Combined Regression and Tripletwise Learning for Conversion Rate Prediction in Real-Time Bidding Advertising

Lili Shan
Harbin Institute of Technology
Harbin, China
shanll@insun.hit.edu.cn

Lei Lin
Harbin Institute of Technology
Harbin, China
linl@insun.hit.edu.cn

Chengjie Sun
Harbin Institute of Technology
Harbin, China
cjsun@insun.hit.edu.cn

ABSTRACT

In real-time bidding advertising (RTB), the buyers bid for individual advertisement impressions provided by publishers in real time. The final goal of the buyers is to maximize the return on their investment. To gain higher returns, buyers prefer to first purchase more conversion impressions than click-only ones and then purchase more click-only impressions prior to non-click ones. Simultaneously, to reduce the expense, they need to accurately estimate a reasonable bid price, the predicted precision of which depends on the precision of the predicted conversion rate (CVR) or predicted click-through rate (CTR). Therefore, the predicted CVR or predicted CTR must provide not only good ranking values but also correct regression estimations. This paper is focused on the CVR estimation problem for buy-sides in RTB and a combined regression and tripletwise ranking method (CRT) is proposed that jointly considers regression loss and tripletwise ranking loss to estimate the CVR. This method attempts to rank conversion impressions above click-only ones and simultaneously rank click-only impressions above non-click ones. Meanwhile, through simultaneously utilizing the historical conversion and click information to alleviate sparsity, the CRT method is also aimed to achieve a good two category-ranking performance, as well as a good regression performance for predicting the CVR.

CCS CONCEPTS

• **Information systems → Computational advertising**; **Learning to rank**; • **Computing methodologies → Machine learning approaches**;

KEYWORDS

Ranking algorithm, Computational advertising, Machine learning, Conversion rate prediction

ACM Reference Format:
Lili Shan, Lei Lin, and Chengjie Sun. 2018. Combined Regression and Tripletwise Learning for Conversion Rate Prediction in Real-Time Bidding Advertising. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3209978.3210062

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210062

1 INTRODUCTION

1.1 Real-time Bidding

Real-time bidding (RTB) is becoming an increasingly popular trading means for advertising inventory that is bought and sold through the public auction of individual impressions in real time, i.e., while a consumer is waiting. RTB advertising has become an increasingly important approach for enabling advertisers to promote their advertising performance through more accurately targeting online potential users. In the RTB enabled display advertising ecosystem, publishers (sellers) supply advertising inventory to buyers (advertisers, agencies, etc.) through advertisement exchange systems. Advertisement exchanges aggregate the advertising inventories of multiple publishers and sell them to a number of buyers via a real-time auction of each impression. The display of an advertisement placed on a publisher's site (Web page, application, etc.) is considered an advertisement impression. When a user clicks on a hyperlink to a publisher's Web page, in addition to producing and showing high-quality content to the user, the second main task of the publisher is to sell its advertisement inventory for monetization. If there is an advertisement space that needs to be sold by RTB, the publisher initiates the RTB trading process by sending a bid request to a number of bidders (buyers) through an advertisement exchange or other supply source. The bid request consists of at least advertisement space attributes and user attributes, and optionally additional attributes providing the impression context. When receiving a bid request, the bidders need to use bidding algorithms to instantly decide whether to bid and determine their bidding price, and then return their bidding price to the advertisement exchange in real time. The impression is sold to the highest bidder and the winner's advertisement is placed on the publisher's Website. At the same time, the buyer must pay the seller for this advertisement impression, which completes this RTB trading process.

Advertisers obtain an unequal return from these impressions according to the subsequent responses of users who visit the Web page containing the advertisements. The possible responses are divided into three types, as follows. First, if the user takes no action on the impression, as if the advertisement does not exist, this response is defined as non-click. Second, if the user clicks on the advertisement and reaches the landing page of the advertiser, but takes no further action, this response is defined as click-only. Third, if the user further completes one of the predefined actions on the publisher's Website, such as registration, order placement, purchase, or subscription to an email list, we define it as a conversion response. In general, for performance-based advertising, conversions may bring advertisers multiple times the profits of click-only actions, and click-only events may bring them more profit than non-clicks.

Therefore, during a certain RTB trading period, buyers aim to purchase more conversion impressions first and then more click-only impressions with their fixed budget. To achieve this, buyers must accurately estimate the conversion rate (CVR) or click-through rate (CTR) for each advertisement impression before giving the final bid price, which directly affects the bid acceptance probability and the return on their investment. This paper is focused on the CVR estimation problem for buy-sides in RTB. A combined regression and tripletwise ranking method (CRT) is proposed that jointly examines regression loss and tripletwise ranking loss to estimate the CVR.

1.2 Motivation

The final goal of buyers is to maximize their return on investment (ROI) during an RTB trading period for one or more advertising campaigns, which contains a series of RTB bidding processes for impressions. To gain higher returns when their budget is fixed, buyers prefer to first purchase more conversion impressions than click-only ones and then more click-only impressions prior to non-click ones. Therefore, the predicted CVR (pCVR) or predicted CTR (pCTR) should be able provide ranking of conversion impressions above click ones and click impressions above non-click ones. Meanwhile, to reduce the expense, buyers need to accurately estimate a reasonable bid price, which directly determines the investment quantity and the bidding result. Since pCVR and pCTR both play key roles in buyers' computation of the bid price, it is critical that the actual CVR or CTR estimation is as accurate as possible to enable efficient bid pricing [18]. Moreover, it is important for buyers that the pCVR and pCTR not only yield good ranking values, but also provide good regression estimates [15, 18]. Thus, methods for estimating the CVR and CTR more accurately constitute a key technology for buyer's bidding algorithms. In recent years, CVR estimation has increasingly attracted research attention because of the substantial returns that advertisement conversions yield. Because of the more serious sparsity of historical conversion data for buy-sides, in RTB it is more difficult for buyers to predict the CVR than the CTR. However, historical click information is more plentiful than conversion data. Furthermore, in general impressions with a click event have a greater probability of being converted than those with no click event. Therefore, appropriate utilization of click information would facilitate improvement in the pCVR accuracy.

