Toward User Engagement Optimization in 2D Presentation

Liang Wu
Airbnb
San Francisco, CA
liang.wu@airbnb.com

Mihajlo Grbovic
Airbnb
San Francisco, CA
mihajlo.grbovic@airbnb.com

Jundong Li
University of Virginia
Charlottesville, VA
jundong@virginia.edu

ABSTRACT

Given a collection of items to display, such as news, videos, or products, how can we optimize their presentation order to maximize user engagements, such as click-through rate, viewing time, and the number of purchases? The problem becomes more complicated when the items are displayed in a grid-based, 2-dimensional presentation on a widescreen. For example, many E-Commerce websites such as Amazon and Etsy are displaying their products in a grid-like format, and so are streaming services like Youtube and Netflix. Unlike 1-dimensional space, where products can be naturally ranked in a vertical order, the presentation in 2-dimensional space poses a novel challenge about how to find the best presentation order - should we put the best listing on the top left corner, or the central position on the second row? We are aware that many traditional methods can be applied to solve the problem, such as conducting an attention heatmap web test, or a randomization experiment by shuffling positions of listings. However, both tests are costly to perform and they may downgrade the quality of users' search experience. By contrast, we focus on utilizing existing search log data to reveal propensity of positions, which is readily available and ubiquitously abundant.

In a nutshell, the study presents how we find an optimal way of presentation in a grid-based environment - more relevant content should be placed in a more noticeable position. The position noticeability is further quantified to help ranking models better understand the signal of relevance manifested in user feedbacks. Our investigation paves the way for an end-to-end item presentation framework that learns the optimal layout for optimizing user engagements. Experimental results based on real-world data show the superiority of the proposed approach over state-of-the-art methods.

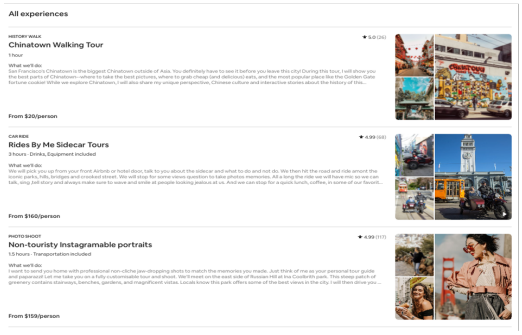
KEYWORDS

E-Commerce, Optimization, Natural Experiment

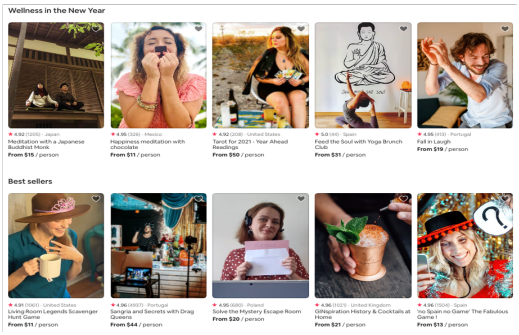
ACM Reference Format:
Liang Wu, Mihajlo Grbovic, and Jundong Li. 2021. Toward User Engagement Optimization in 2D Presentation. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441749

WSDM '21, March 8–12, 2021, Virtual Event, Israel

ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441749



(a) 1-D presentation of search results: listings are ranked vertically.



(b) 2-D presentation of search results: listings are displayed in a grid-based format.

Figure 1: Examples of search results displayed in 1-D and 2-D formats.

1 INTRODUCTION

2-dimensional presentation is ubiquitous in online applications. For example, most E-Commerce search platforms, such as Taobao, Amazon and Ebay, would present content in a grid-based format on widescreens. However, existing applications are heavily influenced by general web search, where users are assumed to browse items in a top-down, 1-dimensional manner. Figure 1 shows examples of 1-D and 2-D presentations of search results. As illustrated in Figure 1(a), customers who search naturally scan results from top to bottom. When items are displayed in a 2-dimensional grid space as shown in Figure 1(b), users may not start their browsing from the top left corner. In this work, we aim to investigate how the 2-dimensional presentation would influence user behaviors and how an optimal relevance ranking algorithm can be learned. In particular, we will focus on the problem of E-Commerce search, where items are products.

