Debiasing Learning to Rank Models with Generative Adversarial Networks

Hui Cai[1], Chengyu Wang[1], Xiaofeng He[2⋆]

[1] School of Software Engineering, East China Normal University
[2] School of Computer Science and Technology, East China Normal University
huicai.me@gmail.com, chywang2013@gmail.com
hexf@cs.ecnu.edu.cn

Abstract. Unbiased learning to rank aims to generate optimal orders for candidates utilizing noisy click-through data. To deal with such problem, most models treat the biased click labels as combined supervision of relevance and propensity, which pay little attention to the uncertainty of implicit user feedback. We propose a semi-supervised framework to address this issue, namely ULTRGAN (Unbiased Learning To Rank with Generative Adversarial Networks). The unified framework regards the task as semi-supervised learning with missing labels, and employs adversarial training to debias click-through datasets. In ULTRGAN, the generator samples potential negative examples combined with true positive examples for the discriminator. Meanwhile, the discriminator challenges the generator for better performances. We further incorporate pairwise debiasing to generate unbiased labels diffusing from the discriminator to the generator. Experimental results over both synthetic and real-world datasets show the effectiveness and robustness of ULTRGAN.

Keywords: unbiased learning to rank · inverse propensity weighting · generative adversarial networks · semi-supervised learning

1 Introduction

Learning To Rank (LTR) [19] is a family of machine learning models, used in a wide range of applications in Information Retrieval (IR), such as Web search, recommender systems and question answering. Given a query and the potential candidates, LTR maps query-document feature vectors to relevance scores for the generation of optimal orders. Existing LTR models optimize scoring functions over individual documents [11], document pairs [5,6] or the whole ranked list [7] in the setting of supervised learning.

Human-labeled relevance scores are necessary for the training of supervised LTR models, which requires the time-intensive manual work to curate. In some special scenarios such as personalized search, manual annotations are even inaccessible due to privacy restrictions [29,30]. More severely, real user preferences

[⋆] Corresponding author.

can not be precisely annotated, which are dynamic and context-aware. In modern IR systems, click-through data can be collected in massive amount as the substitute for relevance scores [16]. However, such data is heavily biased. For example, position bias is a typical noise that people tend to click on the results presented in higher positions [17]. If LTR models directly consider the click and non-click signals as positive and negative, they actually learn the user bias instead of the inherent relevance between queries and candidate documents.

Unbiased Learning To Rank (ULTR) [2, 21] tries to solve the problem with the biased click data. Counterfactual LTR is a popular solution, which mostly consists of two types of methods, i.e., click models [10] and randomization experiments [18, 29]. Click models make assumptions about user behaviors and maximize the received click likelihood. Randomization experiments extract propensities for each position by presenting documents in random orders. These models split ULTR into two separate stages: i) label debiasing and ii) relevance learning. Hence, the prevalent techniques could introduce uncertainty to the follow-up work when the bias is not completely rectified. With the rapid advancement of counterfactual LTR, end-to-end algorithms are proposed [1, 15, 30], in order to improve ranking performances and to make inferences about selection bias.

Despite the success made in recent years, we observe that existing approaches utilize the inverse propensity weighting technique to discriminate against all the candidates [1, 15, 18, 29, 30]. It should be noted that in the task of relevance prediction, only part of labels generated by such approaches (especially head exposures) are valid, non-clicks (mostly presented in tail candidates) do not necessarily reflect irrelevance [17]. Therefore, the selection bias of these supervised ULTR models is still avoidable, resulting from the neglect of sampling competitive document pairs. This problem naturally motivates us to treat ULTR as a semi-supervised learning problem, with a large number of missing labels. It is also similar to a causal inference problem of selection bias [25].

In this paper, we propose a new framework named ULTRGAN (Unbiased Learning To Rank with Generative Adversarial Networks) to further improve the performance of ULTR. It is built upon the minimax game from Generative Adversarial Network (GAN) [14], and optimizes rankings with limited labels [28]. Specifically, in ULTRGAN, a generator plays as a sampler to generate hard negative results (i.e., less irrelevant candidates) for the discriminator, while the discriminator challenges the generator for better performances. Meanwhile, we incorporate the label debiasing technique [15] during the training of the discriminator, which enables true relevance to propagate from the discriminator to the generator. Experimental results demonstrate the advantages of ULTRGAN. In summary, we make the following major contributions:

- We formulate the ULTR problem in a semi-supervised setting, and propose the ULTRGAN framework to improve ULTR based adversarial learning.
- We design the minimax game between the two components in ULTRGAN, and incorporate the pairwise debiasing technique to the discriminator.
- We experimentally show the effectiveness and robustness of ULTRGAN over both synthetic and real-world datasets.

The rest of this paper is organized as follows. In Section 2, we review the prior related literature. Section 3 and Section 4 give the theoretical analysis on ULTR and describe the proposed model ULTRGAN, respectively. Experimental setups and result analysis are described in Section 5. Finally, we conclude the paper and discuss the future work in Section 6.

2 Related Work

In this section, we give a brief overview on the related work of LTR, ULTR and adversarial learning techniques for IR.