1.3 Contributions

In this paper, we focus on the CVR estimation problem for buy-sides in RTB trading and propose a combined regression and tripletwise ranking method, CRT. The proposed CRT method simultaneously utilizes historical click information and conversion information via tripletwise learning optimization to alleviate the sparsity of conversion data available for CVR estimation. Furthermore, our method also jointly examines regression and ranking loss while estimating the pCVR. Therefore, CRT attempts to rank conversion impressions above click-only ones and click-only impressions above non-click ones (in terms of the area under the receiver operating characteristics (ROC) curve (AUC) and multi-class AUC), as well as to achieve a good regression-based performance (in terms of the squared error). The key contributions of this paper are as follows.

(1) We propose a tripletwise learning algorithm for three-category ranking. This learning algorithm is aimed to determine the correct order of each pair of conversion and click-only events, as well as the correct order of each pair of click-only and non-click events. Therefore, it simultaneously uses the conversion and click information in the impression history log to alleviate sparsity for predicting the CVR.
(2) We propose a combined regression and tripletwise ranking algorithm, CRT, which jointly examines the regression and ranking while estimating the CVR. The CRT method attempts to rank conversion impressions above click-only ones and click-only impressions above non-click ones (in terms of the AUC and multi-class AUC), as well as to achieve a good regression-based performance in terms of the squared error.
(3) We describe the evaluation of our methods using content datasets. The experimental results show that our approach not only shows a promising performance for binary classification as compared to the baseline and some existing combined methods, but also provides a significant advantage for the three-category ranking task.

This paper is organized as follows. In Section 2, we review existing studies related to ours and discuss their limitations. Then, we present the problem formulation and preliminaries in Section 3. Section 4 presents details of our motivation and approach for combining regression and tripletwise ranking. Finally, we provide the results of experiments on real-world datasets conducted to validate our analysis and test the efficacy of our method.

2 RELATED WORK

2.1 Click-through Rate and Conversion Rate Prediction

In a real-time bidding advertising scenario, it is essential to model the probability of an advertisement being clicked or converted rather than only to predict whether or not it will be clicked or converted [15, 17]. Therefore, most of the existing studies separately addressed the CTR or CVR prediction as a regression problem. Owing to their easy implementation, immediate prediction, acceptance, and performance, logistic regression (LR) or generalized linear models and their variants have been widely applied for CVR or CTR estimation [1, 2, 9, 25] in display advertising, especially in industrial systems [6], while with the increasing complexity of feature engineering other collaborative filtering models, such as matrix factorization (MF) or its variants, which perform well in recommender systems because of their capability of mining the underlying relationships between features, have widely been adopted to cope with response prediction for online advertising [14, 24].

Meanwhile, many algorithms based on the learning-to-rank approach have been introduced to predict the CTR or CVR in contextual advertising, sponsored search advertising, and social media advertising [8, 10, 22, 24]. Wu et al. [24] modeled advertisements' CTR prediction in Track 2 of the KDD Cup 2012 competition as a

recommendation problem solvable by ranking-based matrix factorization [24], which is the best individual model they proposed. The tensor factorization model has also been applied to learn the ternary relation between users, publishers, and advertisements [19, 20], and achieved a better prediction quality than traditional matrix factorization models. However, its complexity is a barrier to its implementation in actual applications.

Most of the above methods handle CTR or CVR prediction only separately and therefore fail to jointly use conversion and click information. To the best of our knowledge, in only a few previous studies [11] has CTR and CVR estimation been combined. Li et al. [11] proposed a dynamic collective matrix factorization (DCMF) approach for conversion prediction, which jointly models the temporal relationships between click events and purchase events in display advertising. However, these methods optimize the regression performance or ranking performance only in isolation. If the methods for tackling the problem are based on regression measures, such as mean square error (MSE) or one-sided linear or square penalty, as the objective function, they are likely to perform suboptimally in terms of a ranking performance measure such as AUC, and vice versa.

2.2 Combined Regression and Ranking

Furthermore, noteworthy studies exist that were focused on combining regression and ranking [3, 15, 18, 23]. However, in all these studies attempts were made to fuse pairwise ranking or pairwise labels with regression, even if their specific implementations for pairwise ranking were diverse. Sculley [18] previously presented a combined regression and ranking (CRR) method that explicitly optimizes the regression and the pairwise ranking objective simultaneously for learning. This method avoids the effort required to combine two types of scores when two separate models are learned, one for ranking and one for regression. The objective of CRR is to provide a strong performance in terms of both regression and ranking metrics. In their method, Menon et al. [15] adopted a semiparametric technique that offers both good ranking and regression performances through optimizing ranking loss, followed by isotonic regression. This method is aimed first to maximize the pairwise ranking performance, and then, in the isotonic regression step, to achieve a low squared error value. Obviously, although this post-processing technique achieves probabilities after ranking loss, only the ranking loss of the problem is optimized when training, and the post-processing step is unable to correct an incorrectly ordered pair caused by the previous ranking. When regression and ranking objectives are optimized simultaneously [18], some incorrectly ordered pairs may be corrected through adjusting their estimation values in order to reduce their regression loss. Furthermore, for historical impression data, only one classification label other than probability value is associated with each sample and this makes it difficult to perform post-processing isotonic regression. In addition, Chen et al. [3] and Wang et al. [23] adopted similar methods of fusing pointwise and pairwise labels for image label prediction, classification, or image retrieval. However, their framework needs massive labeling work to obtain a sufficient number of pairwise labels.

In summary, in all these studies attempts were made to fuse pairwise ranking or pairwise labels with regression rather than to apply tripletwise ranking. Pairwise ranking is aimed to achieve the correct order of only one pair, such as a conversion and non-conversion pair, without examining the correct sequence of the other pair of click-only and non-click. Therefore, these methods are unable to meet the three-category ranking requirement of buyers in RTB transactions. For new application scenarios, we combined regression and tripletwise ranking objectives simultaneously to handle the CVR estimation problem for buyers. Furthermore, we also elaborately designed a novel and suitable training strategy to optimize our new combined loss objective.