It can be time-consuming and labor-intensive to manually investigate users' browsing behaviors. Therefore, we propose to solve the problem based on the massive amount of search log data. Previous literature shows that search log data can help us understand how users would distribute their attention on different positions of a search result page [3, 13, 16, 23, 26, 27]. Existing methods aim to tackle the challenge of presentation bias in web search, and a main intuition here is to consider how positions would affect users' behaviors. For example, a click may be driven by both the relevance between user intent and search results, and how the position would attract attention of the user. Similarly, we aim to utilize product search log data to identify the browsing patterns of online customers. Search log data also allows for convenient extension to a personalized approach by soliciting personal data.

However, solutions from web search cannot be directly applied due to the intrinsic distinctions between product search and web search. Early work requires an intervention experiment to be conducted to estimate propensity of each position [16], *e.g.*, we may swap results on two positions to estimate the relative influence of presentation bias. However, such intervention experiments unavoidably influence search experience in a negative way since it requires a large number of tests to reduce the variance. Later research proposes to directly estimate the influence using search log data, assuming the propensity is a variable associated to each position [2, 3, 13, 27]. However, they assume that search results are vertically ranked in a list, while E-Commerce sites usually display products in a 2-dimensional, grid-based space.

A particular challenge of 2-dimensional presentation is that it hinders convergence of classic 1-dimensional approaches. To estimate the influence of each position, which is also called propensity [16], an end-to-end method will have to iteratively estimate the nested two problems [3, 13], *i.e.,* relevance estimation and propensity estimation. Note that the convergence of one problem relies on the convergence of the other one. Therefore, it is less challenging to control the convergence in 1-dimensional space since the propensity is naturally monotonically decreasing (users browse from top to bottom). In a 2-dimensional grid-based space, the number of parameters for propensity prediction increases exponentially, and it is difficult to assume fashions of browsing.

In this work, we present an end-to-end solution for product ranking in E-Commerce search by dealing with presentation bias in a 2-dimensional space. The proposed method utilizes search log data and user feedback signals including clicks and conversions. This is a nontrivial task due to the nested nature of the two tasks, *i.e.,* propensity prediction and relevance prediction. Our proposed method leverages causal inference to dismantle the influence of position from relevance. Contributions can be summarized as

- We define a new problem of optimizing 2-dimensional product ranking in E-Commerce search, where items are displayed in a grid-based space;
- We propose an end-to-end approach that utilizes only search log data to rank items in 2-d presentation without additional intervention experiments.
- We conduct extensive experiments with real-world data to validate the performance of the proposed algorithm against competitive baseline methods.

The rest of the paper is organized as follows. In Section 2 we introduce the problem of 2-dimensional product search in E-Commerce, and formally define the computational problem. In Section 3, we present the presentation bias problem in product search and present the proposed framework. Experiments will be discussed in Section 4. Section 5 will discuss related work. Section 6 will conclude the paper and provide future directions.

Table 1: Notations and Description

Notation	Description
$\mathbf{I}, \mathbf{Q}, \mathbf{S}, \mathbf{M}$	Set of items, queries, sessions, impressions
$C(i, q)$	Click information where $C(i, q) = 1$ means i was clicked
$CTR(i, q)$	Click-through rate of an item given query q
R_{max}, C_{max}	The maximum number of rows and columns on a search result page
p_r	Propensity score of position where row = r
$p_{r,c}$	Propensity score of position where row = r and column = c
$r(q, i)$	Ground-truth relevance score of between a pair of q and i
$P(rel = 1\|i, q)$	The probability of i and q being relevant
$m = (q, i, rel, b, p)$	An impression with its query, item, relevance, user behavior and position. We will use $m.q$, $m.i$, *etc.*

2 PROBLEM STATEMENT

When a query $q \in Q$ is submitted, a subset of items will be selected as candidates for being displayed to the user. The problem of sorting the candidates is a typical post-ranking problem that we focus on in this work. We assume the display space is 2-dimensional, where $r \leq R_{max}$ is the id of row and $c \leq C_{max}$ is the id of column and R_{max} and C_{min} denote the maximum number of rows and columns, respectively. Each position is associated with a propensity score $p_{r,c}$. A larger propensity score indicates that the position is more likely to be viewed/clicked by a user. For example, the position with highest propensity is the position that attracts most user attention instead of the top left one. Notations with descriptions throughout the work are presented in Table 1.