2.1 LTR and ULTR

In classical IR research, LTR [19] is mostly considered as a supervised learning problem which optimizes the ranking function, mapping from feature vectors to relevance scores. Typically, human annotations in TREC style [8] are used as supervision, which are expensive and unpractical under certain circumstances [29, 30]. Click data is a resource that implies real user preferences without privacy restrictions. However, the heavy inherent bias in the click data is a critical concern for designing IR models, such as position bias [17], presentation bias [33] and trust bias [23]. To infer true preferences, early attempts apply result interleaving and heuristic rules. For example, Joachims [16] proposes the "skip-above" strategy to filter pairs with high confidence. However, these methods either bring in instability nor are limited to identified counterfactual samples.

ULTR [2, 21] optimizes relevance prediction functions with noisy click data. As summarized in [21], two types of techniques have been proposed for the problem. The first one is online LTR, which directly interacts with users and adjusts to immediate feedbacks [22, 32]. Another is called counterfactual LTR, performing offline training with historical data, which is the focus of this work.

Click models [10] are a collection of counterfactual LTR methods, which employ Bayesian graph models to simulate user behaviors. The Position-Based Model (PBM) [24] assumes that the click probability only relates to that of relevance and observation. The Cascade Model (CM) [12] believes that users examine results from head to tail and click only once. The User Browsing Model (UBM) [13] allows for multiple clicks and considers former-click effects. Recently, the neural click model is proposed in [4]. These models rely on various assumptions to justify user behaviors. Another type of solutions is called randomization experiments [18, 29]. According to observational studies on causal inference [25], we consider whether a user examines the result as the treatment, and the user's action (click or non-click) as the outcome. However, user behaviors are influenced by surfing habits and presentation orders (i.e., the selection bias). By presenting results in random orders at the cost of user experiences, the bias can be eliminated in a theoretically principal way [18, 29].

Above methods share a common thinking of estimating the selection bias in advance, which has negative effects on the final ranking if propensities are not accurately estimated. Recent studies [1, 15, 30] differ from the above methods in an end-to-end way. Wang et al. [30] propose a novel regression-based EM algorithm for propensity and relevance estimation simultaneously. Ai et al. [1] propose a dual learning algorithm for deep ranking models. Hu et al. [15] extend pointwise learning to pairwise and apply to LambdaMart [6]. Our proposed approach is different from above algorithms for treating click labels as semi-supervised signals and sampling pairs for discriminative learning.

2.2 Adversarial Learning in IR

Adversarial learning has been leveraged for designing various IR systems. For example, GAN-related models have been utilized to deal with semi-supervised learning [20,28] and Positive-Unlabeled (PU) learning [3] problems in IR. Unlike the original GAN [14] which generates continuous data such as images, these models select discrete documents or words from candidates. The discriminator tries to distinguish positive instances from selected negative instances by the generator, while the generator aims to estimate real data distribution. The usage of adversarial learning in ULTR differs from existing models in the need to alleviate the propensity of examination from click labels.

3 Theoretical Analysis

In this section, we present the definition of ULTR, and give theoretical analysis on debiasing pairwise LTR. Table 1 summaries the important annotations.

Table 1: A summary of notations.

$\mathbf{Q}$, Q, q	The universal set of queries $\mathbf{Q}$, a sample set Q, a query instance q.
π_q, x, i, y	A ranked list π_q of query q produced by ranking system, a document x in the ith position and its relevance y.
$o_q^{x,i}$ (o_i^+,o_i^-), $c_q^{x,i}$ (c_i^+, c_i^-), $r_q^{x,i}$ (r_i^+, r_i^-)	Bernoulli variables that represent whether a document x in the ith position of the ranked list π_q is observed ($o_q^{x,i}$), clicked ($c_q^{x,i}$), or perceived as relevant ($r_q^{x,i}$).
G, θ, D, ϕ	A generator G with parameters θ, a discriminator D with parameters ϕ.
$g_\theta(x, q)$, $f_\phi(x, q)$	The generative and discriminative retrieval functions of G and D for document x given query q.
t_i^+, t_j^-	The positive position ratio for a clicked item in the ith position and the negative position ratio for an unclicked item in the jth position.

3.1 Preliminaries of ULTR

The goal of LTR is to learn the ranking function f that minimizes the global loss. In reality, it is impractical to obtain the universal set of queries $\mathbf{Q}$. Given a subset of queries Q, the normalized loss is defined as: $\hat{L}(f) = \frac{1}{|Q|} \sum_{q \in Q} l(f, q)$, where $l(f, q)$ is the individual ranking loss. The empirical loss function measures the distance between the relevance score y and the predicted score $f(x, q)$ for document x given query q. For IR, the ranking matrices (Mean Average Precision (MAP), normalized Discounted Cumulative Gain (nDCG), etc.) pay the most attention to relevant documents. Hence, the individual loss is defined to approximate the evaluation matrices:

$$l_{rel}(f, q) = \sum_{x \in \pi_q, y=1} l(f(x, q), y)$$

Under the setting of ULTR, the relevance score y is not available. For document x related to query q presented in the position i, let $c_q^{x,i}$, $o_q^{x,i}$ and $r_q^{x,i}$ denote the click, observation and intrinsic relevance respectively. The basic assumption of ULTR is that the probability of being clicked is related to that of being examinated and perceived relevance. Therefore, we need to alleviate selection bias $P(o_q^{x,i})$, with the click-based unbiased loss function as:

$$l_{IPW}(f, q) = \sum_{x \in \pi_q, y=1 \wedge o_q^{x,i}=1} \frac{l(f(x, q), y)}{P(o_q^{x,i} = 1)} = \sum_{x \in \pi_q, c_q^{x,i}=1} \frac{l(f(x, q), y)}{P(o_q^{x,i} = 1)}$$

As proved in [18], the expectation of $l_{IPW}(f, q)$ is equal to the initial LTR loss, i.e., $\mathbb{E}_{o_q}[l_{IPW}(f, q)] = l_{rel}(f, q)$.

