Electronic Commerce Research and Applications 16 (2016) 30–42

Contents lists available at ScienceDirect

Electronic Commerce Research and Applications

journal homepage: www.elsevier.com/locate/ecra

Predicting ad click-through rates via feature-based fully coupled interaction tensor factorization

Lili Shan *, Lei Lin, Chengjie Sun, Xiaolong Wang

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

ARTICLE INFO

Article history:
Received 24 May 2015
Received in revised form 23 January 2016
Accepted 23 January 2016
Available online 3 February 2016

Keywords:
Click through rate prediction
Tensor factorization
Feature-based
Fully coupled interaction
Real-time bidding
Demand-side platform

ABSTRACT

In the real-time bidding (RTB) display advertising ecosystem, when receiving a bid request, the demand-side platform (DSP) needs to predict the click-through rate (CTR) for ads and calculate the bid price according to the CTR estimated. In addition to challenges similar to those encountered in sponsored search advertising, such as data sparsity and cold start problems, more complicated feature interactions involving multi-aspects, such as the user, publisher and advertiser, make CTR estimation in RTB more difficult. We consider CTR estimation in RTB as a tensor complement problem and propose a fully coupled interactions tensor factorization (FCTF) model based on Tucker decomposition (TD) to model three pairwise interactions between the user, publisher and advertiser and ultimately complete the tensor complement task. FCTF is a special case of the Tucker decomposition model; however, it is linear in runtime for both learning and prediction. Different from pairwise interaction tensor factorization (PITF), which is another special case of TD, FCTF is independent from the Bayesian personalized ranking optimization algorithm and is applicable to generic third-order tensor decomposition with popular simple optimizations, such as the least square method or mean square error. In addition, we also incorporate all explicit information obtained from different aspects into the FCTF model to alleviate the impact of cold start and sparse data on the final performance. We compare the performance and runtime complexity of our method with Tucker decomposition, canonical decomposition and other popular methods for CTR prediction over real-world advertising datasets. Our experimental results demonstrate that the improved model not only achieves better prediction quality than the others due to considering fully coupled interactions between three entities, user, publisher and advertiser but also can accomplish training and prediction with linear runtime.

1. Introduction

With the emergence and development of spot markets, real-time bidding (RTB) advertising has become an increasingly important way for publishers to sell their ad inventory. In the RTB-enabled display advertising ecosystem (see Fig. 1), there are three major entities: the supply-side platform (on behalf of publishers), ad exchange and demand-side platform (on behalf of advertisers). The arrows in Fig. 1 represent the path an advertiser's dollar could take to reach a publisher. Publishers supply advertising inventory to advertisers through ad exchange systems. Ad exchanges aggregate advertising inventory from multiple publishers and sell ads to several demand-side platforms (DSPs) via real-time auction or bidding. DSPs help many advertisers manage their display advertising campaigns simultaneously across multiple direct ad exchanges and employ specialized technology solutions to reach the most receptive online audiences in the right context, who will then hopefully click their displayed ads and eventually take a desired action.

The display of an ad on a webpage to a user is considered an *ad impression*. The life of a programmatic RTB ad impression is illustrated in Fig. 2. When a user clicks a hyperlink to a publisher's webpage, besides producing and showing high-quality content to the user, the publisher's main task is to sell its ad inventory to advertisers for monetization. If the publisher decides to monetize the ad impression through RTB, the publisher would pass the ad slot information to an ad exchange through a supply-side platform (SSP), and then the ad exchange composes a bid request and sends it to several DSPs. When receiving a bid request, a DSP needs to use bidding algorithms to decide whether to bid for the ad impression and what the appropriate bidding price is and then reply to the ad exchange in real-time. The impression will be sold to the highest bidder in the public auction. The publisher's web server requests the advertiser's ad server for the winning ad creative and displays

* Corresponding author.

http://dx.doi.org/10.1016/j.elerap.2016.01.004
1567-4223/

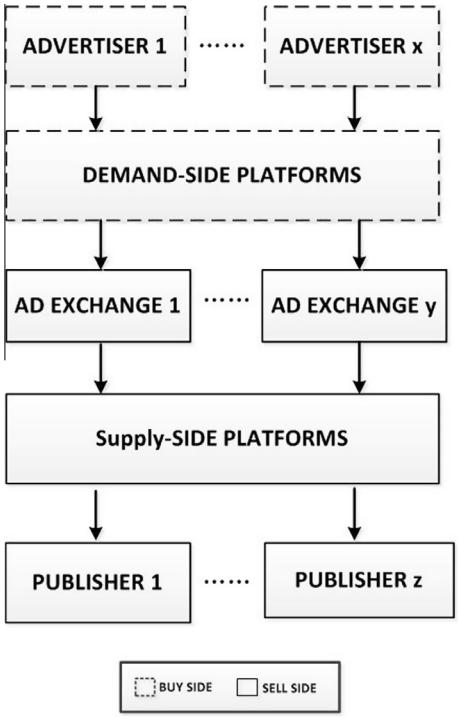

Fig. 1. Real-time bidding ecosystem through the lens of advertisers.

it on their webpage. Finally, the user will see the ad on the publisher's webpage. Note that the entire life of a programmatic RTB ad impression, from a user clicking a hyperlink to the publisher's webpage to a final ad impression, must be accomplished during a very short interval of time, such as 100 ms.

From the perspective of a DSP, to achieve optimal return on investment (ROI), each bid price must be lower than the expected cost-per-impression (eCPM) of that impression. If cost per click (CPC) is the pricing model between DSPs and advertisers, eCPM is equal to the click-through-rate (CTR) for the impression multiplied by the CPC (or click value for advertisers). Therefore, the eCPM directly depends on how well the CTR can be estimated. If the CTR is overestimated, bid prices will be higher than what they should be, and the campaign budget of advertisers will be wasted on useless impressions. Conversely, if these quantities are underestimated, high-value impressions that may have led to actions will be missed, and the campaign will under-deliver (Lee et al. 2012).

In this paper, we focus on the DSP's approach for CTR estimation for ad impressions in real-time bidding systems. We consider the CTR prediction problem as a recommendation problem, where ads must be recommended for appropriate users and collaborative filtering techniques are employed to handle this. However, compared with recommender systems or sponsored search advertising, there are at least three new challenges specific to this problem. Firstly, more complicated interactions between multiple features primarily involving at least three different aspects (the user, publisher and advertiser) greatly increase the difficulty of CTR prediction in RTB. This is because a user's response to an impression relies on how well the topic of the ad matches the user's requirements, as well as the quality of the publishing context. Publishing context involves the size, position and format of an ad slot, the content of the surrounding webpage, etc. For example, if an ad is outside the range of a user's vision, even if the content of the ad exactly satisfies the user's taste, the probability of the ad being clicked will still be dramatically reduced. Secondly, RTB compels the DSP to reply to the bid request in real-time. Therefore, the DSP must select a time-saving prediction algorithm for ad CTR to ensure its timely response. Lastly, the data in RTB being more extremely sparse and imbalanced than in other systems makes it more difficult to achieve better optimal prediction results. Data statistics show that the average CTR for desktop display advertising in practice is usually approximately 0.1%. This quantity is not only far less than that in recommendation datasets, such as the approximate 4.5% in MovieLens or 1.2% in Netflix (Lü et al. 2012), but it is also far lower than the average click-through rate on AdWords paid search ads (2%) according to Google. Factorization models provide a powerful technique to make use of explicit data to overcome the sparsity of the implicit data. Popular latent factor models based on a matrix for collaborative filtering have been successfully used to address the recommendation problem (Koren et al. 2009; Chen et al. 2012) and even ad CTR prediction for sponsored search (Wu et al. 2012). In typical recommender systems or sponsored

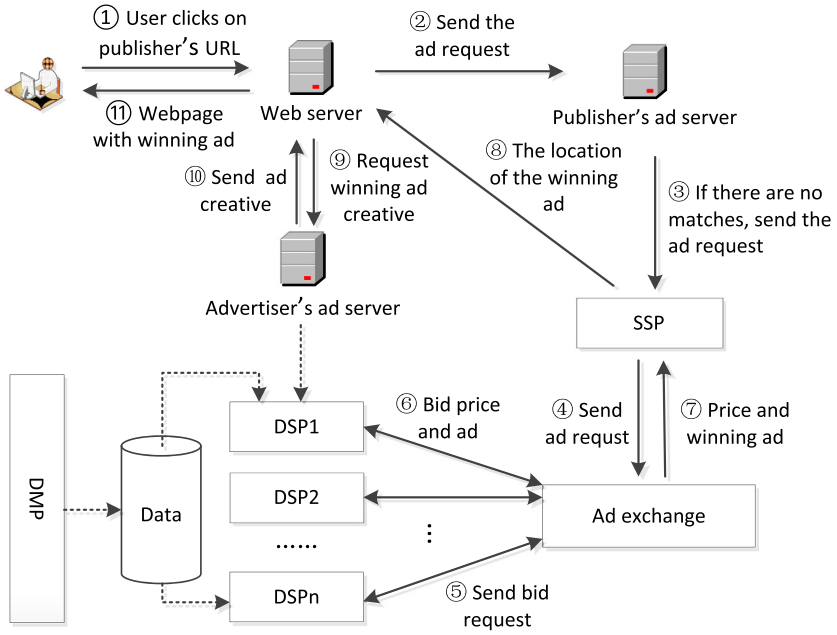

Fig. 2. The life of a programmatic RTB ad impression.