Given user search log data, we aim to learn a function $f(\cdot)$ to generate a ranking score for each item i with query q, and a function $g(\cdot)$ to estimate the propensity value for each position $p_{r,c}$. Existing search log data may be biased by the presentation, where an item at position (r, c) is more/less likely to be clicked by a factor of $p_{r,c}$.

3 TWO-DIMENSIONAL PRODUCT RANKING

3.1 One-Dimensional Ranking for Web Search

In the area of search ranking, label information is conventionally generated by search experts who manually evaluate each result for a query [20]. Therefore, we can assume that an oracle function is available for providing the ground-truth relevance score $r(q, i)$ for

each query-item pair, and pair-wise and list-wise ranking methods $f(\cdot)$ can directly be tuned with the relative and complete order being generated with $r(\cdot)$. Therefore, the label information $r(q, i)$ is generated in an unbiased way by manual annotation of search experts. The resultant model is also unbiased.

3.2 Propensity in One-Dimensional Search

A conventional practice in E-Commerce is to utilize user clicks to determine the label. Intuitively, if an item is relevant to the query, it is more likely to be clicked. Therefore, instead of employing manual annotators, metrics derived from logs, such as clicks and conversions are widely used for product search [17, 29]. In order to utilize metrics derived from search logs, the relevance between an item and a query is usually assumed to be proportional to the probability of click (or conversion, purchases *etc.*),

$$P(rel = 1|i \in \mathbf{I}, q \in \mathbf{Q}) \propto Prob[C(i, q) = 1], \tag{1}$$

where $Prob[C(i, q) = 1]$ can be observed from the production system, and the probability of being relevant $P[rel = 1|i \in \mathbf{I}, q \in \mathbf{Q}]$ is what we try to estimate. The main intuition is that a relevant item will be more likely to be clicked by a user. However, user behaviors can be heavily biased by the production system, which is called presentation bias or position bias, and an item ranked higher by the production system is more likely to be clicked [8]. In order to deal with presentation bias, we can explicitly integrate the effect of positions, and the corresponding formulation can be written as,

$$P(rel = 1|i \in \mathbf{I}, q \in \mathbf{Q}) \times p_r \propto Prob[C(i, q) = 1], \tag{2}$$

where the product of relevance and the propensity p_r is proportional to the label of a click. The weight p_r is usually larger when the position is higher (the position r is smaller). The weight has also been referred as propensity scores recently [16]. For rest of the paper, we will use propensity scores and position weight interchangeably. Note that the hidden assumption of using p_r is that all results are ranked on a 1-dimensional space. In E-Commerce applications, results are usually rendered in a 2-dimensional space, and the propensity score can be denoted as $p_{r,c}$, instead.

3.3 Unbiased Two-Dimensional Ranking

We now introduce how presentation bias can be estimated in a 2-dimensional space. By replacing 1-dimensional propensity p_r with $p_{r,c}$, the formulation in Eq.(2) can be rewritten as,

$$P(rel = 1|i \in \mathbf{I}, q \in \mathbf{Q}) \times p_{r,c} \propto Prob[C(i, q) = 1], \tag{3}$$

where the propensity score $p_{r,c}$ is correlated with both row and column information. We assume that propensity score along two different dimensions are independent, and the formulation can be further rewritten as,

$$P(rel = 1|i \in \mathbf{I}, q \in \mathbf{Q}) \times p_r \times p_c \propto Prob[C(i, q) = 1], \tag{4}$$

where the propensity score $p_{r,c}$ is decomposed into two factors p_r and p_c. The decomposition helps reduce the total number of parameters to estimate and we may better predict each of them individually. If there is no presentation bias, our goal is reduced to finding the optimal $f(\cdot)$ that minimizes the empirical risk,

$$\min_f Risk(f(\cdot)|\mathbf{M}) = \frac{1}{|\mathbf{M}|} \sum_{(q,i)\in\mathbf{M}} \ell[f(q, i)|P(rel = 1|i, q)], \tag{5}$$

where $\mathbf{M}$ includes all impressions and $\ell(\cdot)$ represents a loss function over prediction and ground truth. In order to incorporate the induced propensities, we adopt Inverse Propensity Scoring (IPS) [16] and the formulation can be rewritten as,