3.2 Pairwise ULTR

Hu et al. [15] extend the previous function to the pairwise setting. Assume that:

$$P(c_i^+ \mid x_i) = t_i^+ P(r_i^+ \mid x_i) \qquad P(c_j^- \mid x_j) = t_j^- P(r_j^- \mid x_j)$$

Given the basic assumption $P(c_i^+ \mid x_i) = P(r_i^+ \mid x_i) \cdot P(o_i^+ \mid x_i)$, we directly transform the position ratios as follows:

$$t_i^+ = P(o_i^+ \mid x_i)$$

$$t_j^- = \frac{1 - P(c_j^+ \mid x_j)}{1 - P(r_j^+ \mid x_j)} = \frac{1 - P(r_j^+ \mid x_j) \cdot P(o_j^+ \mid x_j)}{1 - P(r_j^+ \mid x_j)}$$

Therefore, the positive position ratio implies the probability of observation, which is supposed to decrease with position increasing. The negative position ratio is the combination of average relevance probability and observation probability, which depends on the initial ranker.

In pairwise LTR, the empirical ranking loss is defined over the set of document pairs (x_i, x_j) where x_i is relevant and x_j is irrelevant [5, 6]. The pairwise loss concentrates on the relative order between two documents, shown as follows:

$$l_{rel}(f, q)^{pair} = \sum_{x_i, x_j \in \pi_q, r_i^+ \wedge r_j^-} l(f(x_i, q), r_i^+, f(x_j, q), r_j^-)$$

We prove that the ranking model based on our assumption produces unbiased ranking. In ULTR, there exists a set of document pairs (x_i, x_j) in the ith position and jth position where x_i is clicked and x_j is unclicked. The pairwise loss function can be derived as follows:

$$\begin{aligned} l_{unbiased}(f,q)^{pair} &= \mathbb{E}_{c_i^+,c_j^-}\Big[\sum_{x_i,x_j\in\pi_q,c_i^+\wedge c_j^-} \frac{l(f(x_i,q),r_i^+,f(x_j,q),r_j^-)}{t_i^+\cdot t_j^-}\Big] \\ &= \sum_{x_i,x_j\in\pi_q} \frac{\mathbb{E}_{c_i^+,c_j^-}[c_i^+\cdot c_j^-]\cdot l(f(x_i,q),r_i^+,f(x_j,q),r_j^-)}{\frac{P(c_i^+|x_i)P(c_j^-|x_j)}{P(r_i^+|x_i)P(r_j^-|x_j)}} \\ &= \sum_{x_i,x_j\in\pi_q} \frac{P(c_i^+\mid x_i)P(c_j^-\mid x_j)l(f(x_i,q),r_i^+,f(x_j,q),r_j^-)}{\frac{P(c_i^+|x_i)P(c_j^-|x_j)}{P(r_i^+|x_i)P(r_j^-|x_j)}} \\ &= \sum_{x_i,x_j\in\pi_q} P(r_i^+\mid x_i)P(r_j^-\mid x_j)l(f(x_i,q),r_i^+,f(x_j,q),r_j^-) \\ &= \sum_{x_i,x_j\in\pi_q,r_i^+\wedge r_j^-} l(f(x_i,q),r_i^+,f(x_j,q),r_j^-) \end{aligned} \tag{1}$$

Therefore, it is easy to see that $l_{unbiased}(f,q)^{pair} = l_{rel}(f,q)^{pair}$. Based on this conclusion, in the next part, we introduce the model ULTRGAN in detail.

4 The Proposed Approach

In this section, we formally present the ULTRGAN framework, followed by the model details and optimization methods.

4.1 The ULTRGAN Framework

Before deriving our approach for ULTR, we firstly review current problems. Existing ULTR models [1,15,18,29,30] focus on label debiasing to better conduct supervised learning, making all the unclicked documents contribute to discriminative function. However, unclicked samples are composed of true negative (irrelevant) and skipped positive (relevant) results [16,24]. Even with propensity weighting, relevance is still hard to discriminate especially for tail exposures. Therefore, we regard ULTR as a task of semi-supervised, with a small amount of labeled data and a large amount of unlabeled data. Additionally, current adversarial learning models for search problems [20,28] have not employed propensity weighting to deal with labels that are missing not at random (MNAR) [26,31], possibly due to the ignorance of making good use of the side information (e.g. initial presentation orders).