search advertising, the relations that need to be learned are usually binary ones between users and candidate items (ads). However, for display advertising in RTB, the complicated interactions are ternary relations between users, publishers and ads, as shown in Fig. 3. Both the matrix factorization model and linear regression model widely used in industrial systems are weak in regard to learning these complicated interactions. Therefore, factorization models based on the Tucker decomposition model (TD), such as higher order singular value decomposition (HOSVD) (Symeonidis 2008) and ranking tensor factorization (Rendle et al. 2009), which have been used to exploit ternary relations between users, items and tags for personalized tag recommendation, are skilled in learning these complicated interactions between more than three aspects. However, the drawback of using the full TD is that the model equation is cubic in the factorization dimension. This makes it unfeasible for TD models to be used with a high factorization dimension for midsized and large datasets (Rendle and Schmidt-Thieme 2010). Thus, it is difficult to apply TD models directly to address CTR estimation in real-time for RTB. There are other tensor decomposition models, such as the canonical decomposition (CD) and the pairwise interaction tensor factorization (PITF) (Rendle and Schmidt-Thieme 2010), that have linear runtime complexity for the number of factorization dimensions, but the performance of the CD model for solving our problem is not outstanding, and the PITF model depends on specific optimization criteria and particular data interpretations that are not suitable for our circumstances. To address these problems, we propose a novel tensor factorization named fully coupled interaction tensor factorization (FCTF), which is based on the Tucker decomposition model. Because it considers fully coupled interactions between three entities (the user, publisher and advertiser), FCTF not only has linear runtime complexity for training and prediction but also has more promising performance than traditional factorization models.

The rest of the paper is organized as follows:

We discuss related work in the next section. In Section 3, we give the notations and formulate the task performed in our system. Section 4 gives the specific implementation of tensor factorization models and explains the relationship between FCTF and these approaches. In Section 5, we propose the featured-based FCTF model, which incorporates side information to relieve the impact of sparse data on the final performance. Experimental results and analyses are given in Section 6, and in the last section, we summarize and outline the possible future work in this research direction.

2. Related work

2.1. Ad click-through rate prediction

Existing methods for CTR prediction can be categorized as feature-based or maximum likelihood estimate-based. MLE-based methods usually smooth the raw MLE via statistical models of clicks and impressions, with popular choices being the Gamma-Poisson model (Agarwal et al. 2009; Agarwal et al. 2010) and Binomial model (Lee et al. 2012). Feature-based learning methods often use standard classification or regression models in which all factors that have an impact on a user's response are included explicitly as features (Yan et al. 2014; Chapelle et al. 2014; Menon et al. 2011; Shan et al. 2014; Richardson et al. 2007; Zhang et al. 2014; Graepel et al. 2010). Fighting against data sparsity is one of the main tasks for CTR prediction. Data hierarchies of explicit features of the publisher, advertiser or user are frequently used to relieve data sparsity in both MLE-based and feature-based methods (Lee et al. 2012; Menon et al. 2011; Agarwal et al. 2010; Oentaryo et al. 2014; Wang et al. 2010). Owing to its easy implementation, immediate prediction and acceptable performance, logistic regression (LR) or generalized linear models have been widely applied for ad CTR prediction (Lee et al. 2012; Yan et al. 2014; Chapelle et al. 2014) based on features in display advertising, especially in industrial systems (Graepel et al. 2010). However, LR is a linear model in which the features contribute to the final prediction independently. Therefore, other collaborative filtering models, such as matrix factorization or its variants, which are popular in recommender systems owing to their significant performance (Koren et al. 2009), are adopted to create a personalized click model for web search (Shen et al. 2012) or to cope with response prediction for online advertising (Wu et al. 2012; Menon et al. 2011. Kuan-Wei Wu et al. model the ad CTR prediction in sponsored search advertising (track 2 of KDD Cup 2012) as a recommendation problem solvable by matrix factorization (MF) (Wu et al. 2012), which becomes the best individual model they have. Collaborative filtering techniques are again used in De Lathauwer et al. (2000) with hierarchies and side information for response prediction for publishers to counter sparse data and cold-start pages and ads. However, CTR prediction in display advertising encounters more difficult situations than in recommender systems or in sponsored search advertising. In addition to sparser data, there are more diverse types of ad formats, a richer variety of ad slots, and more possible actions of the user than in sponsored search. Thus, all these factors lead to more complicated interactions between these features, which must be exploited for better-quality CTR estimation. Both LR and MF models are weak in regard to capturing the ternary complex relationship between features of the user, publisher and ad in RTB display advertising. Therefore, some tensor factorization models have been attempted to learn such ternary relations (Shan et al. 2014), and they have achieved better prediction quality than various traditional models. However, the runtime complexity of a tensor factorization model is cubic, which makes it unfeasible for large datasets when using high factorization dimensions.

2.2. Tensor factorization models

Tensor factorization models have been widely applied to solve the personalized recommendation problem in recommender systems (Symeonidis 2008; Rendle et al. 2009) and train personalized click models for web search (Shen et al. 2012). Factorization models based on the Tucker decomposition (TD) model, such as higher order singular value decomposition (HOSVD) (Symeonidis 2008) and ranking tensor factorization (Rendle et al. 2009), have been used to exploit ternary relations between users, items and tags for personalized tag recommendation. Although these models can directly exploit all information of the ternary relations between users, items and tags, due to the high time complexity of these tensor factorization models for training and prediction, it is unfeasible to directly apply them to cope with ad CTR prediction in RTB.

There is another tensor factorization-based model, the pairwise interaction tensor factorization (PITF) model (Rendle and Schmidt-Thieme 2010), that is also used to address personalized tag recommendation and presents outstanding prediction quality over other personalized tag recommendation algorithms. Furthermore, PITF is indeed linear in runtime complexity for training and prediction. However, the PITF model is dependent on Bayesian personalized ranking (BPR) optimization criteria, which are specially designed for the personalized tag recommendation scenario, and not all problems have a BRP optimization-based solution (including CTR prediction in RTB). To overcome this problem, we extend the two-pairwise-interactions tensor factorization model that fits personalized tag recommendation based on BPR optimization to a three-dimensional fully coupled interactions model that fits generic third-order tensor factorization. Consequently, not only is the runtime complexity of the model equation linear for the

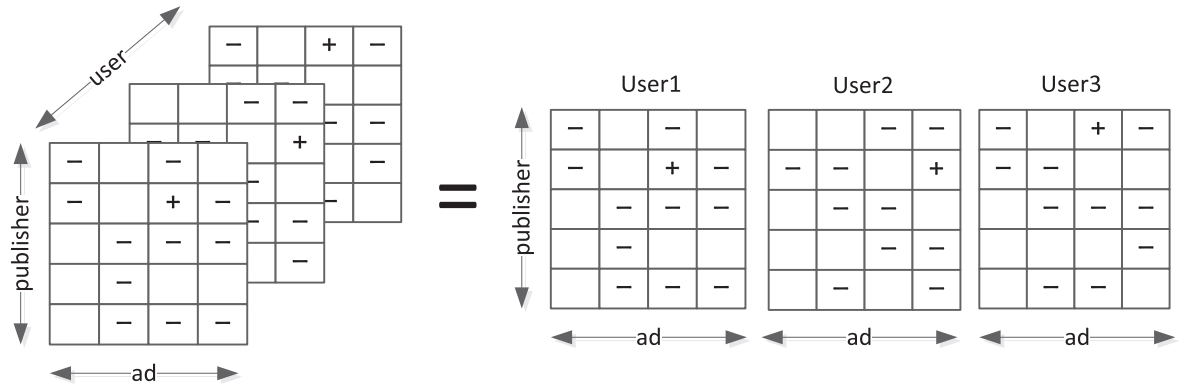

Fig. 3. A 3rd-order tensor with missing values represents a ternary relation D between users U, ads A and publisher P. If user u clicks ad a in the context of publisher p, then the cell (u, p, a) is assigned " + "; otherwise, it is assigned "-." If the triple (u, p, a) has never been observed before, the entry is left empty.

number of factorization dimensions, but this model is suitable for generic third-order tensor factorization as well, no matter what type of optimization strategy is used.

3. Problem formulation

A bid request that an ad exchange sends to a DSP is denoted as

$$bid = \{user : u, publishing\ context(\text{or } publisher) : p\} \quad (1)$$

This indicates that a user u has just clicked a link to a webpage where there is an ad impression occasion with the publishing context p. An ad impression occasion refers to there being an ad slot on a publisher's webpage where an ad has a chance of being displayed. The publishing context p primarily consists of the page domain, size of the ad slot, slot format and/or slot position. Therefore, we also refer to this information as publisher features. The DSP has an ad set $A = \{a_1, a_2, ..., a_n\}$ whose elements need to be displayed on the publisher's webpage. Furthermore, the DSP collects user features offline to target a personalized audience for their advertisers. Therefore, when a bid request arrives, information relevant to the users, publishers and ads is aggregated together on the DSP side. The bidding algorithm of the DSP estimates the click-through rate for each ad and then determines the next action.