$$\min_{f,p_r,p_c} Risk(f(\cdot)|\mathbf{M}) = \frac{1}{|\mathbf{M}|} \sum_{(q,i)\in\mathbf{M}} \frac{\ell[f(q, i)|P(rel = 1|i, q)]}{p_r \times p_c}, \tag{6}$$

where each instance is weighted by the inverse value of its propensity. IPS was first introduced to solve position bias problem in web search by Joachims *et al.* and it has been theoretically proven to provide an unbiased estimation [16]. Since the propensity score of a lower position is usually smaller, the main intuition here is that the positive feedback of a less "visible" data instance should be higher weighted. Therefore, in order to apply IPS with a particular learning algorithm, $\frac{1}{p_r \times p_c}$ can be incorporated as a weighting term that can be directly applied for most learning methods. A recent work discussed how the weights can be used in a deep model [2]. We will discuss how we apply it in a gradient boosting method. Since pairwise approaches usually perform better than point-wise methods, we extend Eq.(6) into a pairwise manner,

$$\min_{f,p_r^+,p_c^+,p_r^-,p_c^-} Risk(f(\cdot)|\mathbf{M}) = \frac{1}{|\mathbf{M}|} \sum_{(q,i^+,i^-)\in\mathbf{M}} \frac{\ell[f(q, i^+), f(q, i^-)]}{p_r^+ \times p_c^+ \times p_r^- \times p_c^-}, \tag{7}$$

where i^+ and i^- are a pair of items for query q and $C(i^+) = 1$ while $C(i^-) = 0$; p_c^+ and p_r^+ denote the propensity values of i^+ and p_c^- and p_r^- denote those of i^-. Motivated by a recent work on debiasing LambdaMART [13], we differentiate p^+ from p^- for the same position. Instead of optimizing each item independently, we focus on relative orders of items here. Therefore, the 2-dimensional unbiased ranking problem can be reduced to minimizing the following loss function,

$$\min_{f,p_r^+,p_c^+,p_r^-,p_c^-} \sum_{(q,i^+,i^-)\in\mathbf{M}} \frac{\ell[f(q, i^+), f(q, i^-)]}{p_r^+ p_c^+ p_r^- p_c^-} + \lambda \cdot \alpha(p_r^+, p_c^+, p_r^-, p_c^-), \tag{8}$$

where we introduce a regularizer $\alpha(\cdot)$ to control the model complexity of propensity estimation, and λ controls the extent of how we control model complexity. A larger λ leads to a simpler model. The denominator $|\mathbf{M}|$ is omitted here since we assume data is uniformly drawn. The new formulation enables us to model a 2-dimensional search result page. We have multiple variables to optimize in Eq.(8), and the problem is not convex *w.r.t.* all three. We will iteratively optimize each of them separately and details of optimization will be covered in the next subsection.

3.4 Optimization

The optimization problem in Eq.(8) is convex *w.r.t.* a single variable when the rest is being fixed. We will first fix the ranker $f(\cdot)$ and try to learn the parameters of propensity estimation. The partial derivatives of the objective function in Eq.(8) *w.r.t.* propensity values can be formulated as,

$$\frac{\partial Risk}{\partial p_r^+} = \sum_{(q,i^+,i^-)\in\mathbf{M}} \frac{\ell(f(q, i^+), f(q, i^-))}{-p_r^{+,2} p_c^+ p_r^- p_c^-} + \lambda \frac{\partial \alpha}{\partial p_r^+}, \tag{9}$$

$$\frac{\partial Risk}{\partial p_c^+} = \sum_{(q,i^+,i^-)\in\mathbf{M}} \frac{\ell(f(q,i^+),f(q,i^-))}{-p_r^+ p_c^{+,2} p_r^- p_c^-} + \lambda\frac{\partial \alpha}{\partial p_c^+}, \tag{10}$$

$$\frac{\partial Risk}{\partial p_r^-} = \sum_{(q,i^+,i^-)\in\mathbf{M}} \frac{\ell(f(q,i^+),f(q,i^-))}{-p_r^+ p_c^+ p_r^{-,2} p_c^-} + \lambda\frac{\partial \alpha}{\partial p_r^-}, \tag{11}$$

$$\frac{\partial Risk}{\partial p_c^-} = \sum_{(q,i^+,i^-)\in\mathbf{M}} \frac{\ell(f(q,i^+),f(q,i^-))}{-p_r^+ p_c^+ p_r^- p_c^{-,2}} + \lambda\frac{\partial \alpha}{\partial p_c^-}, \tag{12}$$