3 BACKGROUND

In this paper, we examine two main types of methods that are widely used in response prediction, as described in the related work section: logistic regression and matrix factorization models. First, as required we present the problem formulation based on the matrix factorization model and the related terminology. Second, we present the regression loss function that is widely used for logistic regression and matrix factorization. Finally, the pairwise ranking loss function used in this study as the baseline is introduced.

3.1 Problem Setup and Formulation

An impression history log set is

$$D = \{(d_i, r_i) \,|i = 1, \ldots, n\}$$

where d_i is a triple in the form (u, p, a) and means that advertisement a has been displayed to user u in the publishing context p. This is called an impression. An associated label is $r_i \in \{0, 1\}$. $r_i = 1$ means this impression leads to a conversion; otherwise, it leads to no conversion. The definition of the context depends on the specific application scenario, such as page content in context advertising, advertisement space attributes in display advertising, and user geographical position in mobile advertising.

Our purpose is to learn a prediction function $\hat{r} : U \times P \times A \rightarrow \mathbb{R}$ that is itself parametrized by a set of model parameters Θ. For an arbitrary impression (u, p, a), the output of the function represents the probability or possibility measure that it will lead to a conversion. In the logistic regression model, the prediction function is

$$\hat{r}(\theta, x) = \frac{1}{1 + \exp(-\theta^T x)} \tag{1}$$

where x is the features vector associated with the user, publisher, and advertisement.

In the matrix factorization model, a response matrix $\mathcal{R} \in \mathbb{R}^{|U|\times|A|}$ is constructed based on historical impression data D. The matrix factorization model is aimed to learn latent factors $\mathcal{P} \in \mathbb{R}^{|U|\times f}$ for all users and latent factors $\mathcal{Q} \in \mathbb{R}^{f\times|A|}$ for all advertisements, which are aimed to satisfy the approximation equation

$$\mathcal{R} \approx \mathcal{P} \cdot \mathcal{Q} \tag{2}$$

where f is the dimension of each latent factor. If the latent factor of user i is $p_i \in \mathbb{R}^f$ and that of advertisement j is $q_j \in \mathbb{R}^f$, the prediction function is

$$\hat{r}(i, j) = q_i^T \cdot p_j \tag{3}$$

In order to blend the various features available into a matrix factorization model, we adopted methods similar to those mentioned in previous papers [4, 14] to integrate hierarchies and side information. We call the integration model feature-based matrix factorization. Accordingly, all the features available must be divided into two types, user and advertisement. We blend all the latent factors corresponding to the features associated with user u to represent user u and combine all the latent factors of the features associated with advertisement a to describe advertisement a. Finally, the prediction function for matrix factorization is

$$\hat{r}(u,a) = \left(\sum_{i \in C(u)} u_i \alpha_i\right)^T \cdot \left(\sum_{j \in C(a)} a_j \beta_j\right) + \sum_{k \in C(u) \cup C(a)} b_k \quad (4)$$

where $C(u)$ is a feature set of user u and $C(a)$ is a feature set of advertisement a. α_i and β_j are feature values respectively for user u and advertisement a. u_i and a_j are latent factors respectively associated with feature values α_i and β_j. $b_k \in \mathbb{R}$ is the bias of feature k.

3.2 Regression Loss Functions

In this paper, we focus on the two most frequently used loss functions respectively associated with logistic regression and matrix factorization. In fact, similarly to CRR [18], almost all convex loss functions can be applied in our framework.

3.2.1 Logistic Loss. The logistic loss function is commonly applied in logistic regression, which is frequently used as a classification method for binary classification, but also can be seen as a regression method for predicting the real-value probability scores of a sample belonging to a specific class. The logistic loss function for $r \in [0,1]$ and $\hat{r} \in [0,1]$ is

$$L(\Theta, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} r_i log\hat{r}_i + (1 - r_i) log(1 - \hat{r}_i) \quad (5)$$

3.2.2 Euclidean Distance Loss. The frequently used regression loss function for matrix factorization is Euclidean distance:

$$L(\mathcal{R}, \hat{\mathcal{R}}) = \left\| \mathcal{R} - \hat{\mathcal{R}} \right\|^2 = \sum_{ij} (r_{ij} - \hat{r}_{ij})^2 \quad (6)$$

The loss for a single element $\hat{r}$ as compared with true label r is given by $l(r, \hat{r}) = (r - \hat{r})^2$. Since it is similar to squared loss, it can easily be implemented efficiently in practice. We use it as regression loss for matrix factorization.

3.3 Pairwise Ranking Loss Functions

Various pairwise ranking methods also have been applied for CVR or CTR prediction [18, 24]. We take their performance as baselines with which to compare the performance of our method. The first pairwise loss function applied is associated with the logistic regression prediction function in previous studies [18]. Two arbitrary predicted values $\hat{r}_i$ and $\hat{r}_j$, as compared with their true labels r_i and r_j, are given as

$$l(r_i, r_j, \hat{r}_i, \hat{r}_j) = \Delta r_{ij} \log(\Delta \hat{r}_{ij}) + (1 - \Delta r_{ij}) \log(1 - \Delta \hat{r}_{ij}) \quad (7)$$

where $\Delta r_{ij} = r_i - r_j$ and $\Delta \hat{r}_{ij} = \hat{r}_i - \hat{r}_j$.

The second pairwise loss function is Bayesian personalized ranking (BPR) proposed by Rendle et al. [16]. We adopt it as pairwise ranking loss for matrix factorization. First, the training data D must be divided into two new sets, N^+ for all positive and N^- for all negative samples. We denote by $P = \{(x_i, x_j) | x_i \in N^+, x_j \in N^-\}$ the set of all the impression pairs composed of conversion and non-conversion impressions. According to the BPR-based pairwise optimization function, the task is to minimize

$$\mathrm{L}(\Theta, P) = \sum_{i=1}^{N^+} \sum_{j=1}^{N^-} -\log \sigma\left(\hat{r}_i - \hat{r}_j\right) \quad (8)$$

where σ is the logistic sigmoid $\sigma(x) = \frac{1}{1+\exp(-x)}$

4 COMBINING REGRESSION AND TRIPLETWISE RANKING: CRT

In this section, we present the CRT approach. First, we describe the idea of the tripletwise learning optimization in detail and then elaborate the combination framework and the SGD-based learning algorithm implemented in CRT.