For the formalization of ad click-through rate prediction, we define a triple (u,p,a) as an impression in which an ad a is impressed to a user u in a publishing context p. In this paper, we define a user response to an impression as either a *click* behavior or *non-click* behavior. We let U be the set of all users, P the set of all publishing contexts (publishers) and A the set of all ads. The historical impression information is denoted by a triple set $D \subseteq U \times P \times A$. A triple $(u,p,a) \in D$ means that an ad a has ever been impressed to a user u in a publishing context p in the past. Given a training set $T = \{(e_i, r_i)|i = 1, \ldots, N\}$ in which $e_i \in D$ with the form (u,p,a) and $r_i \in \{0,1\}$ with $r_i = 1$ denoting a *click* event and $r_i = 0$ denoting a *non-click* event after an ad impression, an incomplete third-order tensor $\mathcal{R} \in \mathbb{R}^{|U|\times|P|\times|A|}$ is created to represent the ternary relationships between users, publishers and ads, as shown in Fig. 3. Each element of (u, p, a) has one of three values: 1, 0 and unknown. If (u, p, a) is observed in historical impressions, that is to say, $(u,p,a) \in D$, the entry at (u,p,a) is the user response to that impression, "1" for a click event and "0" for a non-click event. If (u, p, a) is un-occurred in historical impressions, that is to say, $(u,p,a) \notin D$, the entry at (u,p,a) is defined as *unknown* or *missing*. Then, our purpose is to fill in these unknown entries with the predicted scores that indicate the probabilities of a user's click behavior after ad impressions. We treat this problem as a cube complement problem for the tensor $\mathcal{R}$ and employ a tensor factorization model to predict the unknown values inside $\mathcal{R}$.

4. CTR Prediction with tensor factorization models

Display advertising in an RTB system involves more features regarding contexts (publishers) as well as users and ads (advertisers), such as the position of an ad slot in a webpage, ad format (fixed, pop, float, background, etc.), and ad visibility (first view, second view, etc.). This induces more complicated tripartite interactions between these features, which should be taken into consideration when CTR is predicted. However, approaches that are currently widely used in industrial systems, such as linear models and regularized matrix factorization models, are weak in modeling these complicated tripartite interactions between publishers, users and ads. For example, for matrix factorization models and the bilateral interactions between users and items, although three types of attributes are available according to a user, an ad and a publisher in our problem, these features are divided into two groups. Without loss of generality, it is alleged that one group is composed of all attributes from a user and that the other group is composed of those from an ad or a publisher. Thus, the interactions between the ad and the publisher, both inside the second group, will not be learned. The CTR estimation in RTB is a tripartite interaction between publishers, users and ads. For example, an ad has a lower probability of being clicked if it is placed outside of the first view than in the first view (publisher's aspect), no matter how well the content of the ad (ad's aspect) conforms to the user's preferences (user's aspect). For another example, the topic of the webpage clicked by a user reveals the user's real-time intention, and whether the user will click an ad on the page is influenced not only by how well the topic of the ad satisfies the preference of the user (bilateral interaction) but also by to what extent the content of the ad matches the topic of the page (tripartite interaction). For example, if the same ad related to merchandise discounts is displayed to the same user but under different publishing contexts, one time on an online learning website, and another time on an e-commence website, it seems reasonable to obtain different CTR estimation values for these different ad impressions. Therefore, we consider this interaction as a ternary relation between users, publishers and ads, as shown in Fig. 3, and apply a tensor factorization model to address the three-dimensional cube complement problem (Shan et al. 2014). The main idea is to capture the underlying relationships between users-publishers-ads by reducing the rank of the original tensor to minimize the effect of noise on the underlying population.

There are two main approaches to tensor factorization, the so-called Tucker/higher-order singular value decomposition (Tucker 1966; De Lathauwer et al. 2000) and canonical decomposition (Carroll and Chang 1970), and parallel factors (Harshman 1970) (the CP expansion). In fact, they are all based on Tucker decomposition. In this section, we give the specific implementation of these approaches and explain the relationship between our FCTF model and other approaches.

4.1. Tucker decomposition and canonical decomposition model

Tucker decomposition (TD) (Tucker 1966) and high-order singular value decomposition (HOSVD) (De Lathauwer et al. 2000) extend two-dimensional-matrix singular value decomposition (SVD) to high-order tensors. According to their main ideas, a third-order tensor $\mathcal{R} \in \mathbb{R}^{|U|\times|P|\times|A|}$ can be factored into (see Fig. 4):

$$\mathcal{R} = \mathcal{C}\times_u U \times_p P \times_a A \tag{2}$$

in which,

$$\mathcal{C} \in \mathbb{R}^{k_u\times k_p\times k_a}, U \in \mathbb{R}^{|U|\times k_u}, P \in \mathbb{R}^{|P|\times k_p}, A \in \mathbb{R}^{|A|\times k_a} \tag{3}$$

and k_u, k_p and k_a are latent factor numbers corresponding to user, publisher and ad, respectively. We apply the Tucker decomposition factorization model to address the third-order tensor $\mathcal{R}$ complement problem (Shan et al. 2014), and then, according to the Tucker decomposition model, the value of element (u, p, a) can be estimated by the following equation:

$$\hat{r}^{TD}_{u,p,a} = \sum_{l}^{k_u}\sum_{m}^{k_p}\sum_{n}^{k_a} c_{l,m,n}\cdot u_{u,l}\cdot p_{p,m}\cdot a_{a,n} \tag{4}$$

where $c_{l,m,n} \in \mathbb{R}, u_{u,l} \in \mathbb{R}, p_{p,m} \in \mathbb{R}$ and $a_{a,n} \in \mathbb{R}$ are all model parameters that need to be learned. Then, for example, given any instance triple (u, p, a), where u, p and a are features respectively corresponding to user, publisher and advertiser, three latent factor vectors $u_u \in \mathbb{R}^{k_u}, p_p \in \mathbb{R}^{k_p}$, and $a_a \in \mathbb{R}^{k_a}$, respectively corresponding to the features u, p and a, and $\mathcal{C}$, a smaller tensor than $\mathcal{R}$, are learned through training. The entry value at (u, p, a) can be estimated by Eq. (4).

Obviously, if $k = \min(k_u, k_p, k_a)$, the runtime complexity for predicting one triple (u, p, a) is $O(k^3)$. This makes it unfeasible to use a high factorization dimension for midsized and large datasets and also difficult to satisfy the high real-time requirement in RTB for CTR estimation.

In fact, there is another tensor factorization model that can achieve prediction with linear runtime complexity called the canonical decomposition model (CD) (Carroll and Chang 1970; Harshman 1970). CD is a special case of the general Tucker decomposition model, as illustrated in Fig. 4, when the core tensor $\mathcal{C}$ is a diagonal one where:

$$c_{l,m,n} = \begin{cases} 1, & \text{if } l = m = n \\ 0, & \text{else} \end{cases} \tag{5}$$

The above assumption results in the triple (u, p, a) being estimated by:

$$\hat{r}^{CD}_{u,p,a} = \sum_{f}^{k} u_{u,f}\cdot p_{p,f}\cdot a_{a,f} \tag{6}$$

where $u_{u,f}\cdot p_{p,f}$ and $a_{a,f}$ are all model parameters that need to be learned. The corresponding tensor product formula is:

$$\mathcal{R} = \mathcal{C}\times_u U \times_p P \times_a A \tag{7}$$

where, excluding tensor $\mathcal{C}$, U, P and A are all model parameters that need to be learned. Obviously, only the first $k = \min(k_u, k_p, k_a)$ features are used, i.e., if the dimensionality of the feature matrices differs, some features are not used, as the core will be 0 for these entries (Rendle and Schmidt-Thieme 2010). Obviously, the CD model has a much better runtime complexity as the model equation contains no nested sums and thus is in $O(k)$. Although better runtime complexity makes CD feasible for application to a high factorization dimension for midsized or large datasets, the TD or CD model does not consider the individual interaction information between each couple within the three entities u, p, and a. More specifically, from Eqs. (4) and (6), we can find that, using the TD or CD model, we can achieve only one factor vector for each entity of the triple (u, p, a). Each factor vector is learned through interacting simultaneously with the other two objects. For instance, the latent factor $\hat{u}$ is learned for user u as its new representation by interacting simultaneously with the other two objects, p and a. Therefore, to achieve such a trade-off between interacting with p and a, $\hat{u}$ is actually neither the best representation of user U interrelating with p nor the best representation of user U interrelating with a. Thus, as we can see from the following experiment results, its performance is not outstanding.

HOSVD is a special case of Tucker decomposition when the matrices involved in Eq. (7) are orthogonal and matrix slices of core tensor $\mathcal{C}$ are mutually orthogonal.

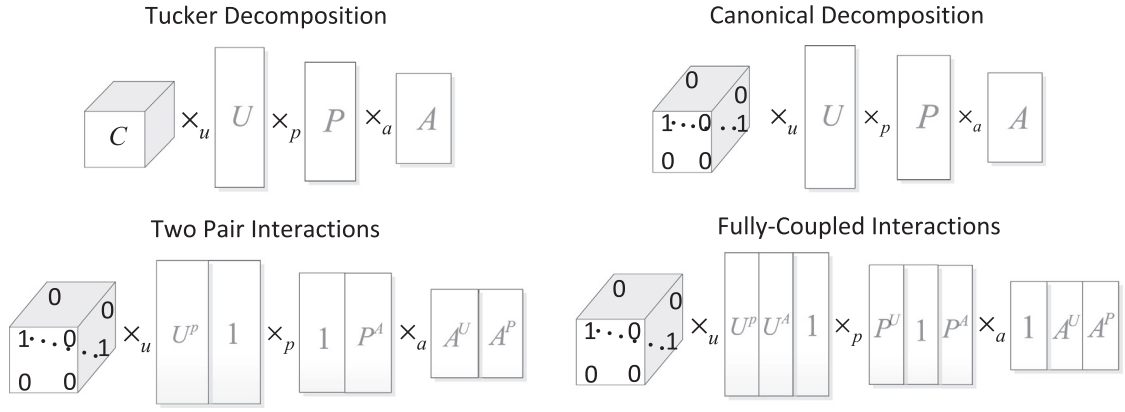

Fig. 4. Tensor Factorization models: $\boldsymbol{C}, \boldsymbol{U}, \boldsymbol{P}$ ***and*** $\boldsymbol{A}$ are all of the model parameters. The core $\boldsymbol{C}$ in Tucker Decomposition is variable, but it is fixed as a diagonal one in others, however. The factorization dimensions can differ in TD and be equal in others. In Pairwise and Fully Coupled Interactions, different parts of the feature matrices are fixed, which corresponds to modeling pairwise interaction.