Based on the above considerations, we design a general framework for adversarial ULTR. It is composed of a discriminator, a generator and a bias estimator as shown in Fig.1: i) The minimax game between the two players naturally provides the most difficult cases for each other; ii) The bias estimator fully utilizes propensity-related information for pairwise debiasing. The three elements are introduced as follows:

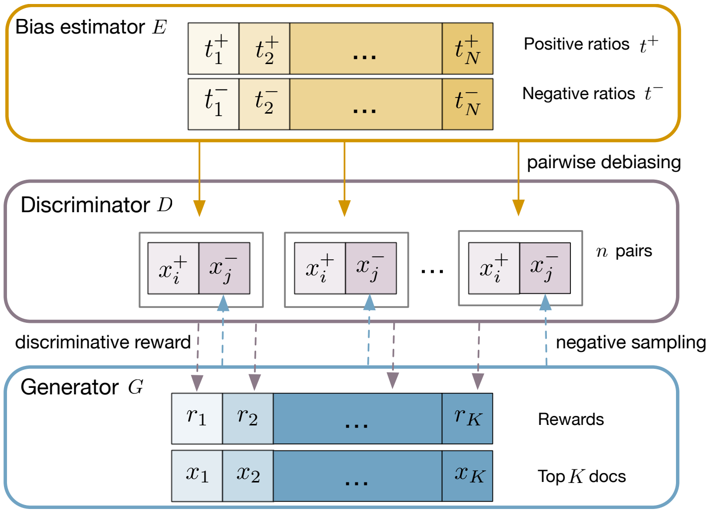

Fig. 1: The general framework of ULTRGAN, which contains a bias estimator E, a discriminator D and a generator G.

- **Bias estimator** E**:** It learns the top N position ratios of t^+, t^- to eliminate selection bias induced by presentation orders for the discriminator.
- **Discriminator** D**:** It learns the classifier $f_\phi(x, q)$, which tries to discriminate between relevant and irrelevant documents. By providing discriminative reward for generated (or selected) documents, it also retrains the generator.
- **Generator** G**:** It learns the distribution $p_\theta(x, q)$, which tries to generate (or select) the K-most relevant documents from candidate pool. It also plays as a dynamic sampler, selecting a less irrelevant document x_j^- for each relevant instance x_i^+ to push the discriminator to its limit.

4.2 Model Details

ULTRGAN is a minimax game played by the generator and discriminator, together with the estimation of position ratios. The discriminator tries to maximize the expectation of data distributions of relevant and irrelevant documents. The generator tries to fit the distribution of true relevant documents by minimizing the refined objective function. Our model considers click propensity for instances respectively in positive and negative groups. It can also be explained as a reweighting method of inverse propensity.

In Web search, for query q, the true data distribution can be described as $p_{true}(x, c_q^{x,i})$, which consists of query-document feature vectors and user clicks.

The overall loss function is:

$$\begin{aligned} J^{G^*,D^*} &= \underset{\theta}{min}\,\underset{\phi}{max} \sum_{q\in Q} \mathbb{E}_{x_i \sim p_{true}(x, c_q^{x,i}=1)} \left[\frac{log(f_\phi(x_i, q))}{t_i^+} \right] \\ &+ \mathbb{E}_{x_j \sim p_\theta(x, c_q^{x,j}=0)} \left[\frac{log(1 - f_\phi(x_j, q))}{t_j^-} \right] \\ &= \underset{\theta}{min}\,\underset{\phi}{max} \sum_{q\in Q} \mathbb{E}_{x_i^+} \left[\frac{log(f_\phi(x_i^+, q))}{t_i^+} \right] + \mathbb{E}_{x_{j,\theta}^-} \left[\frac{log(1 - f_\phi(x_{j,\theta}^-, q))}{t_j^-} \right] \end{aligned}$$

where x_i^+ represents a clicked document for query q in the position i, and $x_{j,\theta}^-$ represents an unclicked document for query q in the position j, sampled by the generator with parameters θ. Our model selectively learns debiased preferences over competitive document pairs, which is different from previous work.

Optimizing Discriminator: In the overall loss, the objective for the discriminator is to find optimal parameters ϕ that maximize the log-likelihood of correctly distinguishing the true and selected relevant documents as follows:

$$\phi^* = \underset{\phi}{argmax} \sum_{q\in Q} \mathbb{E}_{x_i^+} \left[\frac{log(f_\phi(x_i^+, q))}{t_i^+} \right] + \mathbb{E}_{x_{j,\theta}^-} \left[\frac{log(1 - f_\phi(x_{j,\theta}^-, q))}{t_j^-} \right] \tag{2}$$

Pairwise risk function has been proved to be unbiased in Eq.(1), which pays more attention to the relative order between document pairs. Therefore, we can easily extend Eq.(2) with pairwise loss function L:

$$\begin{aligned} \phi^* &= \underset{\phi}{argmax} \sum_{q\in Q} \mathbb{E}_{x_i^+, x_{j,\theta}^-} \left[\frac{L(f_\phi(x_i^+, q) - f_\phi(x_{j,\theta}^-, q))}{t_i^+ \cdot t_j^-} \right] \\ &= \underset{\phi}{argmin} \sum_{q\in Q} \mathbb{E}_{x_i^+, x_{j,\theta}^-} \left[\frac{L(-(f_\phi(x_i^+, q) - f_\phi(x_{j,\theta}^-, q)))}{t_i^+ \cdot t_j^-} \right] \end{aligned} \tag{3}$$

Note that in Eq.(3), the discriminator parameters ϕ, positive ratio t_i^+ and negative ratio t_j^- are unknown. We follow the work [15] to estimate t_i^+ and t_j^-. Denote the discriminative objective function as follows:

$$\mathbb{L} = \sum_{q\in Q} \sum_{i,j:x_i^+ \wedge x_{j,\theta}^-} \left[\frac{L(-(f_\phi(x_i^+, q) - f_\phi(x_{j,\theta}^-, q)))}{t_i^+ \cdot t_j^-} \right] + \lambda \|t^+\|_p^p + \lambda \|t^-\|_p^p \tag{4}$$

