^{\mathrm{c}}) \in D^{\mathrm{c}}} \mathcal{L}_{\boldsymbol{w}^{\mathrm{c}}}(\boldsymbol{x}^{\mathrm{c}}, y^{\mathrm{c}}) + \lambda ||\boldsymbol{w}^{\mathrm{c}}||_2^2 \tag{16}$$

$$\text{CTR TASK} : \overset{*}{\boldsymbol{w}}{}^{\mathrm{r}} = \arg\min_{\boldsymbol{w}^{\mathrm{r}}} \sum_{(\boldsymbol{x}^{\mathrm{r}}, y^{\mathrm{r}}) \in D^{\mathrm{r}}} \mathcal{L}_{\boldsymbol{w}^{\mathrm{r}}}(\boldsymbol{x}^{\mathrm{r}}, y^{\mathrm{r}}) + \lambda ||\boldsymbol{w}^{\mathrm{r}} - \overset{*}{\boldsymbol{w}}{}^{\mathrm{c}}||_2^2. \tag{17}$$

Besides the difference between the linear LR and non-linear FM, this method is a two-stage learning scheme, where the first stage Eq. (16) is disjoint with the second stage Eq. (17). Thus we denoted it as DisjointLR.
- Joint: Our proposed model, as summarised in Eq. (1), which performs the transfer learning when jointly learning the parameters on the two tasks.

Table 1. Overall AUC performance: DisjointLR vs Joint.

DisjointLR	Joint	Improvement
68.44 %	72.18 %	5.46 %

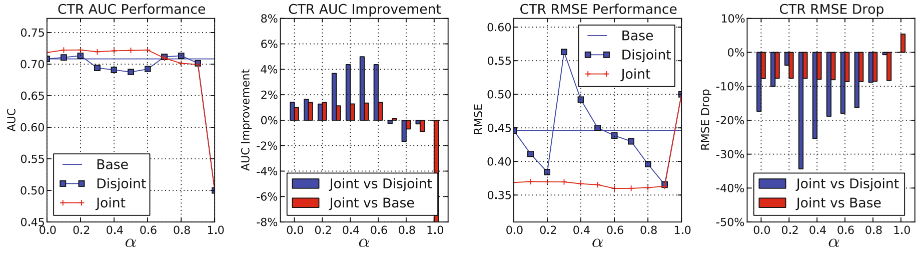

Fig. 2. Performance improvement with basic setting.

4.5 Result

Basic Setting Performance. Figure 2 presents the AUC and RMSE performance of Base, Disjoint and Joint and the improvement of Joint against the hyperparameter α in Eq. (11) based on the basic experiment setting. As can be observed clearly, for a large region of α, i.e., $[0.1, 0.7]$, Joint consistently outperforms the baselines Base and Disjoint on both AUC and RMSE, which demonstrates the effectiveness of our model to transfer knowledge from webpage browsing data to ad click data. Note that when $\alpha = 0$, the CF side model $\boldsymbol{w}^{\mathrm{c}}$ does not learn but Joint still outperforms Disjoint and Base. This is due to the different prior of $\boldsymbol{w}^{\mathrm{r}}$ and $\boldsymbol{V}^{\mathrm{r}}$ in Joint compared with those of Disjoint and Base. In addition, when $\alpha = 1$, i.e., no learning on CTR task, the performance of Joint reasonably gets back to initial guess, i.e., both AUC and RMSE are 0.5.

Table 1 shows the transfer learning performance comparison between Joint and the state-of-the-art DisjointLR with both models setting optimal hyperparameters. The improvement of Joint over DisjointLR indicates the success of (1) the joint optimisation on the two tasks to perform knowledge transfer and (2) the non-linear factorisation machine relevance model on catching feature interactions.

Appending Side Information Performance. From the Joint model as in Eq. (11) we see when α is large, e.g., 0.8, the larger weight is allocated on the CF task to optimise the joint likelihood. As such, if a large-value α leads to the optimal CTR estimation performance, it means the transfer learning takes effect. With such method, we try adding different features into the Joint model to obtain the optimal hyperparameter α leading to the highest AUC to check whether a certain feature helps transfer learning. On the contrary, if a low-value or 0 α leads to the optimal performance of Joint model when adding a certain feature, it means such feature has no effect of performing transfer learning.