4.2. Pairwise Interaction tensor factorization model

Another tensor factorization model with linear runtime complexity for training and prediction is the pairwise interaction tensor factorization (PITF) model, which was proposed for personalized tag recommendation. This model builds on special optimization criterion Bayesian personalized ranking (BPR) (Rendle and Schmidt-Thieme 2010; Rendle et al. 2009). The BPR learning algorithm is elaborately designed for personalized tag recommendation to learn the interaction between users, items and tags. However, due to this particular algorithm, PITF explicitly models only two pair interactions, users-tags and items-tags, excluding users-items. More details are given as follows.

The purpose of a personalized tag recommender is to recommend a personalized list of tags that depends on both the user and the item. That is to say, for a given post (u, i), the personalized tag recommender needs to rank all candidate tags according to their relevance to the post (u, i) and recommend the top N tags of the ranking to user u based on the tagging of item i. The personalized tag recommender takes the user's past tagging behavior into account when recommending tags. According to the BPR learning algorithm, a training sample is constructed into a quadruple (u, i, t_A, t_B), which indicates that user u assigns tag t_A and not t_B to item i. The optimization function of BPR is:

$$\text{BPR} - \text{OPT} = \ln \prod_{(u,i,t_A,t_B)\in D} \sigma\left(y_{u,i,t_A,t_B}\right) p(\Theta) \tag{8}$$

where $y_{u,i,t_A,t_B} = y_{u,i,t_A} - y_{u,i,t_B}$, y is a scoring function for a triple (u, i, t), and σ is the logistic function:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

From Eq. (8), we can see that, for tag recommendation, when given a post (u, i), the interaction score between user u and item i in y_{u,i,t_A} equals that in y_{u,i,t_B} (first necessary condition). Thus, the *user-item* interaction score does not affect the final value, y_{u,i,t_A,t_B}. Then, the user-item interaction score is removed from the finally estimated score, and only two other interaction relationships are retained, as shown in Eq. (9). However, such an approach will not change the final ranking of all candidate tags (second necessary condition). This results in the final model equation as the PITF model (Rendle and Schmidt-Thieme 2010):

$$y_{u,i,t} = \sum_f u_{uf} \cdot t^U_{tf} + \sum_f i_{if} \cdot t^I_{tf} \tag{9}$$

with the following model parameters: $U \in \mathbb{R}^{|U|\times k}$, $I \in \mathbb{R}^{|I|\times k}, T^U \in \mathbb{R}^{|T|\times k}, T^I \in \mathbb{R}^{|T|\times k}$ and $u_u, t^U_t, t^I_t, i_i, \in \mathbb{R}^k$ are latent factors corresponding to the user u, tag t involved with the user, and tag t involved with the item. Meanwhile, $u_u \in \mathbb{R}^k$ is one of the row vectors corresponding to user u in U, and $u_{uf} \in \mathbb{R}$ is the f-th entry in the vector u_u. There are similar relationships between other parameters, though they are not repeated here to conserve space.

PITF is a special case of the CD model with $2{\cdot}k$ dimensionality (see Fig. 4) where:

$$u^{CD}_{uf} = \begin{cases} u_{uf}, & \text{if } f \le k \\ 1, & \text{else} \end{cases} \quad i^{CD}_{if} = \begin{cases} 1 & \text{if } f \le k \\ i_{if-k}, & \text{else} \end{cases}$$
$$t^{CD}_{uf} = \begin{cases} t^U_{tf}, & \text{if } f \le k \\ t^I_{tf-k}, & \text{else} \end{cases} \tag{10}$$

As a special case of the CD model, the PITF model also has a much better runtime complexity than the TD model and is in $O(k)$ for predicting a triple (u, i, t). Furthermore, due to taking into consideration the two pairs of interactions between user, tag and item, the PITF model demonstrates better performance than the TD and CD models in tag recommendation. However, PITF is derived from the combination of personalized tag recommendation and Bayesian personalized ranking optimization criteria. The necessary conditions for such a combination are not available in the scenario of CTR prediction. As for the CTR prediction problem, our purpose is to fill in the missing values in the tensor instead of only ranking scores. Thus, no part of the estimated score can be discarded, as otherwise the estimated CTR score will deviate significantly from the real value. Then, the second necessary condition for PITF is unsatisfied. Alternately, we need to estimate the score for triples (u, p, a) where pairs (u, p) (equivalent to (u, i) in PITF) are not always identical in different instances. Thus, the second necessary condition for BPR mentioned above is also unsatisfied. In spite of having state-of-the-art performance, it is impossible to straightforwardly apply it to predict ad CTR in RTB. Therefore, we modify it into the form of FCTF, which can be used to address the CTR prediction issue based on any type of optimization method, including a simple quadratic loss function optimization algorithm.

4.3. Fully coupled interaction tensor factorization model

Inspired by the PITF model, we take into account the fully coupled interactions between three entities, the user, publisher and advertiser, instead of only one trilateral interaction as the TD and CD models do and call this model the fully coupled interaction tensor factorization (FCTF) model. As seen from Eq. (11), the FCTF model trains two different factor matrixes for each entity, and each factor matrix represents that entity's interaction with one of two other objects. Therefore, each of the pair interactions achieves a better match when unnecessarily satisfying the trade-off condition as TD or CD demands. For example, for user u, two factors, $u^A \in \mathbb{R}^k$ and $u^P \in \mathbb{R}^k$, are learned; u^A is the representation of user u for interaction with ad A, and u^P is another representation for interaction with publisher p. Therefore, the FCTF model is able to explicitly learn fully coupled interactions between three entities, which is difficult to accomplish via the TD or CD model.

FCTF models the three pairs of interactions between users, publishers and ads: users-publishers, publishers-ads and users-ads. Accordingly, $\hat{r}_{u,p,a}$ is estimated as:

$$\hat{r}_{u,p,a} = \sum_f u^P_{uf} \cdot p^U_{pf} + \sum_f p^A_{pf} \cdot a^P_{af} + \sum_f u^A_{uf} \cdot a^U_{af} \tag{11}$$

with the following model parameters to be learned:

$$U^A \in \mathbb{R}^{|U|\times k}, U^P \in \mathbb{R}^{|U|\times k}, \quad A^U \in \mathbb{R}^{|A|\times k}, \quad A^P \in \mathbb{R}^{|A|\times k}, \quad P^U \in \mathbb{R}^{|P|\times k}, \quad P^A \in \mathbb{R}^{|P|\times k}$$

Like PITF, FCTF is another special case of the CD model with a dimensionality of $3{\cdot}k$ (see Fig. 4) where:

$$u^{CD}_{uf} = \begin{cases} u^P_{uf}, & \text{if } f \le k \\ u^A_{uf-k}, & \text{if } k < f \le 2k \\ 1, & \text{else} \end{cases}$$
$$p^{CD}_{pf} = \begin{cases} p^U_{pf}, & \text{if } f \le k \\ 1, & \text{if } k < f \le 2k \\ p^A_{pf-2k}, & \text{else} \end{cases} \tag{12}$$
$$a^{CD}_{af} = \begin{cases} 1, & \text{if } f \le k \\ a^U_{af-k}, & \text{if } k < f \le 2k \\ a^P_{af-2k}, & \text{else} \end{cases}$$

Fig. 4 illustrates the relation between TD, CD, PITF and FCTF. Eq. (11) has another equivalent form:

$$\hat{r}_{u,p,a} = u_u^P \cdot \left(p_p^U\right)^T + p_p^A \cdot \left(a_a^P\right)^T + u_u^A \cdot \left(a_a^U\right)^T \tag{13}$$

where $u_u^P, p_p^U, p_p^A, a_a^P, u_u^A,$ and $a_a^U \in \mathbb{R}^k$ are latent factors corresponding to different entities interacting with other objects, and k is the factor number. Obviously, each of the addends in Eq. (13) is a dot product of two vectors, both with length k. Thus, the FCTF model also takes linear time $O(k)$ to predict $\hat{r}_{u,p,a}$ for a triple (u, p, a). This makes it feasible for real-time CTR prediction in RTB.

Due to its easy implementation, we also use a stochastic gradient descent algorithm to learn the model parameters. The gradients for the FCTF model are:

$$\begin{aligned} &\frac{\partial \hat{r}_{u,p,a}}{\partial u_{u,f}^P} = p_{p,f}^U, \frac{\partial \hat{r}_{u,p,a}}{\partial p_{p,f}^U} = u_{u,f}^P, \frac{\partial \hat{r}_{u,p,a}}{\partial p_{p,f}^A} = a_{a,f}^P \\ &\frac{\partial \hat{r}_{u,p,a}}{\partial a_{a,f}^P} = p_{p,f}^A, \frac{\partial \hat{r}_{u,p,a}}{\partial u_{u,f}^A} = a_{a,f}^U, \frac{\partial \hat{r}_{u,p,a}}{\partial a_{a,f}^U} = u_{u,f}^A \end{aligned} \tag{14}$$

Different from PITF, FCTF is independent from any BPR optimization criterion, and thus it fits the generic third-order tensor factorization task and can be trained by least square error optimization or other popular optimization algorithms, which are easy to implement. FCTF captures enhanced relations between objects, and the results of our experiments, given below, also confirm this argument.