Here, $\lambda\|\cdot\|_p^p$ is L_p regularization term, with parameter $p > 0$ and $\lambda > 0$ controlling the degree of imposed regularization. We can optimize ϕ, t_i^+ and t_j^- iteratively. In each iteration, when the optimal parameters ϕ^* have been computed for several times, we fix ϕ and estimate the values of t_i^+ and t_j^- by partial derivative of the objective function in Eq.(4):

$$t_i^+ = \left[\frac{\sum_q \sum_{j:x_i^+, x_j^-} \left(L(-(f_{\phi^*}(x_i^+, q) - f_{\phi^*}(x_{j,\theta}^-, q)))/(t_j^-)^* \right)}{\sum_q \sum_{k:x_1^+, x_k^-} \left(L(-(f_{\phi^*}(x_1^+, q) - f_{\phi^*}(x_{k,\theta}^-, q)))/(t_k^-)^* \right)} \right]^{\frac{1}{p+1}} \tag{5}$$

$$t_j^- = \left[\frac{\sum_q \sum_{i:x_i^+,x_j^-} \left(L(-(f_{\phi^*}(x_i^+,q) - f_{\phi^*}(x_{j,\theta}^-,q)))/(t_i^+)^* \right)}{\sum_q \sum_{k:x_k^+,x_1^-} \left(L(-(f_{\phi^*}(x_k^+,q) - f_{\phi^*}(x_{1,\theta}^-,q)))/(t_k^+)^* \right)} \right]^{\frac{1}{p+1}} \tag{6}$$

The results calculated by Eq.(5)(6) are normalized to make the position bias at the first position to be 1. The pairwise function $L(-(f_\phi(x_i^+,q) - f_\phi(x_{j,\theta}^-,q)))$ is implemented as $log(1+exp(-(f_\phi(x_i^+,q)-f_\phi(x_{j,\theta}^-,q))))$ in our experiments. Then we use the updated ratios to optimize parameters ϕ. To speed up the training of the discriminator, we update the discriminator e-step times and update the position ratios once. This process iterates until convergence. The optimization algorithm of the discriminator is shown in Algorithm 1.

Optimizing Generator: In the minimax game, the generator is expected to minimize the objective function, fitting in the underlying relevance distribution via the signals from the discriminator:

$$\begin{aligned} J^{G^*} = \underset{\theta}{min}\underset{\phi}{max} \sum_{q\in Q} \mathbb{E}_{x_i \sim p_{true}(x,c_q^{x,i}=1)}[log(f_\phi(x_i,q))] \\ + \mathbb{E}_{x_j \sim p_\theta(x,c_q^{x,j}=0)}[log(1-f_\phi(x_j,q))] \end{aligned}$$

which can be converted into the following maximization function:

$$\begin{aligned} \theta^* &= \underset{\theta}{argmin} \sum_{q\in Q} \mathbb{E}_{x_j \sim p_\theta(x,c_q^{x,j}=0)}[log(1-f_\phi(x_j,q))] \\ &= \underset{\theta}{argmax} \sum_{q\in Q} \mathbb{E}_{x_j \sim p_\theta(x)}[log(f_\phi(x_j,q))] \end{aligned}$$

Inspired by the work IRGAN [28], we employ the policy gradient algorithm [27] for optimization. Following [28], we also constrain the reward function $D(x,q)$ in $(-1,1)$: $D(x,q) = 2\sigma(f_\phi(x,q)) - 1$. The gradient of the loss function is derived as:

$$\bigtriangledown_\theta\, J^G(q) \simeq \frac{1}{K}\sum_{k=1}^{K} \bigtriangledown_\theta log p_\theta(x_k) log(D(x_k,q)) \tag{7}$$

The true relevance distribution $p_{true}(x, c_q^{x,i})$ is dynamic and uncertain, which makes the equilibrium between the discriminator and the generator comparatively hard to reach. Convergence analysis in this problem is still an open question in current research literature [14, 28]. We summarize the overall learning algorithm of ULTRGAN in Algorithm 1.

5 Experiments

In this section, we conduct extensive experiments to evaluate ULTRGAN [3]. Specifically, we aim to answer the following research questions:
RQ1: Can ULTRGAN effectively and robustly estimate inherent relevance?
RQ2: Does ULTRGAN have a better performance, compared to other pairwise debiasing models?

[3] Code is available at `https://github.com/April-Cai/Debiasing-Learning-to-Rank-Models-with-GANs`

Algorithm 1: ULTRGAN

```
Input: click dataset $\mathbb{D} = \{q, \pi_q, c_q\}$, discriminator $f_\phi$, generator $p_\theta$;
Output: $\phi$, $\theta$, $t^+$, $t^-$;
Initialize $\phi$, $\theta$, $t^+ \Leftarrow 1$, $t^- \Leftarrow 1$;
repeat
    for d-step do
        Prepare training set $\mathbb{S} = \{(x_i^+, x_j^-)\}$ by using current $p_\theta$ to select $x_j^-$
          from unclicked documents for each clicked document $x_i^+$;
        Update $f_\phi$ with $t^+$, $t^-$ on $\mathbb{S}$ with Eq.(3);
        if d-step mod e-step = 0 then
            Update position ratios $t^+$, $t^-$ by with Eq.(5)(6);
        end
    end
    for g-step do
        Use current $p_\theta$ to select most relevant $K$ documents for each $q$;
        Update $p_\theta$ via policy gradient with Eq.(7);
    end
until ULTRGAN converges;
```

5.1 Experiments over Synthetic Dataset

Dataset and Experimental Settings. To our knowledge, the Yahoo! learning-to-rank challenge [8] is one of the largest datasets for LTR, which has been divided into training, valuation and test sets. In this dataset, each record contains a query-document identifier followed by a 700-dimension feature vector. The corresponding human-annotated relevance labels include Perfect (4), Excellent (3), Good (2), Fair (1) and Bad (0).