Table 2 collects the AUC improvement of the Joint model for the conducted experiments. We observe that user browsing `hour`, ad slot `position` in the webpage are the most valuable features that help transfer learning, while the user `screen size` does not bring any transfer value. When adding all these features into Joint model, the optimal α is around 0.5 for AUC improvement and 0.6 for RMSE drop (see Fig. 3), which means these features along with the basic

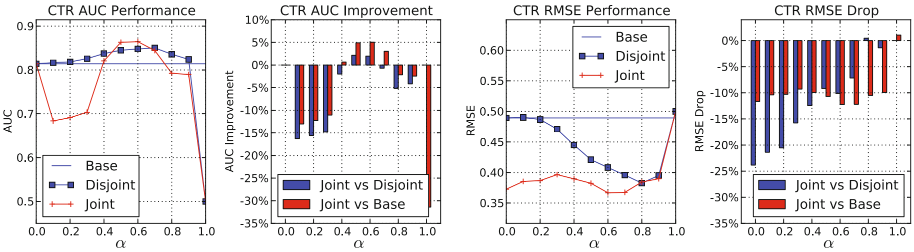

Fig. 3. Performance improvement with different side information.

Table 2. CTR task performance

	Joint vs Disjoint				Joint vs Base			
	$\overset{*}{\alpha}$	AUC Lift	Joint AUC	Disjoint AUC	$\overset{*}{\alpha}$	AUC Lift	Joint AUC	Base AUC(%)
Basic Setting	0.5	3.43 %	72.18 %	68.75 %	0.2	1.41 %	72.24 %	70.83 %
+ $\boldsymbol{x}^u$: hour	0.8	2.44 %	89.35 %	86.91 %	0.6	1.99 %	89.35 %	87.36 %
+ $\boldsymbol{x}^u$: browser	0.0	7.92 %	76.36 %	68.44 %	0.2	8.08 %	76.52 %	68.44 %
+ $\boldsymbol{x}^u$: os	0.1	6.66 %	76.86 %	70.2 %	0.1	6.71 %	76.86 %	70.15 %
+ $\boldsymbol{x}^u$: user_agent	0.0	2.57 %	67.12 %	64.55 %	0.8	4.31 %	68.86 %	64.55 %
+ $\boldsymbol{x}^u$: screen_size	0.0	9.39 %	76.43 %	67.04 %	0.0	9.39 %	76.43 %	67.04 %
+ $\boldsymbol{x}^p$: exchange	0.6	1.56 %	66.80 %	65.24 %	0.0	0.64 %	68.49 %	67.85 %
+ $\boldsymbol{x}^p$: url	0.3	11.9 %	66.56 %	54.66 %	0.0	11.55 %	69.36 %	57.81 %
+ $\boldsymbol{x}^p$: position	0.6	2.63 %	66.89 %	64.26 %	0.4	0.69 %	67.14 %	66.45 %
+ $\boldsymbol{x}^a$: advertiser	0.4	2.39 %	84.98 %	82.59 %	0.5	0.87 %	85.07 %	84.20 %
+ $\boldsymbol{x}^a$: campaign	0.2	1.29 %	85.81 %	84.52 %	0.1	0.48 %	85.91 %	85.43 %
+ $\boldsymbol{x}^a$: size	0.0	0.59 %	69.16 %	68.57 %	0.0	0.59 %	69.16 %	68.57 %
+ all features	0.5	6.91 %	88.32 %	81.41 %	0.6	6.91 %	88.32 %	81.41 %

user, webpage IDs provide an overall positive value of knowledge transfer from webpage browsing behaviour to ad click behaviour.

5 Conclusion

In this paper, we proposed a transfer learning framework with factorisation machines to build implicit look-alike models on user ad click behaviour prediction task with the knowledge successfully transferred from the rich data of user webpage browsing behaviour. The major novelty of this work lies in the joint training on the two tasks and making knowledge transfer based on the non-linear factorisation machine model to build the user and other feature profiles. Comprehensive experiments on a large-scale real-world dataset demonstrated the effectiveness of our model as well as some insights of detecting which specific features help transfer learning. In the future work, we plan to explore on the user profiling utilisation based on the learned latent vector for each user. We also plan to extend our model to cross-domain recommendation problems.

Acknowledgement. We would like to thank Adform for allowing us to use their data in experiments. We would also like to thank Thomas Furmston for his feedback on the paper. Weinan thanks Chinese Scholarship Council for the research support.