5. Incorporating side-information with FCTF

5.1. Feature-based FCTF model

Ad CTR prediction suffers from sparse data more seriously than the recommendation problem (Lü et al. 2012) or sponsored search advertising do and also encounters the cold start problem whereby CTR estimation becomes more difficult for new ads or new users. Fortunately, there are many types of side-information available regarding users, publishers and ads. For example, a user often has tags and location (i.e., region and city) features. In addition to the URL, there are domain, slot width and height features related to the publishing context. As for the ad, the advertiser, campaign features, and even text description of the ad are usually supplied to the DSP. We incorporate this side-information into the FCTF model to smooth the estimation value with more collaborative information and thereby effectively alleviate the impact of data sparsity on the final performance. When a user has no direct historical click data to be taken for reference, CTR estimation can be achieved based on the user's other features, such as tags or location. We call the FCTF model that incorporates side-information the feature-based FCTF model. Similar ideas have been successfully applied in matrix factorization for recommender systems (Koren et al. 2009; Chen et al. 2012) to cope with similar challenges.

In practise, we divide all information in the dataset into two categories. One consists of single-value attributes, such as the region or city, etc., and one user has at most one feature value in each corresponding value field. The other type of information is multiple-value attributes, such as user tags, where one user may have dozens of them or even none. It is good practice to normalize multi-value attributes.

Taking the representation of a user u for example, it is supposed that the user $\underline{u}$ has only one multi-value feature tag and several single-value features. Let the notation $T(u)$ denote the set of tags of the user u and $C(u)$ denote a set of other single-value attributes; then, the user u can be represented via Eq. (15):

$$u_u = |T(u)|^{-0.5} \sum_{i \in T(u)} t_i + \sum_{c \in C(u)} u_c \tag{15}$$

where $t_i \in \mathbb{R}^f$ is the latent factor for tag i and $u_c \in \mathbb{R}^f$ is the latent factor for attribute c. $|T(u)|^{-0.5}$ is the normalization coefficient for multi-value feature tags.

Similarly, a publisher p and an ad a both receive similar treatment. $C(p)$ is used to characterize single-value attributes of the publisher p, and $C(a)$ is for the ad a.

Thus, the feature-based FCTF model is as follows:

$$\hat{r}_{u,p,a} = \sum_f u_{u,f}^P \cdot p_{p,f}^U + \sum_f p_{p,f}^A \cdot a_{a,f}^P + \sum_f u_{u,f}^A \cdot a_{a,f}^U \tag{16}$$

where $u_{u,f}^P, p_{p,f}^U, p_{p,f}^A, a_{a,f}^P, u_{u,f}^A,$ and $a_{a,f}^U \in \mathbb{R}$ are the f-th elements of $u_u^P, p_p^U, p_p^A, a_a^P, u_u^A,$ and $a_a^U \in \mathbb{R}^f$, respectively, which are defined as:

$$\begin{aligned} u_u^P &= |T(u)|^{-0.5} \sum_{i \in T(u)} t_i^P + \sum_{c \in C(u)} u_c^P, p_p^U = \sum_{c \in C(p)} p_c^U, \\ p_p^A &= \sum_{c \in C(p)} p_c^A, a_a^P = \sum_{c \in C(a)} a_c^P, \\ u_u^A &= |T(u)|^{-0.5} \sum_{i \in T(u)} t_i^A + \sum_{c \in C(u)} u_c^A, a_a^U = \sum_{c \in C(a)} a_{c.}^U \end{aligned} \tag{17}$$

Note that $t_i^P \in \mathbb{R}^f$ and $t_i^A \in \mathbb{R}^f$ are two different latent factors of tag i, respectively corresponding to interactions with the publisher p and the ad a, and the same applies for the other two pair factors.

5.2. Adding biases

One benefit of the factorization model is its flexibility in dealing with various data aspects. However, much of the observed variation in click events is due to effects associated with users, publishers or ads, known as biases or intercepts, independent of any interactions (Koren et al. 2009). For example, some users exhibit a higher tendency to click ads than other users, and some ads also receive more clicks than other ads. Therefore, a first-order approximation of the bias $\hat{b}$ involved in $\hat{r}_{u,p,a}$ is presented as Eq. (18):

$$b = \frac{\sum_{t \in T(u)} b_t}{|T(u)|} + \sum_{c \in C(u) \bigcup C(p) \bigcup C(a)} b_c \tag{18}$$

The notation b_t involved in b indicates the observed deviations of the tag t, and b_c indicates the observed deviations of the feature c that the user u, publisher p or ad a possess. The final estimation formulation with the bias extent is presented as Eq. (19):

$$\begin{aligned} \hat{r}_{u,p,a} = &\sum_f u_{u,f}^P \cdot p_{p,f}^U + \sum_f p_{p,f}^A \cdot a_{a,f}^P + \sum_f u_{u,f}^A \cdot a_{a,f}^U + \frac{\sum_{t \in T(u)} b_t}{|T(u)|} \\ &+ \sum_{c \in C(u) \bigcup C(p) \bigcup C(a)} b_c \end{aligned} \tag{19}$$

The parameters are learned by minimizing the squared error function of the training dataset with a stochastic gradient-descent algorithm. The gradients for the feature-based FCTF model are:

$$\begin{aligned} &\frac{\partial \hat{r}_{u,p,a}}{\partial t_{i,f}^P} = |T(u)|^{-0.5} \sum_{c \in C(p)} p_{c,f}^U, \frac{\partial \hat{r}_{u,p,a}}{\partial u_{c,f}^P} = \sum_{c \in C(p)} p_{c,f}^U, \\ &\frac{\partial \hat{r}_{u,p,a}}{\partial p_{c,f}^U} = |T(u)|^{-0.5} \sum_{i \in T(u)} t_{i,f}^P + \sum_{c \in C(u)} u_{c,f}^P, \\ &\frac{\partial \hat{r}_{u,p,a}}{\partial p_{c,f}^A} = \sum_{c \in C(a)} a_{c,f}^P, \frac{\partial \hat{r}_{u,p,a}}{\partial a_{c,f}^P} = \sum_{c \in C(p)} p_{c,f}^A, \\ &\frac{\partial \hat{r}_{u,p,a}}{\partial t_{i,f}^A} = |T(u)|^{-0.5} \sum_{c \in C(a)} a_{c,f}^U, \frac{\partial \hat{r}_{u,p,a}}{\partial u_{c,f}^A} = \sum_{c \in C(a)} a_{c,f}^U, \\ &\frac{\partial \hat{r}_{u,p,a}}{\partial a_{c,f}^U} = |T(u)|^{-0.5} \sum_{i \in T(u)} t_{i,f}^A + \sum_{c \in C(u)} u_{c,f}^A. \end{aligned} \tag{20}$$

To update the model, we use the following update rules to perform stochastic gradient descent training, where λ is the regularization coefficient, α is the learning rate and $e_{u,p,a} = r_{u,p,a} - \hat{r}_{u,p,a}$.

$$t^P_{i,f} += \alpha\left(e_{u,p,a}\cdot\frac{\partial\hat{r}_{u,p,a}}{\partial t^U_{i,f}} - \lambda\cdot t^P_{i,f}\right), u^P_{c,f} += \alpha\left(e_{u,p,a}\cdot\frac{\partial\hat{r}_{u,p,a}}{\partial u^P_{c,f}} - \lambda\cdot u^P_{c,f}\right)$$

$$p^U_{c,f} += \alpha\left(e_{u,p,a}\cdot\frac{\partial\hat{r}_{u,p,a}}{\partial p^U_{c,f}} - \lambda\cdot p^U_{c,f}\right), p^A_{c,f} += \alpha\left(e_{u,p,a}\cdot\frac{\partial\hat{r}_{u,p,a}}{\partial p^A_{c,f}} - \lambda\cdot p^A_{c,f}\right)$$

$$a^P_{c,f} += \alpha\left(e_{u,p,a}\cdot\frac{\partial\hat{r}_{u,p,a}}{\partial a^P_{c,f}} - \lambda\cdot a^P_{c,f}\right), t^A_{i,f} += \alpha\left(e_{u,p,a}\cdot\frac{\partial\hat{r}_{u,p,a}}{\partial t^A_{i,f}} - \lambda\cdot t^A_{i,f}\right)$$

$$u^A_{c,f} += \alpha\left(e_{u,p,a}\cdot\frac{\partial\hat{r}_{u,p,a}}{\partial u^A_{c,f}} - \lambda\cdot u^A_{c,f}\right), a^U_{c,f} += \alpha\left(e_{u,p,a}\cdot\frac{\partial\hat{r}_{u,p,a}}{\partial a^U_{c,f}} - \lambda\cdot a^U_{c,f}\right) \quad (21)$$

6. Experimental evaluation

6.1. Datasets

We used three season datasets of a global bidding algorithm competition released by the DSP company iPinYou (Liao et al. 2014) in 2014 to evaluate our proposed method. Each season dataset contains impression, click, and conversion logs collected from several advertisers during various days and is divided into two parts, a training dataset and test dataset. For each season dataset, we split the training dataset into two parts according to the impression date and use the last two or three days of data as a validation dataset to train model hyperparameters, such as the learning rate and regularization coefficient. For example, the training dataset of season 2 contains historical bidding logs collected from 5 advertisers during the seven days from June 6th to 12th, and a dataset collected from the following three days from June 13th to 15th is used for offline testing purposes. We extracted the last two days of data (from June 11th to June 12th) from the training dataset as a validation dataset. We present the corresponding date for each season's data in Table 1.