4.1 Tripletwise Ranking

Our motivations for tripletwise ranking are derived from two distinct aspects. First, the final goal of buyers is to maximize their ROI in RTB transactions. In general, in performance-based advertising, conversions can bring advertisers multiple times the profit of click-only actions, and click-only events may bring them more profit than non-clicks. Therefore, given a fixed campaign budget, buyers prefer to purchase more conversion impressions prior to click-only ones and then purchase more click-only impressions than non-click ones. This idea motivated us to consider the CVR prediction problem as a three-category ranking problem, the purpose of which is to rank conversion impressions above click-only ones and non-click impressions below click-only ones. Therefore, we attempted to design a novel learning optimization algorithm to realize three-category ranking. Second, historical impression logs show that conversion events are even sparser than click events. This situation makes it more difficult to accurately estimate CVR than CTR. Therefore, our objective was to use history click information, as well as history conversion information, for predicting CVR to alleviate conversion data sparsity.

On the basis of the above motivations, we propose tripletwise learning optimization derived from BPR based on pairwise learning [16]. The tripletwise learning optimization considers the response prediction problem as a three-category ranking problem, the objective of which is to rank conversion impressions before click-only ones and non-click impressions after click-only ones. More specifically, the training data D is divided into three sets, N^{++}, N^+, and N^- according to the type of sample labels. Set N^{++} contains the instances with conversion events, N^+ is composed of instances with click-only events that result in clicks and no conversion, and N^- contains ones with non-click. The relationship between the various types of responses is illustrated in Fig. 1. For an arbitrary impression triple (x_i, x_j, x_k), where $x_i \in N^{++}$ is a conversion sample, $x_j \in N^+$ is a click-only sample and $x_k \in N^-$ is a non-click sample, the tripletwise ranking method attempts to

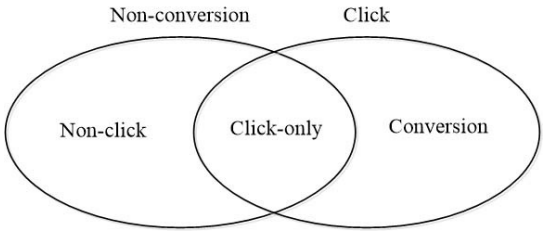

Figure 1: Relationship between response types.
The entire training set can be divided into two parts, conversion and non-conversion, or into a different two parts, click and non-click, or into three parts, conversion, click-only, and non-click. The non-conversion subset can also be divided into two parts, click-only and non-click. The click subset can also be divided into two parts, click-only and conversion.

correctly rank x_i above x_j and x_k below x_j. Alternatively, we denote this condition by $x_i \succ x_j$ and $x_j \succ x_k$. Thus, our method takes into consideration the correct order of random pairs of conversion and click-only, as well as the correct ranking of random pairs of click-only and non-click.

We denote by $T = \{(x_i, x_j, x_k) | x_i \in N^{++}, x_j \in N^{+}, x_k \in N^{-}\}$ the set of all impression triples composed of conversion, click-only, and non-click. We extend BPR based on pairwise learning [16] from one pairwise comparison to one tripletwise comparison. The Bayesian formulation for finding the correct ranking is to maximize the follow posterior probability, where Θ represents the parameter vector of an arbitrary model class (e.g., matrix factorization).

$$(\Theta | \succ) \propto p(\succ | \Theta)\, p(\Theta) \tag{9}$$

Using a similar idea and inference process [16, 21], we can formulate the maximum posterior estimator to derive our generic optimization function for tripletwise ranking, TWR-OPT. More details are given in the literature [21].

$$\begin{aligned} TWR-OPT &:= \ln p(\Theta | \succ) \\ &= \ln p\left(x_i \succ x_j \wedge x_j \succ x_k | \Theta\right) p(\Theta) \\ &= \ln \prod_{(x_i, x_j, x_k) \in T} p(x_i \succ x_j | \Theta) p(x_j \succ x_k | \Theta) p(\Theta) \\ &= \sum_{(x_i, x_j, x_k) \in T} \ln \sigma(\Delta \hat{r}_{ij}) + \ln \sigma(\Delta \hat{r}_{jk}) + \ln p(\Theta) \\ &= \sum_{(x_i, x_j, x_k) \in T} \ln \sigma\left(\Delta \hat{r}_{ij}\right) + \ln \sigma\left(\Delta \hat{r}_{jk}\right) - \lambda_\Theta \|\Theta\|^2 \end{aligned} \tag{10}$$

where $\Delta \hat{r}_{ij} = \hat{r}_i - \hat{r}_j$, σ is the logistic sigmoid $\sigma(x) = \frac{1}{1+\exp(-x)}$, and λ_Θ represents the model specific regularization parameters. Our goal is to maximize the log likelihood of correctly ordering the advertisement impression chances. This is equivalent to minimizing the negative one:

$$\mathrm{L}(\Theta, T) = \sum_{(x_i, x_j, x_k) \in T} -\alpha \ln \sigma(\Delta \hat{r}_{ij}) - (1-\alpha) \ln \sigma(\Delta \hat{r}_{jk}) + \lambda_\Theta \|\Theta\|^2 \tag{11}$$

where the parameter $\alpha \in [0, 1]$ trades off the optimization of conversion and click-only pairwise loss and the optimization of click-only and non-click pairwise loss. If $\alpha = 1$ or $\alpha = 0$ is set, the loss is degraded to pairwise comparison learning for binary classification. Setting α to an intermediate value forces the optimization to consider both pairs' ranking loss, that is, a triple ranking loss.

4.2 Framework of Combined Loss Function

The predicted CVR scores with tripletwise ranking yield a good ranking metric as measured by the AUC, but poor regression values as measured by the mean squared error. In this section, we adopt a combined framework similar to that proposed by Sculley [18] to combine the regression loss functions described in Section 2 with the tripletwise ranking loss function. The combined loss function is

$$J(\Theta) = \beta L_1(\Theta, D) + (1-\beta) L_2(\Theta, T) + \frac{\lambda}{2} \|\Theta\|^2 \tag{12}$$

where $L_1(\Theta, D)$ represents the regression loss and $L_2(\Theta, T)$ the triplet ranking loss. The parameter $\beta \in [0, 1]$ trades off optimization of the regression loss and optimization of the tripletwise ranking loss. If $\beta = 1$ is set, the loss is degraded to a single regression loss, and if $\beta = 1$ is set, it is degraded to a separate tripletwise loss. Setting β to an intermediate value forces the optimization to consider both regression and ranking loss terms [18]. In addition, the parameter λ adjusts the extent of the regularization performed.