where we estimate one of the parameters while keeping the rest and $f(\cdot)$ fixed. Similarly, by keeping all parameters for propensity estimation fixed, we have the derivatives for learning the ranking function $f(\cdot)$ as,

$$\frac{\partial Risk}{\partial f} = \sum_{(q,i^+,i^-)\in\mathbf{M}} \frac{\partial \ell(f(q,i^+),f(q,i^-))}{\partial f(\cdot)} \frac{1}{p_r^+ p_c^+ p_r^- p_c^-}, \tag{13}$$

where the derivative of the ranking model is inversely weighted by the propensity score $p_r^+ p_c^+ p_r^- p_c^-$, *i.e.*, a larger ranking loss will lead to a smaller propensity score, and vice versa.

Based on the above derivatives, it is easy to obtain update rules when a certain ranking model and regularization terms are selected. In this work, we will adopt ℓ_2-norm as the regularization term. Therefore, the closed-form solution of the propensity value can be obtained by setting the derivatives in Eq.(9) - (12) as zero. We adopt LambdaMART [6] as the ranking model. The algorithm is shown in Algorithm 1. We first initialize parameters for propensity prediction in step 1. The initialization here is particularly important since the two optimization tasks are nested and will rely on the results. A bad starting point may sufficiently prevent all subsequent tasks to converge to an optimal point. We will elaborate more in the next subsection, introducing how we initialize the parameters using causal inference. Step 3 optimizes for the parameters of the ranking function $f(\cdot)$, and in step 4 we update the parameters for propensity scores. The early termination rule can be defined based on a validation dataset in step 5.

Algorithm 1 Algorithm for Two-Dimensional Product Ranking

Input: Search log data: $\mathbf{M}$, λ, Maximum number of iterations $MaxIter$, Early termination condition EM

Output: Ranker: $f(\cdot)$, Propensity scores: $p_r^+, p_c^+, p_r^-, p_c^-$

1: Initialize $p_r^+, p_c^+, p_r^-, p_c^-$,
2: For $iter$ from 1 to $MaxIter$:
3: Update $f(\cdot)$ based on Eq.(13)
4: Update $p_r^+, p_c^+, p_r^-, p_c^-$ based on Eq.(9) - (12):
5: If meets EM:
6: Break
7: Return $f(\cdot)$, $p_r^+, p_c^+, p_r^-, p_c^-$

In the next subsection, we will introduce how we can initialize the parameters in step 1. This is particularly important since we have two nested tasks here, *i.e.,* estimating propensity scores and learning a ranking function, and convergence of one task relies on the convergence of another. We investigate different possibilities and propose a novel method based on causal inference.

3.5 Propensity Estimation with Intervention

Since the four propensity scores $p_r^+, p_c^+, p_r^-, p_c^-$ can be similarly initialized, we will omit the superscripts and subscripts here. A classic method to estimate presentation bias is to use randomized experiments, where we may randomly shuffle positions of all candidate results. Therefore the empirical CTR from randomized experiments can be directly used to estimate presentation bias [8]. Pair-wise intervention experiments have been studied to infer the propensity score in web search [16].

Pair-wise tests are relatively easier to perform since they require a fewer number of interventions. For a ranking list of length K, exactly $K-1$ swap experiments are needed. We will use an example to introduce how the experiment can be done in more details. Consider there are 3 examples being ranked by relevance in a descending order. Assume function $g(\cdot)$ predicts the probability that an item would be clicked with given relevance rel. We set the propensity score at the top position p_1 as 1. By incorporating the function with Eq.(2), the expected CTR of of the first item in the ranking list, $CTR(i_1@1)$, can be estimated as,

$$E(CTR(i_1@1)) = g(rel) \times p_1, \tag{14}$$

where $CTR(i_1@1)$ means the click through rate of first item i_1 being ranked at the first position (also its original position in this case), and $g(rel)$ can be regarded as the unbiased expected CTR and the propensity score p_1 introduces the influence of presentation bias. After N impressions, the empirical CTR can be estimated as,

$$CTR(i_1@1) = \sum_N \frac{g(rel) \times p_1}{N}, \tag{15}$$

where $\sum_N g(rel) \times p_1$ can be viewed as the expected number of clicks after N impressions. In order to estimate propensity score for the second position p_2, an intervention experiment can be performed to swap the position of i_1 and i_2. Similarly, an empirical CTR of item i_1 being ranked at the second position can be calculated,