The number of impressions (samples), clicks and statistical CTR results respectively corresponding to each dataset or advertiser key are given in Tables 1–3. The advertiser keys and their industrial categories are listed in Table 4 (Liao et al. 2014).

Note that, in season 1, there is no advertiser ID column, but the landing page URL can be used as the key to distinguish different advertisers (Liao et al. 2014). From these tables, we can see that all CTRs, either on each dataset or on each advertiser, are less than 0.1% except for advertiser 2997 (0.444%). The average CTR for desktop display advertising in practice is usually approximately 0.1%, which is far less than that in MovieLens (approximately 4.5%) or Netflix (approximately 1.2%) (Lü et al. 2012). However, advertiser 2997 is a mobile e-commerce app install related to the mobile environment (see Table 4), where an increased number of inadvertent clicks are easily generated by fat fingers due to the limited space of touchscreens. The detailed log data format and the dimensionality of major features are shown in Tables 5 and 6.

Table 1
Characteristics for three season datasets.

Season	Dataset	Date	Impressions	Clicks	CTR (%)
1	Training	May 11–May 17	9,262,861	7482	0.076
	Test	May 18–May 20	2,594,386	8934	0.075
2	Training	June 6–June 12	12,237,229	8961	0.073
	Test	June 3–June 15	2,524,630	1873	0.074
3	Training	October 19–October 27	3,158,171	2709	0.086
	Test	October 21–October 28	1,579,086	1120	0.071

Generally, each record contains four types of information: user features (iPinYou ID, user-agent, region, city, etc.), publisher features (ad slot ID, slot width, slot height, domain etc.), ad features (creative ID, advertiser ID, landing page URL, etc.) and other features regarding the auction (ad exchange, bidding price, paying price, etc.). Features related to the auction are usually exploited for real-time bidding strategies or bid optimization research (Zhang et al. 2014; Wu et al. 2015; Zhang et al. 2014; Zhang and Wang 2015). We discarded these features regarding the auction when estimating CTR.

6.2. Experimental setup

To verify the effectiveness of our approach, we used the feature-based Tucker decomposition model and feature-based canonical decomposition model as the baselines, which estimated the CTR via Eqs. (22) and (23), respectively

$$\hat{r}^{TD}_{u,p,a} = \sum_{l}^{k_u}\sum_{m}^{k_p}\sum_{n}^{k_a} c_{l,m,n}\cdot u_{u,l}\cdot p_{p,m}\cdot a_{a,n} + b \quad (22)$$

$$\hat{r}^{CD}_{u,p,a} = \sum_{f}^{k} u_{u,f}\cdot p_{p,f}\cdot a_{a,f} + b \quad (23)$$

where the bias b is defined by equation (18). The parameters $u_{u,f}, p_{p,f}$ and $a_{a,f}$ are the f-th elements of $u_u \in \mathbb{R}^f, p_p \in \mathbb{R}^f$ and $a_a \in \mathbb{R}^f$, respectively, which are defined as:

$$u_u = |T(u)|^{-0.5}\sum_{i\in T(u)} t_i + \sum_{c\in C(u)} u_c, p_p = \sum_{c\in C(p)} p_c, a_a = \sum_{c\in C(a)} a_c \quad (24)$$

To update the model, we conducted stochastic gradient descent training. The learning rate was set to 0.00001, and the regularization coefficient to 0.001. The model parameters were initialized with small random values drawn from the normal distribution $N(0,0.0001)$. We also designed experiments to investigate the impact of the number of latent factors on the final prediction quality.

Furthermore, we also implemented the logistic regression-based method and used its performance as a reference. The experimental results of the gradient boosting regression tree (GBRT) reported in Zhang et al. (2014) are also directly presented as reference.

6.3. Experimental results and discussions

We employed the area under the ROC curve (AUC) Fawcett (2004) and the root mean square error (RMSE) to compare the prediction quality of our model with that of the baselines. AUC is a widely used metric for testing the quality of ad CTR prediction (Wu et al. 2012; Yan et al. 2014; Zhang et al. 2014; Wu et al. 2015; Graepel et al. 2010; Oentaryo et al. 2014), and we implemented algorithm 3 from Fawcett (2004) to calculate AUC. RMSE is also chosen as the evaluation measure in the final prediction quality comparison as it is widely used in various regression tasks.

6.3.1. Impact of the dimension of latent factors

Firstly, we investigated the influence of the dimension of latent factors on the prediction quality. Similar experimental results obtained on three different datasets indicate that, with the increase of the dimension of latent factors, the prediction quality increased

Table 2
Training dataset statistics.

Season	Advertiser key	Imps	Clicks	CTR (%)
1	9f4e2f16b6873a7eb504df6f61b24044	3,251,782	3055	0.094
1	3a7eb50444df6f61b2409f4e2f16b687	3,182,633	2644	0.083
1	df6f61b2409f4e2f16b6873a7eb50444	2,828,446	1303	0.046
2	1458	3,083,056	2454	0.080
2	3358	1,742,104	1358	0.078
2	3386	2,847,802	2076	0.073
2	3427	2,593,765	1926	0.074
2	3476	1,970,360	1027	0.052
3	2259	835,556	280	0.034
3	2261	687,617	207	0.030
3	2821	1,322,561	843	0.064
3	2997	312,437	1386	0.444
Total	12	24,658,119	18,559	0.075

Table 3
Test dataset statistics.

Season	Advertiser Key	Imps	Clicks	CTR (%)
1	9f4e2f16b6873a7eb504df6f61b24044	896,908	850	0.095
1	3a7eb50444df6f61b2409f4e2f16b687	918,846	679	0.074
1	df6f61b2409f4e2f16b6873a7eb50444	778,632	403	0.052
2	1458	614,638	543	0.088
2	3358	300,928	339	0.113
2	3386	542,421	496	0.091
2	3427	536,795	395	0.074
2	3476	523,848	302	0.058
3	2259	417,179	131	0.031
3	2261	343,862	97	0.028
3	2821	661,964	394	0.060
3	2997	153,063	533	0.348
Total	12	6,689,084	5162	0.077

Table 4
Advertiser category. The advertiser key is the landing page URL for season 1 and the advertiser ID for seasons 2 and 3.

Advertiser key	Season	Industrial category
df6f61b2409f4e2f16b6873a7eb50444	1	Consumer packaged Goods (CPG)
3a7eb50444df6f61b2409f4e2f16b687	1	Chinese vertical e-commerce
9f4e2f16b6873a7eb504df6f61b24044	1	Vertical online media
1458	2	Chinese vertical e-commerce
3358	2	Software
386	2	International e-commerce
3427	2	Oil
3476	2	Tire
2259	3	Milk powder
2261	3	Telecom
2821	3	Footwear
2997	3	Mobile e-commerce app install

Table 6
The log data format. Columns with * contain data that is hashed or modified before the log is released. Columns with † are only available in season 2 and season 3, not in season 1.

Col #	Description	Example
*1	Bid ID	015300008a77e7ac18823f5a4f5121
2	Timestamp	20130218001203638
3	Log type	1
*4	iPinYou ID	35605620124122340227135
5	User-Agent	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
*6	IP	118.81.189*
7	Region ID	15
8	City ID	16
9	Ad Exchange	2
*10	Domain	e80f4ec7f5bfbc9ca416a8c01cd1a049
*11	URL	hz55b000008e5a94ac18823d6f275121
12	Anonymous URL	null
13	Ad Slot ID	2,147,689 8,764,813
14	Ad Slot Width	300
15	Ad Slot Height	250
16	Ad Slot Visibility	SecondView
17	Ad Slot Format	Fixed
18	Ad Slot Floor Price	0
19	Creative ID	e39e178ffdf366606f8cab791ee56bcd
*20	Bidding Price	753
*21	Paying Price	15
*22	Landing page URL	a8be178ffdf366606f8cab791ee56bcd
†23	Advertiser ID	2345
*†24	User Profile IDs	123,5678,3456

Table 5
Dimensionality of major features for three season datasets.

Season	Dataset	Users	Tags	Slots	URLs	Advertisers	Campaigns	Creatives
1	Training	6,799,908	Null	124,684	2,082,249	1	3	32
	Test	2,164,525		58,945	811,585	1	3	33
2	Training	10,146,491	45	141,515	2,362,123	5	18	74
	Test	2,310,303	68	48,458	663,218	5	18	74
3	Training	2,818,424	69	53,518	963,576	4	4	57
	Test	1,490,321	58	43,603	552,694	4	4	54

dramatically in the beginning and then slowly reached a relatively steady value when the dimension was close to 8. Here, we take the experimental results on the season 2 dataset as an example. Fig. 5 presents the results for season 2; the horizontal axis is the dimension of factors from 2 to 64, and the legends are the size of the training data from one day's worth to seven days' worth. As shown in the figure, when the dimension was varied from 2 to 64, our model showed a similar improvement in performance on diverse sizes of training data. When the dimension of factors increased to 8, the prediction quality was relatively stable on all sizes of training data. In the inference algorithm, an appropriate choice of this parameter from 8 to 64 could achieve an optimal balance between better prediction quality and less training time. The improvement of 64 dimensions over 32 dimensions is nearly negligible. To achieve a better and more stable prediction quality, the dimension should not be less than 32 (denoted as FCTF 32).