In this paper, we focus mainly on two types of models widely used in response prediction, logistic regression and matrix factorization. We chose logistic loss for the logistic regression model as both regression and as pairwise loss. The Euclidean distance loss given in section 3.2.2 was chosen as the regression loss for matrix factorization, and BPR-based pairwise loss described in section 3.3 is used as the ranking loss for matrix factorization.

4.3 Stochastic Gradient Descent-based Learning Algorithm

In this section, we present an SGD algorithm-based learning algorithm for combined regression and triplet ranking loss optimization. Since the combined loss function is differentiable, gradient descent-based algorithms can be used for minimization. The simplicity and easy implementation of SGD-based learning render CRT optimization suitable for large-scale data problems. A naive method for optimizing our CRT loss function is to enumerate the full set T of candidate triples. Because $|T|$ is cubic in $|D|$, this would be impracticable for massive data because of the resulting unsupportable data size. For more efficient computation, we take an approach similar to that in [16, 18] and adopt bootstrap sampling of training triples from T rather than constructing T explicitly. According to existing research studies [16, 18], not only is the convergence rate of this training method effective , but also its performance is more stable than when the full set T of candidate triples is enumerated. Algorithm 1 gives the steps of the SGD-based efficient learning algorithm for solving CRT optimization. The gradient of Θ is calculated as

$$\frac{\partial J}{\partial \Theta} = \beta \frac{\partial L_1}{\partial \Theta} + (1-\beta) \frac{\partial L_2}{\partial \Theta} + \lambda \Theta \tag{13}$$

In more detail, for θ_i associated with conversion sample x_i, the gradient is calculated as

$$\frac{\partial J}{\partial \Theta_i} = \beta \frac{\partial L_1}{\partial \Theta_i} + (1-\beta)\frac{-\alpha \exp(-\Delta \hat{r}_{ij})}{1+\exp(-\Delta \hat{r}_{ij})} \cdot \frac{\partial \hat{r}}{\partial \Theta_i} + \lambda \Theta_i \tag{14}$$

Then, for θ_j associated with click-only sample x_j, the gradient is calculated as

$$\begin{aligned}\frac{\partial J}{\partial \Theta_j} =& \beta \frac{\partial L_1}{\partial \Theta_j} + (1-\beta)\frac{-(1-\alpha) \exp(-\Delta \hat{r}_{jk})}{1+\exp(-\Delta \hat{r}_{jk})} \cdot \frac{\partial \hat{r}}{\partial \Theta_j} \\ &+ (1-\beta)\frac{\alpha \exp(-\Delta \hat{r}_{ij})}{1+\exp(-\Delta \hat{r}_{ij})} \cdot \frac{\partial \hat{r}}{\partial \Theta_j} + \lambda \Theta_j\end{aligned} \tag{15}$$

Finally, for θ_k associated with non-click sample x_k, the gradient is calculated as

$$\frac{\partial J}{\partial \Theta_k} = \beta \frac{\partial L_1}{\partial \Theta_k} + (1-\beta)\frac{\alpha \exp(-\Delta \hat{r}_{jk})}{1+\exp(-\Delta \hat{r}_{jk})} \cdot \frac{\partial \hat{r}}{\partial \Theta_k} + \lambda \Theta_k \tag{16}$$

Algorithm 1 Combined Regression and Tripletwise Ranking

Input: tradeoff parameter α and β, learning rate η, regularization parameter λ, training data D
Output: Θ

initialize Θ;
construct sets N^{++}, N^{+}, and N^{-} separately;
repeat
  pick x_i uniformly at random from set N^{++};
  pick x_j uniformly at random from set N^{+};
  pick x_k uniformly at random from set N^{-};
  for each $\theta_i \in \Theta_i$ **do**
    calculate $\frac{\partial L_1}{\partial \theta_i}$ according to specific regression loss function and x_i.
    calculate $\frac{\partial \hat{r}}{\partial \theta_i}$ according to specific prediction function.
    update θ_i according to the gradient equation(14);
  end for
  for each $\theta_j \in \Theta_j$ **do**
    calculate $\frac{\partial L_1}{\partial \theta_j}$ according to specific regression loss function and x_j.
    calculate $\frac{\partial \hat{r}}{\partial \theta_j}$ according to specific prediction function.
    update θ_j according to the gradient equation(15);
  end for
  for each $\theta_k \in \Theta_k$ **do**
    calculate $\frac{\partial L_1}{\partial \theta_k}$ according to specific regression loss function and x_k.
    calculate $\frac{\partial \hat{r}}{\partial \theta_k}$ according to specific prediction function.
    update θ_k according to the gradient equation (16);
  end for
until converge
return Θ.

5 EXPERIMENTS

5.1 Datasets

Datasets including both click information and conversion information are rare. We used three season datasets of the global bidding algorithm competition released by the advertising demand-side platform (DSP) iPinYou [12] in 2014 to evaluate our proposed methods, as described in this section. Each season dataset contains impression, click, and conversion logs and is divided into two parts: a training dataset and test dataset. The characteristics of the datasets are shown in Table 1 and Table 2. Season 1 contains three advertisers: a consumer packaged goods (CPG), an e-Commerce, and a vertical online media advertiser. Season 2 contains five advertisers, from e-commerce, software, oil, and tires, and season 3 contains four advertisers including those of milk powder, telecom, footwear, and mobile e-commerce applications.

Table 1 presents the statistical CTR and CVR respectively of the training sets and test sets. Historical response rates refer to the ratio of impressions with click or conversion events to all impressions. It can be seen that the historical CTR in the training datasets is less than 0.1 and the historical CVR in the training datasets is less than 0.02, which is considerably less than that in recommender systems such as MovieLens (approximately 4.5%) or Netflix (approximately 1.2%) [13]. Higher data sparsity makes the CVR estimation task more difficult than the recommendation problem.