We follow the same settings as [1,15,18] to simulate exposures and clicks. A weak Ranking SVM model trained with 1% of the training data is used to create initial ranked list. Then clicks can be generated by simulating user behaviors based on the pre-defined click models [10]. Specifically, we use the position-based model [24] as follows:

$$P(c_i^+) = P(o_i^+) \cdot P(r_i^+)$$

$P(o_i^+)$ is the probability of a document being observed in the position i with $P(o_i^+) = \rho_i^\eta$ where ρ_i is derived from empirical results [18], and η controls the severity of position bias. We set $\eta = 1$ for the main experiment. The perceived relevance probability $P(r_i^+)$ is computed as:

$$P(r_i^+) = \epsilon + (1 - \epsilon)\frac{2^y - 1}{2^{y^{max}} - 1}$$

where $y \in \{0, 1, 2, 3, 4\}$ and y^{max} equals to 4 as the highest relevance score. The severity of relevance bias is controlled by ϵ. By default, we set ϵ as 0.1.

The discriminator and generator are implemented with deep neural networks (DNNs) that work with stochastic gradient decent (SGD). The hidden layers are

512, 256, 128 in size, remain consistent with [1]. Given input feature vector $\mathbf{x}$, the discriminative function $f_\phi(\mathbf{x})$ can be derived from the output of network a_4:

$$a_0 = \mathbf{x}; a_n = leakyReLU(W_{n-1}a_{n-1} + b_{n-1}), n = 1, 2, 3, 4$$

The generative function $p_\theta(\mathbf{x})$ can be derived from the output of network h_4:

$$\begin{aligned} h_0 =& \mathbf{x}; h_n = leakyReLU(W_{n-1}h_{n-1} + b_{n-1}), n = 1, 2, 3 \\ h_4 =& tanh(W_3 h_3 + b_3) \end{aligned}$$

Here, *leakyReLU* and *tanh* are commonly-used activation functions. Scalers a_4 and h_4 indicate predicted relevance score. For both networks, we set the learning rate as 0.05, the batch size as 256, and the weight decay as 1e-4. The parameter p is 0.05 and λ is 1 in Eq.(4). The number of sampled documents K is set as 5. We train the model with d-step as 100, g-step as 50 and e-step as 10 in Algorithm 1.

Baselines. We consider the following debiasing methods as baselines:

- *No Correct*: Directly treat clicks as labels, used as the lower bound.
- *Randomization*: The randomization-based model [18] for bias elimination.
- *Regression-EM*: A regression-based EM algorithm proposed in [30].
- *Dual Learning*: The dual learning algorithm (DLA) implemented in DNN [1] [4].
- *Pairwise Debiasing*: The pairwise debiasing model [15] for LambdaMart [6] [5].

In the experiments, Regression-EM and Randomization are implemented in three rankers (RankSVM, LambdaMart and DNN). DLA is bound to DNN only. Pairwise debiasing has only been employed in LambdaMart. Following [1,15], we use MAP and nDCG as the evaluation matrices. MAP and nDCG at 1, 3, 5 and 10 are reported.

Effectiveness Analysis. As shown in Table 2, our model outperforms all the baselines, and is as effective as (if not better than) the pairwise debiasing [15] of LambdaMart. We can observe that LambdaMart is the most effective base model when no corrections are conducted. However, ULTRGAN is superior to basic DNN, indicating that the combination of generative and discriminative retrieval models makes it possible to exceed the limit of the base model. Meanwhile, ULTRGAN is completely end-to-end, enhancing the discriminative function by sampling negative documents and updating position ratios dynamically. The estimated ratios are in accordance with former analysis shown in Fig.2. These factors give ULTRGAN an advantage over state-of-the-art debiasing methods.

Robustness Analysis. For fair comparison, we perform experiments on different debiasing methods implemented in DNN. The robustness of these models can be evaluated by varying η from 0.2 to 1.8. As shown in Fig.3, the overall ranking approaches become less effective with the bias getting much severer, which is in accordance with assumption. We can observe that our method provides robust performances compared with baselines. This indicates that ULTRGAN scales well and could be adapted to real-world conditions.

[4] `https://github.com/QingyaoAi/Unbiased-Learning-to-Rank-with-Unbiased-Propensity-Estimation`

[5] `https://github.com/acbull/Unbiased_LambdaMart`

Table 2: Comparisons of different unbiased learning-to-rank models.