6.3.2. Learning runtime

In this section, we also take the experimental results obtained on season 2 as an example to explain the difference of learning runtime between different models. Fig. 6 presents the comparison of the convergence of feature-based FCTF with other tensor factorization-based models on season 2. After a model was trained over a span of 100 min, improvement of the prediction quality occurred. FCTF 32 and CD 32 models converged much faster than TD 32. The CD model achieved convergence after only 30 min of training, while FCTF needed more training time, 60 min, to converge. The reason is that the FCTF model needs to update twice the number of parameters of the CD model.

In contrast, as shown in Fig. 7, the TD 32 model needed at least 100 h to converge. It needed more than 50 h to achieve a prediction quality as good as that of CD 32. Even after 150 h of training, the quality of TD was still worse than that of FCTF 32. This worse empirical runtime result of the TD model in comparison to CD and FCTF matches the theoretical runtime complexity analysis of the model equations (Rendle and Schmidt-Thieme 2010).

6.3.3. Prediction quality

Finally, we compared the quality of the FCTF factorization model to the baselines in terms of AUC and RMSE. Table 7 and Table 8 respectively show the AUC and RMSE prediction quality of different methods on the three season datasets. Note that the experimental results of GBRT are directly referenced from Zhang et al. (2014) for comparison and that there are no results for GBRT on the 1st season dataset because they did not give them in the literature. They also presented the experimental results of their LR model, which are slightly poorer than ours. Because of the huge imbalance of click/unclick instances, the empirically best regression model usually predicted the CTR as being very close to zero. This results in the RMSE having quite a small value for all models, and the improvement in RMSE is much slighter compared with AUC (Zhang et al. 2014).

From the experimental results, we can see that different advertisers have significant differences in terms of AUC or RMSE due to the different user behaviors in different contexts. For example, although advertiser 2997 has the highest overall CTR (0.444%) in the historical impression log, that advertiser also has both the lowest observation number (see Table 3) and more noise in the historical impression log due to the "fat finger" effect in the mobile environment substantially increasing the difficulty of predicting CTR. Therefore, all models present poorer prediction performance for advertiser 2997 than for the other advertisers.

We conducted a separate experiment on the second season dataset in which only the tag feature was applied to represent users, and the other conditions were not changed. This experimental result shows that the AUC value was close to 0.9, which indicates that the tag feature of users in season 2 shows very strong informative capability in addressing this problem. As there is no tag information available in season 1, the prediction quality in it decreased significantly compared to that in season 2 and season 3 where an appropriate number of tags can be exploited. In addition, as shown in Tables 7 and 8, all models achieved much better performance on advertisers in season 2 than in season 3, although they have the similar feature structures. iPinYou technicians explained that this is due to the different user segmentation systems between season 2 and season 3 (Zhang et al. 2014).

From the point of view of different methods, the overall performance of the logistic regression model was lower than that of all factorization models; only the CD model was competitive, as it slightly outperformed the LR.

Our model outperformed the baseline approaches. In particular, the FCTF model outperformed both TD and CD in terms of total AUC or RMSE on different datasets. As described in Section 4.3, both the TD and CD models obtain only one latent factor for each entity simultaneously interacting with two other entities, so this factor is not the optimal factor interacting with any entity of the other two. However, FCTF takes into account both optimal interaction scores of each entity involving the two other sides and therefore it could exquisitely capture the underlying pairwise relationships between users, publishers and ads that the other

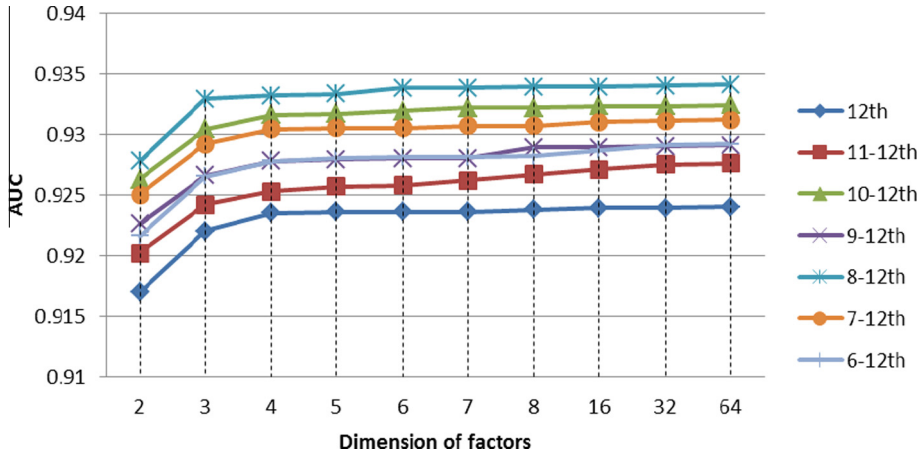

Fig. 5. Impact of the dimension of latent factors on the prediction quality on season 2.

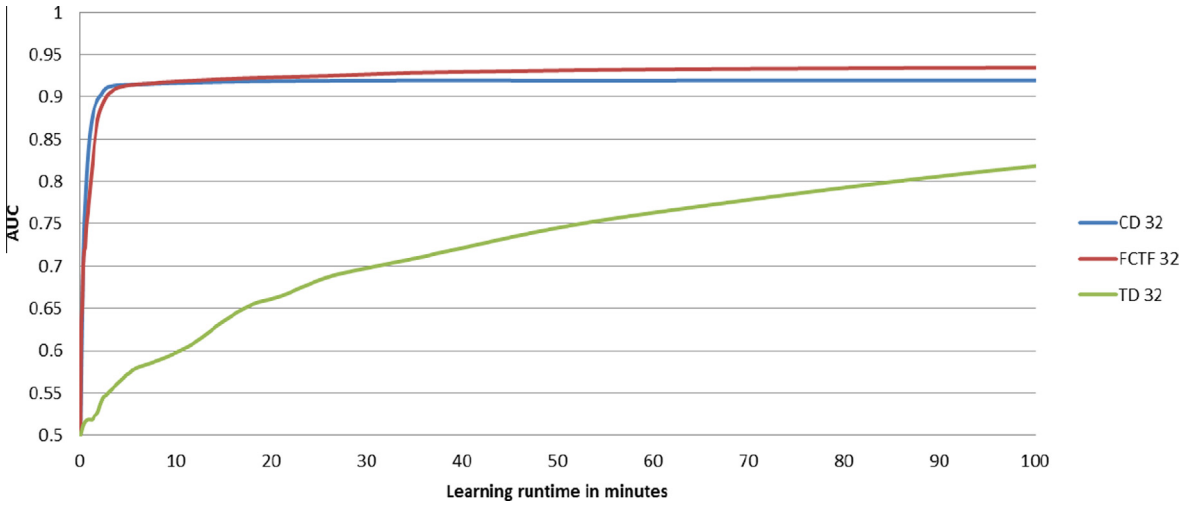

Fig. 6. AUC-Measure after training a model for x minutes on the second season dataset. The FCTF and TC models already give good prediction quality after 60 and 30 min, respectively.

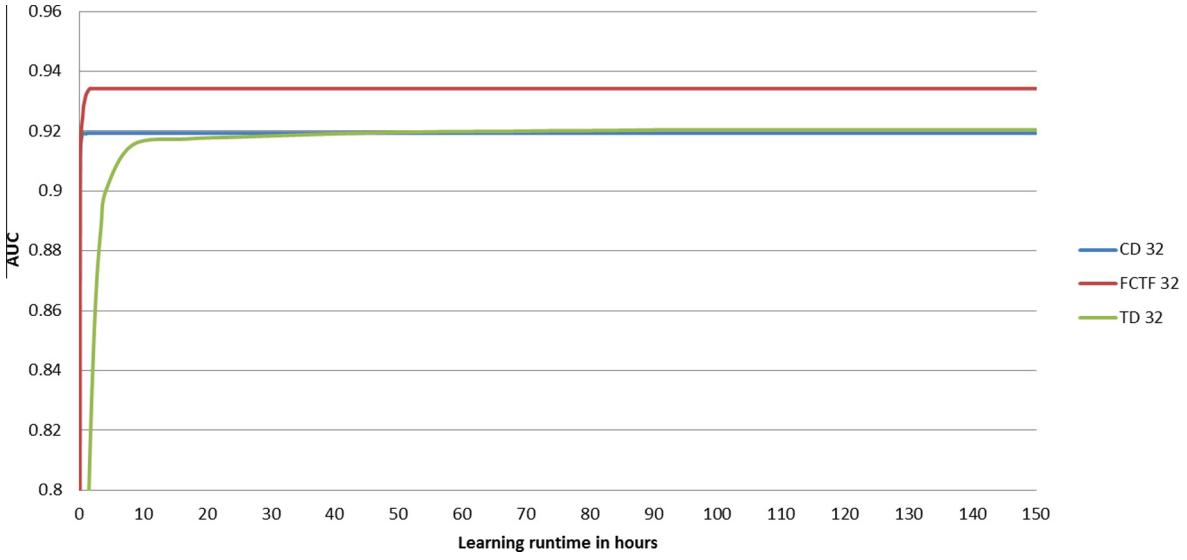

Fig. 7. AUC-Measure after training the TD model for *x* hours on the second season dataset. Learning a high-quality TD model takes several days.