Each record contains four types of information: user features (iPinYou user ID, user-agent (UA), region and city, etc.), publisher features (advertisement slot ID, slot width, slot height, and Web page domain, etc.), advertisement features (creative ID, advertiser ID, landing page URL etc.) and other features related to auction (advertisement exchange, bidding price, paying price, etc.). The dimensionality (denoted as "No." in the tables) of some important features are shown in Table 2. In this table, "UA" refers to user-agent, which contains the type of Web browser a user used, and "domain" means the domain of the Web page a user is browsing. Features related to auctions are usually utilized for real-time bidding strategies or bid optimization research. We discarded these features when estimating the response scores.

5.2 Performance Metrics

We used regression- and rank-based performance metrics to evaluate all the models. In addition, we also evaluated their multi-classification ability in terms of multi-class AUC and a gain-weighted ranking in terms of normalized discounted cumulative gain.

5.2.1 Root Mean Squared Error (RMSE). We used the classical root mean squared error (RMSE) measure as our primary regression-based performance metric. Values closer to zero indicate better performances. The RMSE was calculated as

$$RMSE = \sqrt{\frac{1}{|D|}\sum_{(u,p,a)\in D}\left(r_{u,p,a} - \hat{r}_{u,p,a}\right)^2} \tag{17}$$

5.2.2 Area under the Receiver Operating Characteristics Curve (AUC). Because ROC curves are insensitive to changes in class distribution, the AUC has become a commonly used metric for testing the quality of advertisements' CTR prediction. Therefore, we employed AUC as our ranking-based performance metric for comparing the prediction quality of models for binary classification. We used Algorithm 3 in [5] to calculate the AUC. The AUC value shows the ability of a model to rank a randomly selected positive

Table 1: Characteristics of datasets

Season	First		Second		Third	
Dataset	Training set	Test set	Training set	Test set	Training set	Test set
Date	Mar. 11–Mar. 17	Mar. 18–Mar. 20	Jun. 6–Jun. 12	Jun. 3–Jun. 15	Oct. 19–Oct. 27	Oct. 21–Oct. 28
No. of impressions	9262861	2594386	12237229	2524630	3158171	1579086
No. of clicks	7002	1932	8961	1873	2709	1120
No. of conversions	59	12	413	69	537	241
CTR	0.0756%	0.0745%	0.073%	0.074%	0.0860%	0.0710%
CVR	0.0006%	0.0005%	0.0034%	0.0027%	0.0170%	0.0153%

Table 2: Dimensionality of major attributes

Season	First		Second		Third	
Attributes	Training set	Test set	Training set	Test set	Training set	Test set
No. of users	6799908	2164525	10146491	2310303	2818424	1490321
No. of tags	0	0	45	45	69	69
No. of UA	77	72	83	64	69	58
No. of slots	124684	58945	141515	48458	53518	43603
No. of domains	30434	18504	28505	14695	22803	19163
No. of URL	2082249	811585	2362123	663218	963576	552694
No. of creatives	32	33	74	74	57	54
No. of advertisers	1	1	5	5	4	4

example above a randomly picked negative example. Values closer to 1 indicate better performances.

5.2.3 Multi-class Area under the Receiver Operating Characteristics Curve. The AUC measure can be used to evaluate ranking performance only for binary classification. A three-class problem introduces the issue of combining three pairwise discriminability values. Hanley and McNeal presented an excellent discussion of these issues and derived a formulation that measures the unweighted pairwise discriminability of classes [7]. Their measure is equivalent to

$$Multi - AUC = \frac{2}{|C|\,(|C|-1)} \sum_{\{c_i, c_j\} \in C} AUC\,(c_i,\ c_j) \tag{18}$$

where C is the number of classes and $AUC(c_i, c_j)$ is the AUC value associated with classes c_i and c_j. Values closer to 1 show better performances.

5.2.4 Normalized Discounted Cumulative Gain (NDCG). In our experiments, we also adopted normalized discounted cumulative gain (NDCG), which is a gain-weighted ranking measure. NDCG measures the usefulness, or gain, of the target entity based on its position in the result list. NDCG is calculated as

$$NDCG = \frac{1}{G_n} \cdot \sum_{j=1}^{n} \frac{2^{g_j} - 1}{\log\,(1+j)} \tag{19}$$

where j is the position of a test sample in the final ranking list, g_j is the gain of the instance in position j, and $1/G_n$ is a normalization constant ensuring that the perfect NDCG score for the set of test instances is 1.0. Thus, G_n is commonly defined as the best total gain of optimal ranking. Values closer to 1 show better performances.

5.3 Experimental Setup

For each season dataset, we split the training dataset into two parts according to the impression date and used the last three days' data as the validation dataset to train model hyperparameters, such as learning rate and regularization coefficient, for each method. The tripletwise ranking parameter α was set to 0.3 for season 1 and season 2, and 0.7 for season 3. The CRT parameter β was set to 0.2. Four was chosen as the number of latent factors for each season. The learning rate was set as 0.00001 and the regularization coefficient λ was equal to 0.001. For the logistic regression models, the learning rate and the regularization parameter were both set as 0.001 for each season. Note that, although different approaches maybe have an identical learning rate, the numbers of iterations they need for convergence are commonly unequal. Overall, the convergence speed in order from high to low is CRT, tripletwise ranking methods, pairwise ranking methods, and regression-based methods. In order to examine the NDCG values, we assumed the click gain is always 1 and set the conversion gains respectively as 10 and 20 as required by iPinYou content.

5.4 Experimental Results

We used regression-only, pairwise ranking-only, and the CRR method (logistic regression prediction with logistic loss) [18] as baselines in the evaluation of our CRT method. The experimental results are shown in Table 3 for the first season, in Table 4 for the second season, and in Table 5 for the third season. The notation "NDCG@X" means the NDCG value when the conversion gain is set as "X" times the gain of a click. Overall, the prediction performance for the second season dataset is obviously better than that for the other two season datasets. This is because, as compared with the second

season dataset, the first season dataset not only has no useful type of attribute "user tag," but also lacks sufficient hierarchical information that can efficiently help alleviate data sparsity. As compared with the second season dataset, there are no conversion data for the last days in the third season training dataset, and the size of the third season dataset is only almost a quarter of that of the second season.