Ranker	Debiasing Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
RankSVM	Regression-EM	0.815	0.629	0.648	0.674	0.705
	Randomization	0.814	0.628	0.644	0.672	0.707
	No Correct	0.811	0.614	0.629	0.658	0.697
LambdaMart	Pairwise Debiasing	0.836	0.717	0.716	0.728	0.764
	Regression-EM	0.830	0.685	0.684	0.700	0.743
	Randomization	0.827	0.669	0.678	0.690	0.728
	No Correct	0.820	0.658	0.669	0.672	0.716
DNN	Dual Learning	0.828	0.674	0.683	0.697	0.734
	Regression-EM	0.829	0.676	0.684	0.699	0.736
	Randomization	0.825	0.673	0.679	0.693	0.732
	No Correct	0.819	0.637	0.651	0.667	0.711
GAN	ULTRGAN	**0.842**	**0.722**	**0.718**	**0.730**	**0.766**

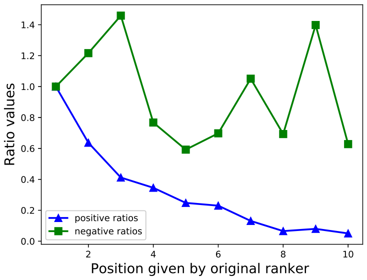

Fig. 2: Position biases (ratios) estimated by ULTRGAN.

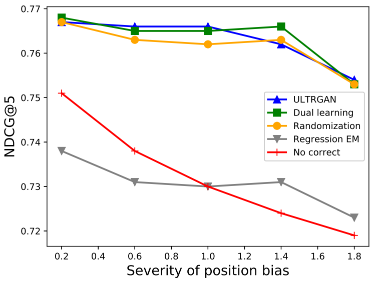

Fig. 3: The performances of different debiasing methods when η varies.

5.2 Experiments over Real-world Dataset

Dataset and Experimental Settings. We perform experiments on a real-world dataset named TianGong-ST [9]. It was collected from a commercial web search engine on a 18-day span search log, which contains 147,155 refined search sessions in total with clicks and positions. This dataset also provides a corpora covered over 90% web pages. For evaluation purpose, a test set of 2000 queries each with top 10 documents is attached, labeled manually in TREC style.

As in [1], we employ content-based algorithms to extract features based on the text of queries and documents. We use Lucene [6] to index and search. The 29 features extracted are as follows: the average term frequency (TF), the average inverse document frequency (IDF), the average $tf \cdot idf$ scores, the BM25 scores, the language model (LM) with Dirichlet smoothing and with Jelinek-Mercer scores [34], the number of terms, each feature calculated in title, URL, content and the whole document, together with the number of slashes in URL.

[6] `https://lucene.apache.org/`

We do stratified sampling by session lengths and acquire 13,484 ranked lists. For each query in the training and test sets, we remove candidates that are invalid or cannot be reached. We evaluate the ranking performances over the test set. The length of initial ranked list is 10 at most, therefore we report nDCG at 1, 3 and 5, respectively. As to the experimental settings, we vary the sizes of hidden layers to 16, 8. The d-step is set as 50, g-step as 10 and e-step as 10.

Baselines. We compare ULTRGAN against the following pairwise debiasing approaches:

- *No Correct*: Directly use clicks as labels in LambdaMart [6].
- *Unbiased LambdaMart*: Pairwise debiasing [15] in LambdaMart.
- *Unbiased DNN*: Pairwise debiasing in DNN.

Comparison and Analysis. As shown in Table 3, our model achieves the best performances compared to Unbiased LambdaMart [15] and Unbiased DNN, implying that our method has the advantage of sampling informative unlabeled instances instead of using all candidates to further optimize the ranking function.

Out of concern for the unstable training of GAN, we outline the learning curve of the discriminator as shown in Fig.4. Here, we only report the performances measured by nDCG@5, other matrices exhibit the similar trend. After training for 50 epochs, the model converges and consistently outperforms baselines. The results imply that our method can steadily achieve a high level of performance that is promising to be applied in production.

Table 3: Performances of different pairwise debiasing models on TianGong-ST.

Ranker	NDCG@1	NDCG@3	NDCG@5
No correct	0.663	0.715	0.771
Unbiased LambdaMart	0.674	0.725	0.776
Unbiased DNN	0.693	0.736	0.788
ULTRGAN	**0.698**	**0.749**	**0.798**

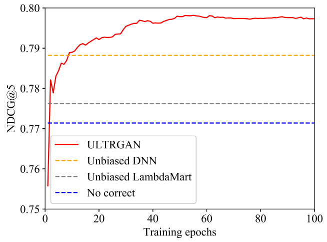

Fig. 4: Learning curves on TianGong-ST.

6 Conclusion and Future Work

In this paper, we formulate ULTR as a ranking problem under the semi-supervised setting. The incorporation of pairwise debiasing into generative adversarial networks better employs competitive negative instances for discriminative learning, which enables unbiased relevance supervision to propagate from the discriminator to the generator. In this way, propensity estimation and relevance learning can be performed at the same time. Empirical results demonstrate effectiveness and robustness of our approach.

This work represents an initial attempt to combine adversarial training mechanism with counterfactual learning and there are still many problems. For example, the sampling strategy is relatively inefficient and the equilibria could not be reached eaily. In the future, we plan to investigate other conditions such as pointwise and listwise ranking functions that could be extended to this framework. Model pre-training may bring in benefits, which is also left for future studies.

Acknowledgements

This work is supported by the National Key Research and Development Program of China under Grant No. 2016YFB1000904. We thank the anonymous reviewers for their careful reading and insightful comments on our manuscript.