$$CTR(i_1@2) = \sum_N \frac{g(rel) \times p_2}{N}, \tag{16}$$

where the unbiased number of clicks $\sum_N g(rel)$ is discounted by p_2. A nice property of the formulation is that, p_2 can be directly obtained by using the two empirical CTRs as,

$$p_2 = \frac{CTR(i_1@2) \times p_1}{CTR(i_1@1)}, \tag{17}$$

where all other terms except propensity can be eliminated. This is also a recursive rule that can help infer the propensity of all positions. In this case, we will need $2 = 3-1$ experiments to estimate propensity scores for a list of size 3. Though swap experiments only require $K-1$ experiments, a practical challenge is that a large N is needed to reduce the variance of the estimation of CTR, which is costly for many emerging applications. By contrast, we will focus on using only search log data.

3.6 Propensity Score Initialization with Causal Inference

Consider a special case of search ranking, where two adjacent items have exactly the same relevance score, *i.e.,* $rel(i_1) = rel(i_2)$. Items in a search result page are ordered with relevance scores, and an

item will be randomly ranked higher than the other if they have the same score. In such cases, the difference of feedback signals like CTR will be only affected by presentation bias. We may start with estimating p_2 with such special cases, and the approach can be recursively applied to lower positions similarly.

First, we collect pairs of impressions $\mathbf{M} = \{(m_1^s, m_2^s)\}$ in each search session s, where the top 2 items have the same relevance score ($m_1^s.rel = m_2^s.rel$). s denotes a session and m_1^s represents the impression of a first-ranked item in a session s. $m_1^s.rel$ denotes the relevance score of the impressed item and $m_1^s.p$ denotes the position. $\mathbf{M}$ represents the set of impression pairs that have the same relevance score on top 2 positions. We further denote $\mathbf{M_1}$ as the set of impressions in $\mathbf{M}$ that have been displayed on the first position and $\mathbf{M_2}$ as the set of impressions on the second position. Therefore, the assignment of positions for pairs in $\mathbf{M}$ can be viewed as an instrumental variable [4] and $\mathbf{M}$ can be used to infer p_2.

Instrumental variables approaches have been widely used to estimate causal relationships in disciplines of statistics, econometrics, *etc.* An instrumental variable randomly separates subjects into control and treatment groups, and a natural experiment can be accordingly conducted for causal inference [7]. Since the assignment of ranking orders in $\mathbf{M}$ is random, the estimation of p_2 can be done with the natural experiment using $\mathbf{M}$. We first represent the estimation of CTR for two groups,

$$CTR(\mathbf{M}_1) = \sum_{m \in \mathbf{M}_1} \frac{m.b}{|\mathbf{M}_1|} = \sum_{m \in \mathbf{M}_1} \frac{g(m.rel) \times p_1}{|\mathbf{M}_1|} \tag{18}$$

$$CTR(\mathbf{M}_2) = \sum_{m \in \mathbf{M}_2} \frac{m.b}{|\mathbf{M}_2|} = \sum_{m \in \mathbf{M}_2} \frac{g(m.rel) \times p_2}{|\mathbf{M}_2|}, \tag{19}$$

where $m.b = 1$ if the impression has been clicked by the user otherwise $m.b = 0$. Since for a particular pair, their relevance scores are the same $m_1^s.rel = m_2^s.rel$, and $|\mathbf{M}_1| = |\mathbf{M}_2|$, by dividing the two equations, we can easily estimate p_2,

$$\frac{CTR(\mathbf{M}_1)}{CTR(\mathbf{M}_2)} = \sum_{(m_1, m_2) \in \mathbf{M}} \frac{g(m_1.rel) \times p_1}{g(m_2.rel) \times p_2} = \frac{p_1}{p_2}, \tag{20}$$

where all other terms are eliminated and the propensity score can be recursively obtained with $w(i+1) = \frac{w(i) \times CTR(\mathbf{M}_{i+1})}{CTR(\mathbf{M}_i)}$. An obvious drawback of the natural experiment is that the required special cases can be rare and a small number of cases may introduce large variance to the estimation. Next, we will present how we adopt fuzzy Regression Discontinuity Design to deal with the issue.