First, we compare the performance of the regression-only method with that of the ranking-only method. Unsurprisingly, the regression-based method is clearly better in terms of the regression-based metric RMSE and easily exceeded by ranking-based methods in terms of AUC and multi-AUC metrics. Meanwhile, ranking-based methods also have disappointing values in terms of the regression-based metric.

Second, we compare the performance of the tripletwise ranking method with that of the pairwise ranking method. As expected, the tripletwise ranking method slightly outperforms or performs very similarly to the pairwise ranking method on the first and second season datasets in terms of AUC. Meanwhile, the tripletwise ranking achieves almost double the regression performance of pairwise ranking in terms of RMSE values on all datasets. However, tripletwise ranking is slightly outperformed by pairwise ranking on the third season dataset in terms of AUC. This is because no conversion data are available for last four days in the training dataset of season 3. Because of the lack of conversion information, tripletwise ranking has to borrow too much click information from the historical data for CVR prediction. Furthermore, the historical data in season 3 contain more click noise caused mainly by mobile advertising. However, it is noteworthy that tripletwise ranking achieves a stronger performance according to multi-AUC and NDCG than pairwise ranking for all datasets. According to the multi-AUC equation 18, the final multi-class AUC is composed of three AUC values associated with three pairs of individual binary classification: conversion vs. click-only, click-only vs. non-click, and conversion vs. non-click. Because, for CVR prediction, the pairwise ranking method optimizes only the correct order of conversion and non-conversion (conversion vs. click-only and conversion vs. non-conversion) and since both click-only and non-click impressions belong to the same non-conversion category, the pairwise ranking method fails to correctly sort click-only and non-click examples. However, the tripletwise ranking method takes the correct orders of two pairs among three classes into consideration, and the tripletwise learning algorithm shows promise of being capable of achieving better ranking for three classes.

Third, we evaluated the performance of our combined CRT method as compared with that of the tripletwise ranking-only method. As shown in Tables 3–5, the CRT method not only shows a performance similar to that of the tripletwise ranking-only method in terms of AUC and multi-AUC, but also greatly improves the regression-based metric in terms of RMSE.

Finally, we evaluated the performance of the proposed CRT method as compared with existing combined CRR methods [18]. Both methods yield a better AUC, as well as a better RMSE, and show a more promising performance than regression-only and ranking-only methods. Although the CRR method slightly outperforms our method in terms of RMSE values, the CRT method yields the best multi-AUC values and NDCG values for all datasets. This means that our method is more suitable for three-class ranking problems, such as response prediction for buyers in RTB.

6 CONCLUSIONS

We proposed a tripletwise learning optimization for three-category ranking and used it to estimate the CVR for buy-sides in RTB. Through borrowing click information from the historical impression data, our approach effectively alleviates the high sparsity of the conversion data and achieves a better prediction performance than pairwise ranking. Furthermore, we presented a combined regression and tripletwise ranking method, CRT, which yields a promising performance in terms of both regression and ranking metrics, especially three category-ranking metrics. An SGD-based learning algorithm was presented, which facilitates the implementation of the CRT algorithm and is efficient when applied to large-scale datasets.

ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China under Grant Nos. 61602131, 61572151, 61672192 and High-tech R&D Program of China (863 Program) 2015AA015405

REFERENCES

[1] Deepak Agarwal, Bee-Chung Chen, and Pradheep Elango. 2009. Spatio-temporal Models for Estimating Click-through Rate. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*. ACM, New York, NY, USA, 21–30. https://doi.org/10.1145/1526709.1526713

[2] Olivier Chapelle, Eren Manavoglu, and Romer Rosales. 2014. Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 4 (2014), 61.

[3] Lin Chen, Peng Zhang, and Baoxin Li. 2015. Fusing pointwise and pairwise labels for supporting user-adaptive image retrieval. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 67–74.

[4] Tianqi Chen, Zhao Zheng, Qiuxia Lu, Weinan Zhang, and Yong Yu. 2011. Feature-Based Matrix Factorization. *arXiv preprint arXiv:1109.2271* (2011). http://arxiv.org/abs/1109.2271

[5] Tom Fawcett. 2004. ROC graphs: Notes and practical considerations for researchers. *Machine learning* 31 (2004), 1–38.

[6] Thore Graepel, Joaquin Qui nonero Candela, Thomas Borchert, and Ralf Herbrich. 2010. 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)*. ACM, New York, NY, USA, 13–20.

[7] JA Hanley and BJ McNeal. 1982. A simple generalization of the area under the ROC curve to multiple class classification problems. *Radiology* 143 (1982), 29–36.

[8] Maryam Karimzadehgan, Wei Li, Ruofei Zhang, and Jianchang Mao. 2011. A Stochastic Learning-to-rank Algorithm and Its Application to Contextual Advertising. In *Proceedings of the 20th International Conference on World Wide Web (WWW '11)*. ACM, New York, NY, USA, 377–386. https://doi.org/10.1145/1963405.1963460

[9] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. 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 (KDD '12)*. ACM, New York, NY, USA, 768–776. https://doi.org/10.1145/2339530.2339651

[10] Cheng Li, Yue Lu, Qiaozhu Mei, Dong Wang, and Sandeep Pandey. 2015. Click-through Prediction for Advertising in Twitter Timeline. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. ACM, New York, NY, USA, 1959–1968. https://doi.org/10.1145/2783258.2788582

[11] Sheng Li, Jaya Kawale, and Yun Fu. 2015. Predicting User Behavior in Display Advertising via Dynamic Collective Matrix Factorization. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. ACM, New York, NY, USA, 875–878. https://doi.org/10.1145/2766462.2767781

[12] Hairen Liao, Lingxiao Peng, Zhenchuan Liu, and Xuehua Shen. 2014. iPinYou Global RTB Bidding Algorithm Competition Dataset. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising (ADKDD'14)*. ACM, New York, NY, USA, Article 6, 6 pages. https://doi.org/10.1145/2648584.2648590