References

1. Ai, Q., Bi, K., Luo, C., Guo, J., Croft, W.B.: Unbiased learning to rank with unbiased propensity estimation. In: SIGIR. pp. 385–394 (2018)
2. Ai, Q., Mao, J., Liu, Y., Croft, W.B.: Unbiased learning to rank: Theory and practice. In: CIKM. pp. 2305–2306 (2018)
3. Bekker, J., Davis, J.: Learning from positive and unlabeled data: A survey. arXiv preprint arXiv:1811.04820 (2018)
4. Borisov, A., Markov, I., De Rijke, M., Serdyukov, P.: A neural click model for web search. In: WWW. pp. 531–541 (2016)
5. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: ICML. pp. 89–96 (2005)
6. Burges, C.J.: From ranknet to lambdarank to lambdamart: An overview. Learning **11**(23-581), 81 (2010)
7. Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F., Li, H.: Learning to rank: from pairwise approach to listwise approach. In: ICML. pp. 129–136 (2007)
8. Chapelle, O., Chang, Y.: Yahoo! learning to rank challenge overview. In: Proceedings of the learning to rank challenge. pp. 1–24 (2011)
9. Chen, J., Mao, J., Liu, Y., Zhang, M., Ma, S.: Tiangong-st: A new dataset with large-scale refined real-world web search sessions. In: CIKM. pp. 2485–2488 (2019)
10. Chuklin, A., Markov, I., Rijke, M.d.: Click models for web search. Synthesis lectures on information concepts, retrieval, and services **7**(3), 1–115 (2015)
11. Cossock, D., Zhang, T.: Subset ranking using regression. In: COLT. pp. 605–619 (2006)
12. Craswell, N., Zoeter, O., Taylor, M., Ramsey, B.: An experimental comparison of click position-bias models. In: WSDM. pp. 87–94 (2008)
13. Dupret, G.E., Piwowarski, B.: A user browsing model to predict search engine click data from past observations. In: SIGIR. pp. 331–338 (2008)
14. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS. pp. 2672–2680 (2014)
15. Hu, Z., Wang, Y., Peng, Q., Li, H.: Unbiased lambdamart: An unbiased pairwise learning-to-rank algorithm. In: WWW. pp. 2830–2836 (2019)
16. Joachims, T.: Optimizing search engines using clickthrough data. In: KDD. pp. 133–142 (2002)

17. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: SIGIR. vol. 51, pp. 4–11 (2017)
18. Joachims, T., Swaminathan, A., Schnabel, T.: Unbiased learning-to-rank with biased feedback. In: WSDM. pp. 781–789 (2017)
19. Liu, T.Y., et al.: Learning to rank for information retrieval. Foundations and Trends® in Information Retrieval **3**(3), 225–331 (2009)
20. Lu, S., Dou, Z., Jun, X., Nie, J.Y., Wen, J.R.: Psgan: A minimax game for personalized search with limited and noisy click data. In: SIGIR. pp. 555–564 (2019)
21. Oosterhuis, H., Jagerman, R., de Rijke, M.: Unbiased learning to rank: Counterfactual and online approaches. arXiv preprint arXiv:1907.07260 (2019)
22. Oosterhuis, H., de Rijke, M.: Differentiable unbiased online learning to rank. In: CIKM. pp. 1293–1302 (2018)
23. O'Brien, M., Keane, M.T.: Modeling result–list searching in the world wide web: The role of relevance topologies and trust bias. In: Proceedings of the 28th annual conference of the cognitive science society. vol. 28, pp. 1881–1886. Citeseer (2006)
24. Richardson, M., Dominowska, E., Ragno, R.: Predicting clicks: estimating the click-through rate for new ads. In: WWW. pp. 521–530 (2007)
25. Rosenbaum, P.R., Rubin, D.B.: The central role of the propensity score in observational studies for causal effects. Biometrika **70**(1), 41–55 (1983)
26. Steck, H.: Training and testing of recommender systems on data missing not at random. In: KDD. pp. 713–722 (2010)
27. Sutton, R.S., McAllester, D.A., Singh, S.P., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: NeurIPS. pp. 1057–1063 (2000)
28. Wang, J., Yu, L., Zhang, W., Gong, Y., Xu, Y., Wang, B., Zhang, P., Zhang, D.: Irgan: A minimax game for unifying generative and discriminative information retrieval models. In: SIGIR. pp. 515–524 (2017)
29. Wang, X., Bendersky, M., Metzler, D., Najork, M.: Learning to rank with selection bias in personal search. In: SIGIR. pp. 115–124 (2016)
30. Wang, X., Golbandi, N., Bendersky, M., Metzler, D., Najork, M.: Position bias estimation for unbiased learning to rank in personal search. In: WSDM. pp. 610–618 (2018)
31. Yang, L., Cui, Y., Xuan, Y., Wang, C., Belongie, S., Estrin, D.: Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In: RecSys. pp. 279–287 (2018)
32. Yue, Y., Joachims, T.: Interactively optimizing information retrieval systems as a dueling bandits problem. In: ICML. pp. 1201–1208 (2009)
33. Yue, Y., Patel, R., Roehrig, H.: Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In: WWW. pp. 1011–1018 (2010)
34. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: ACM SIGIR Forum. vol. 51, pp. 268–276. ACM New York, NY, USA (2017)