References

1. Ahmed, A., Low, Y., Aly, M., Josifovski, V., Smola, A.J.: Scalable distributed inference of dynamic user interests for behavioral targeting. In: KDD (2011)
2. Broder, A., Fontoura, M., Josifovski, V., Riedel, L.: A semantic approach to contextual advertising. In: SIGIR, pp. 559–566. ACM (2007)
3. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: ICML, pp. 161–168. ACM (2006)
4. Chapelle, O.: Modeling delayed feedback in display advertising. In: KDD, pp. 1097–1105. ACM (2014)
5. Chapelle, O., et al.: A simple and scalable response prediction for display advertising. ACM Trans. Intell. Syst. Technol. (TIST) **5**(4), 61 (2013)
6. Dai, W., Xue, G.R., Yang, Q., Yu, Y.: Transferring naive bayes classifiers for text classification. In: AAAI (2007)
7. Dalessandro, B., Chen, D., Raeder, T., Perlich, C., Han Williams, M., Provost, F.: Scalable hands-free transfer learning for online advertising. In: KDD (2014)
8. Graepel, T., Candela, J.Q., Borchert, T., Herbrich, R.: Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In: ICML, pp. 13–20 (2010)
9. 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: ADKDD, pp. 1–9. ACM(2014)
10. Hofmann, T.: Collaborative filtering via gaussian probabilistic latent semantic analysis. In: SIGIR, pp. 259–266. ACM (2003)
11. Jebara, T.: Machine Learning: Discriminative and Generative, vol. 755. Springer Science & Business Media, New York (2012)
12. Juan, Y.C., Zhuang, Y., Chin, W.S.: 3 Idiots Approach for Display Advertising Challenge. Internet and Network Economics, pp. 254–265. Springer, Heidelberg (2011)
13. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Comput. **8**, 30–37 (2009)
14. Lee, K., Orten, B., Dasdan, A., Li, W.: Estimating conversion rate in display advertising from past performance data. In: KDD, pp. 768–776. ACM (2012)
15. Li, B., Yang, Q., Xue, X.: Transfer learning for collaborative filtering via a rating-matrix generative model. In: ICML, pp. 617–624. ACM (2009)
16. Liao, X., Xue, Y., Carin, L.: Logistic regression with an auxiliary data source. In: ICML, pp. 505–512. ACM (2005)
17. Mangalampalli, A., Ratnaparkhi, A., Hatch, A.O., Bagherjeiran, A., Parekh, R., Pudi, V.: A feature-pair-based associative classification approach to look-alike modeling for conversion-oriented user-targeting in tail campaigns. In: WWW, pp. 85–86. ACM (2011)
18. McAfee, R.P.: The design of advertising exchanges. Rev. Ind. Organ. **39**(3), 169–185 (2011)
19. Muthukrishnan, S.: Ad exchanges: research issues. In: Leonardi, S. (ed.) WINE 2009. LNCS, vol. 5929, pp. 1–12. Springer, Heidelberg (2009)

20. Oentaryo, R.J., Lim, E.P., Low, D.J.W., Lo, D., Finegold, M.: Predicting response in mobile advertising with hierarchical importance-aware factorization machine. In: WSDM (2014)
21. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
22. PricewaterhouseCoopers: IAB internet advertising revenue report (2014). Accessed 29 July 2015. http://www.iab.net/media/file/PwC_IAB_Webinar_Presentation_HY2014.pdf
23. Rendle, S.: Factorization machines. In: ICDM, pp. 995–1000. IEEE (2010)
24. Richardson, M., Dominowska, E., Ragno, R.: Predicting clicks: estimating the click-through rate for new ads. In: WWW, pp. 521–530. ACM (2007)
25. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW, pp. 285–295. ACM (2001)
26. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg (2007)
27. Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: a survey. J. Mach. Learn. Res. **10**, 1633–1685 (2009)
28. Wang, J., De Vries, A.P., Reinders, M.J.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: SIGIR (2006)
29. Yan, J., Liu, N., Wang, G., Zhang, W., Jiang, Y., Chen, Z.: How much can behavioral targeting help online advertising?. In: WWW, pp. 261–270. ACM (2009)
30. Yan, L., Li, W.J., Xue, G.R., Han, D.: Coupled group lasso for web-scale ctr prediction in display advertising. In: ICML, pp. 802–810 (2014)
31. Yuan, S., Wang, J., Zhao, X.: Real-time bidding for online advertising: measurement and analysis. In: ADKDD, pp. 3. ACM (2013)
32. Zhang, W., Yuan, S., Wang, J.: Real-time bidding benchmarking with ipinyou dataset. arXiv preprint.(2014). arxiv:1407.7073