Table 7
CTR estimation performance in terms of AUC.

		AUC				
Season	Advertiser key	LR (%)	GBRT (%)	TD 4 (%)	CD 4 (%)	FCTF 4 (%)
1	9f4e2f16b6873a7eb504df6f61b24044	73.47	/	74.75	74.36	75.79
1	df6f61b2409f4e2f16b6873a7eb50444	72.60	/	73.86	73.69	75.73
1	3a7eb50444df6f61b2409f4e2f16b687	69.04	/	68.66	67.36	69.99
2	1458	97.93	97.07	97.73	97.94	98.18
2	3358	96.80	97.22	96.88	97.63	98.31
2	3386	78.48	76.86	75.50	74.79	76.22
2	3427	97.17	93.42	96.18	96.15	96.54
2	3476	92.04	94.22	93.82	92.69	94.29
3	2259	72.34	67.91	69.16	71.80	72.61
3	2261	65.21	57.39	61.26	63.81	64.99
3	2821	67.02	58.20	66.79	66.85	68.63
3	2997	53.30	59.79	56.20	52.29	53.02
1	Total	72.43	/	73.20	73.04	74.47
2	Total	91.83	92.00	92.08	91.94	93.31
3	Total	76.51	77.15	77.36	77.74	78.95

Table 8
CTR estimation performance in terms of RMSE.

		RMSE				
Season	Advertiser key	LR	GBRT	TD 4	CD 4	FCTF 4
1	9f4e2f16b6873a7eb504df6f61b24044	0.0301	/	0.0305	0.0304	0.0306
1	df6f61b2409f4e2f16b6873a7eb50444	0.0217	/	0.0225	0.0225	0.0228
1	3a7eb50444df6f61b2409f4e2f16b687	0.0263	/	0.0270	0.0271	0.0270
2	1458	0.0195	0.0263	0.0235	0.0223	0.0221
2	3358	0.0308	0.0279	0.0268	0.0271	0.0292
2	3386	0.0328	0.0285	0.0271	0.0291	0.0315
2	3427	0.0237	0.0245	0.0237	0.0258	0.0317
2	3476	0.0256	0.0231	0.2165	0.0236	0.0263
3	2259	0.0169	0.0176	0.0175	0.0168	0.0167
3	2261	0.0160	0.0167	0.0163	0.0159	0.0158
3	2821	0.0226	0.0238	0.0229	0.0224	0.0223
3	2997	0.0615	0.0581	0.0601	0.0612	0.0601
1	Total	0.0274	/	0.0235	0.0275	0.0272
2	Total	0.0262	0.0260	0.026	0.0262	0.0256
3	Total	0.0268	0.0268	0.0267	0.0266	0.0264

two methods could not catch. The experimental results demonstrate that the FCTF model has a superior ability to address this problem, as we expected.

7. Conclusions and future work

In this paper, we examine the ad CTR prediction problem in RTB for DSPs. We address the issue by presenting the fully coupled interaction model (FCTF) based on the Tucker decomposition (TD) model. The FCTF model has three major advantages for addressing this problem. Firstly, FCTF is a third-order tensor factorization model that can effectively capture complicated ternary relations between three players: the user, publisher and advertiser in RTB. Instead of training only one latent matrix for each object, FCTF learns two different latent matrices, each of which is trained by interacting with each of another two objects. Therefore, without the loss of attempting to compromise as in TD or CD, FCTF achieves better performance than other TD-based models such as TD and CD in real-world datasets. Furthermore, our model also shows relatively stable performance for both diverse numbers of factors and different sizes of the training dataset. Secondly, the runtime complexity of the FCTF equation is linear for the number of factorization dimensions, which makes it feasible for high dimensions even when fulfilling real-time tasks. Lastly, compared with PITF, which is accompanied by the BPR algorithm and a specific data interpretation, FCTF is suitable for generic third-order tensor factorization tasks regardless of what optimization strategy is applied, including simple methods such as a quadratic loss function or logarithmic loss function.

Moreover, we incorporate all types of side information related to multi-aspects into the FCTF model to represent users, publishers and ads. This approach enormously alleviates the issue of the sparsity of training data and simultaneously overcomes the cold-start problem to a certain extent.

In future work, to relieve the impact of the imbalance of positive and negative samples in the training dataset on the final result, we will attempt to apply a pairwise learning algorithm to train FCTF to optimize the AUC directly. We also want to study how the preferences of users change over time and find the proper size for the training dataset.

Acknowledgments

This work is supported by projects of the National Natural Science Foundation of China (No. 61300114 and No. 61572151), the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20132302120047) and the China Postdoctoral Science Special Foundation (No. 2014T70340).

References

Agarwal, D., Chen, B.-C., Elango, P. Spatio-temporal models for estimating click-through rate, In WWW '09, New York, NY, USA, 2009. ACM, 21–30.
Agarwal, D., Agrawal, R., Khanna, R., Kota, N. Estimating rates of rare events with multiple hierarchies through scalable log-linear models, In KDD '10, New York, NY, USA, 2010. ACM, 213–222.
Carroll, J. Douglas, Chang, Jih-Jie, 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. Psychometrika 35, 283–319.
Chapelle, O., Manavoglu, E., Rosales, R., 2014. Simple and scalable response prediction for display advertising. ACM Trans. Intell. Syst. Technol. 5 (4), 61.
Chen, T., Tang, L., Liu, Q., Yang, D., Xie, S., Cao, X. et al. Combining factorization model and additive forest for collaborative followee recommendation. In KDD CUP, 2012.
De Lathauwer, L., De Moor, B., Vandewalle, J., 2000. A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. 21 (4), 1253–1278.
Fawcett, Tom., 2004. ROC graphs: notes and practical considerations for researchers. Machine Learning 31, 1–38.
Graepel, T., Candela, J.Q., Borchert, T., et al. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine, In Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010, 13–20.
Harshman, Richard A. Foundations of the parafac procedure: models and conditions for an "exploratory" multimodal factor analysis. In UCLA Working Papers in Phonetics, 1970;16, 1–84.
Koren, Yehuda., Bell, Robert., Volinsky, Chris., 2009. Matrix factorization techniques for recommender systems. Computer 8, 30–37.
Lee, Kuang-chih, Burkay Orten, Ali Dasdan, Li Wentong. Estimating conversion rate in display advertising from past performance data. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, 768–776.
Liao, Hairen, Lingxiao Peng, Zhenchuan Liu, Xuehua Shen. iPinYou Global RTB Bidding Algorithm Competition Dataset. In Proceedings of 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, 2014, 1–6.
Lü, Linyuan, Medo, Matúš, Yeung, Chi Ho, Zhang, Yi-Cheng, Zhang, Zi-Ke, 2012. Tao Zhou. Recommender systems. Physics Rep. 519, 1–49.
Menon, Aditya Krishna, Chitrapura, Krishna Prasad, Garg, Sachin, Agarwal, Deepak, Kota, Nagaraj. Response prediction using collaborative filtering with hierarchies and side-information. In Processing of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2011, 141–149.
Oentaryo, R.J., Lim, E.P., Low, J.W., et al. Predicting response in mobile advertising with hierarchical importance-aware factorization machine. In Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM 2014). ACM, 2014, 123–132.
Rendle, Steffen, Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In Proceedings of the Third ACM International Conference on Web Search and Data Mining, 2010, 81–90.
Rendle, Steffen, Christoph Freudenthaler, Zeno Gantner, Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, 2009, 452–461.

Rendle, Steffen, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, 727–736.

Richardson, M., Dominowska, E., Ragno, R., 2007. Predicting clicks: estimating the click-through rate for new ads. In WWW '07, New York, NY, USA. ACM, 521–530.

Shan, Lili, Lin Lei, Shao Di, and Wang Xiaolong. CTR Prediction for DSP with Improved Cube Factorization Model from Historical Bidding Log In Processing of the 21st International Conference on Neural Information Processing, 2014, 17–24.

Shen, S., Hu, B., Chen, W., Yang, Q. Personalized click model through collaborative filtering. In Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM 2012), 2012, 323–332.

Symeonidis, Panagiotis, Alexandros Nanopoulos, and Yannis Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In Proceedings of the 2008 ACM Conference on Recommender Systems, 2008, 43–50.

Tucker, L.R., 1966. Some mathematical notes on three-mode factor analysis. Psychometrika 31 (3), 279–311.

Wang, X., Li, W., Cui, Y., et al. Click-through rate estimation for rare events in online advertising[J]. In Online Multimedia Advertising: Techniques and Technologies, 2010, 1–12.

Wu, Kuan-Wei., Ferng, Chun.-Sung., Ho, C.-H., Liang, A.-C., Huang, C.-H., Shen, W.-Y., et al. A two-stage ensemble of diverse models for advertisement ranking. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012.

Wu, W.C.H., Yeh, M.Y., Chen, M.S. Predicting winning price in real time bidding with censored data. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015, 1305–1314.

Yan, L., Li, W. J., Xue, G. R., and Han, D. Coupled group Lasso for Web-Scale CTR prediction in display advertising. In Proceedings of the 31st International Conference on Machine Learning, 2014 (ICML 2014), 802–810.

Zhang, W., Wang, J. Statistical arbitrage mining for display advertising. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015, 1465–1474.

Zhang, W., Yuan, S., Wang, J., Shen, X. Real-time bidding benchmarking with ipinyou dataset. arXiv preprint arXiv:1407.7073, 2014.

Zhang, W., Yuan, S., Wang, J. Optimal real-time bidding for display advertising, In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2014, 1077–1086.