Table 3: Experimental results for the first season

Method	LR					MF				
	RMSE	AUC	Multi_AUC	NDCG@10	NDCG@20	RMSE	AUC	Multi_AUC	NDCG@10	NDCG@20
Regression-only	**0.0112**	0.7541	0.6436	0.1398	0.1248	**0.0117**	0.7095	0.6349	0.1420	0.1287
Pairwise Ranking-only	0.8965	**0.7625**	0.6349	0.1372	0.1270	0.8987	0.7424	0.6581	0.1434	0.1303
Tripletwise Ranking-only	0.4522	0.7612	**0.6653**	*0.1511*	0.1352	0.4565	**0.7500**	*0.6621*	**0.1503**	*0.1361*
CRR	*0.012*	0.7453	0.6552	0.1457	*0.1356*	*0.0123*	0.7453	0.6552	0.1457	0.1356
CRT	0.0286	*0.7622*	*0.6652*	**0.1512**	**0.1358**	0.0253	*0.7489*	**0.6635**	*0.1502*	**0.1365**

Table 4: Experimental results for the second season

Method	LR					MF				
	RMSE	AUC	Multi_AUC	NDCG@10	NDCG@20	RMSE	AUC	Multi_AUC	NDCG@10	NDCG@20
Regression-only	*0.0175*	0.9104	0.7493	0.3217	0.3169	**0.0193**	0.9125	0.7795	0.5371	0.5276
Pairwise Ranking-only	1.0001	0.8722	0.7165	0.3026	0.3124	0.9709	*0.9601*	0.8045	0.5467	0.5467
Tripletwise Ranking-only	0.5619	*0.9516*	**0.8364**	*0.4361*	*0.3684*	0.4591	0.9600	*0.8735*	*0.5749*	*0.5750*
CRR	**0.0168**	**0.9598**	0.8156	0.3587	0.3597	*0.0198*	0.9598	0.8156	0.5587	0.5597
CRT	0.0460	*0.9516*	*0.8262*	**0.4512**	**0.3756**	0.0256	**0.9616**	**0.8789**	**0.5789**	**0.5759**

Table 5: Experimental results for the third season

Method	LR					MF				
	RMSE	AUC	Multi_AUC	NDCG@10	NDCG@20	RMSE	AUC	Multi_AUC	NDCG@10	NDCG@20
Regression-only	*0.025*	0.7653	0.6479	0.3500	0.3497	*0.0281*	0.6597	0.5951	0.3373	0.337
Pairwise Ranking-only	0.7361	*0.7694*	0.6505	0.3411	0.3356	0.9891	0.7673	0.6592	0.3498	0.3495
Tripletwise Ranking-only	0.4306	0.7688	**0.6956**	**0.3532**	0.3501	0.4934	0.7671	**0.6961**	*0.3515*	*0.3512*
CRR	**0.0232**	**0.7696**	0.6622	0.3508	*0.3505*	**0.0232**	**0.7696**	0.6622	0.3508	0.3505
CRT	0.0503	0.7687	*0.6823*	*0.3524*	**0.3521**	0.03563	*0.7683*	*0.6959*	**0.3528**	**0.3526**

[13] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. 2012. Recommender systems. *Physics Reports* 519, 1 (2012), 1–49.

[14] Aditya Krishna Menon, Krishna-Prasad Chitrapura, Sachin Garg, Deepak Agarwal, and Nagaraj Kota. 2011. Response Prediction Using Collaborative Filtering with Hierarchies and Side-information. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. ACM, New York, NY, USA, 141–149. https://doi.org/10.1145/2020408.2020436

[15] Aditya Krishna Menon, Xiaoqian J Jiang, Shankar Vembu, Charles Elkan, and Lucila Ohno-Machado. 2012. Predicting accurate probabilities with a ranking loss. In *Proceedings of the 29th International Conference on Machine Learning*, Vol. 2012. NIH Public Access, ACM, Edinburgh, Scotland,UK, 703.

[16] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, Arlington, Virginia, United States, 452–461. http://dl.acm.org/citation.cfm?id=1795114.1795167

[17] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*. ACM, ACM, New York, NY, USA, 521–530.

[18] David Sculley. 2010. Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, Washington,USA, 979–988.

[19] Lili Shan, Lei Lin, Di Shao, and Xiaolong Wang. 2014. CTR Prediction for DSP with Improved Cube Factorization Model from Historical Bidding Log. In *Neural Information Processing (Lecture Notes in Computer Science)*, Vol. 8836. Springer International Publishing, New York, NY, USA, 17–24. https://doi.org/10.1007/978-3-319-12643-2_3

[20] Lili Shan, Lei Lin, Chengjie Sun, and Xiaolong Wang. 2016. Predicting ad click-through rates via feature-based fully coupled interaction tensor factorization. *Electronic Commerce Research & Applications* 16, C (2016), 30–42.

[21] Lili Shan, Lei Lin, Chengjie Sun, Xiaolong Wang, and Bingquan Liu. 2017. Optimizing ranking for response prediction via triplet-wise learning from historical feedback. *International Journal of Machine Learning & Cybernetics* 8, 6 (2017), 1777–1793.

[22] Yukihiro Tagami, Shingo Ono, Koji Yamamoto, Koji Tsukamoto, and Akira Tajima. 2013. CTR Prediction for Contextual Advertising: Learning-to-rank Approach. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising (ADKDD '13)*. ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/2501040.2501978

[23] Yilin Wang, Suhang Wang, Jiliang Tang, Huan Liu, and Baoxin Li. 2016. PPP: Joint pointwise and pairwise image label prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, 6005–6013.

[24] Kuan-Wei Wu, Chun-Sung Ferng, Chia-Hua Ho, An-Chun Liang, Chun-Heng Huang, Wei-Yuan Shen, Jyun-Yu Jiang, Ming-Hao Yang, Ting-Wei Lin, Ching-Pei Lee, et al. 2012. A two-stage ensemble of diverse models for advertisement ranking in KDD Cup 2012. *KDDCup* (2012).

[25] Ling Yan, Wu-Jun Li, Gui-Rong Xue, and Dingyi Han. 2014. Coupled group lasso for web-scale CTR prediction in display advertising. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. ACM, New York, NY, USA, 802–810.