In order to expand $\mathbf{M}$, we introduce a threshold of relevance and collect pairs of adjacent impressions that are with *similar* relevance scores. Similarly, we will follow the example of estimating p_2 for simplicity of presentation. Eq.(20) can be reformulated as,

$$\frac{CTR(\mathbf{M}_1)}{CTR(\mathbf{M}_2)} = \sum_{(m_1, m_2) \in \mathbf{M}} \frac{g(m_1.rel) \times p_1}{g(m_2.rel) \times p_2}, \tag{21}$$

where $g(m_1.rel)$ and $g(m_2.rel)$ cannot be eliminated due to the difference of relevance. In order to move forward, the first step is to learn the function $g(rel(i,q))$. Following a conventional practice [1], we will adopt the ranking score $rs(i,q)$ produced by the production system to represent the value of $rel(i,q)$.

Learning $g(rs(i,q))$ In order to learn the function $g(rs(i,q))$, we collect *all* impressions on the first position. We denote the set of impressions as $\mathbf{M}_{all}$. The reason we focus on the first position is that, given p_1 is set as 1, we will have

$$CTR(\mathbf{M}_{all}) = \sum_{m \in \mathbf{M}_{\text{all}}} \frac{g(m.rel)}{|\mathbf{M}_{all}|}, \tag{22}$$

where the CTR can be an unbiased estimation of $\sum_{m \in \mathbf{M}_{\text{all}}} \frac{g(m.rel)}{|\mathbf{M}_{all}|}$. Next, we group impressions into different bins of ranking scores. Let $\mathbf{M}_{bin}$ be a bin of impressions where $\{\forall m \in \mathbf{M}_{bin} | \mathbf{M}_{bin}.lower \leq m.rs < \mathbf{M}_{bin}.upper\}$. $m.rs$ denotes the ranking score of an impression. $\mathbf{M}_{bin}.lower$ and $\mathbf{M}_{bin}.upper$ are boundaries of ranking scores of a particular bin. Since the bin size ($\mathbf{M}_{bin}.upper - \mathbf{M}_{bin}.lower$) can be small, we make a relaxed assumption that impressions falling in the same bin have the same ranking score, $m_i.rel = m_j.rel, \forall m_i, m_j \in \mathbf{M}_{bin}$. Estimation of CTR in this bin can be formulated as,

$$CTR(\mathbf{M}_{bin}) = \sum_{m \in \mathbf{M}_{bin}} g(m.rel)/|\mathbf{M}_{bin}| \tag{23}$$

$$= \frac{|\mathbf{M}_{bin}| \cdot g(m.rel)}{|\mathbf{M}_{bin}|} \tag{24}$$

$$= g(m.rel) = f\left(\frac{\sum_{m \in \mathbf{M}_{bin}} m.rel}{|\mathbf{M}_{bin}|}\right), \tag{25}$$

where the average ranking score is used for a bin. By binning impressions, the CTR of each bin contributes a data point to learn the function $g(rel)$. We use Weighted Linear Regression (WLS) here where each bin is weighted by the number of impressions $|\mathbf{M}_{bin}|$. The propensity can then be estimated with Eq.(20). Here, the function $g(rs(i,q))$ can be viewed as a calibration method that estimates the probability of being positive given a ranking score.

3.7 Summary

The proposed **end-to-end framework** consists of two main components. The first component initializes parameters based on a natural experiment: (1) We design natural experiments to estimate propensity of each position with search log data as described in Section 3.6; (2) The ranker and propensity scores can be learned through iterative optimization.

The **convergence** of the method can be theoretically proven. The two subtasks are both convex as shown in Eq.(9) - Eq.(13). In addition, the objective function in Eq.(8) has lower bounds(*e.g.,* zero), the iterative optimization in Algorithm 1 converges. The **time complexity** for learning the ranker is similar to that of the selected ranking model, which is usually optimized through gradient-based methods [20]. Propensity estimation can also be efficiently optimized with closed-form solutions based on chosen regularizers. The optimal layout can also be directly derived based on the propensity scores - where a position with higher propensity scores indicates that results ranked here will gain more visibility.

4 EXPERIMENTS

In this section, we will present experiments we conducted to validate the effectiveness of the proposed approach. In particular, we focus on answering the following questions,

(1) How effective is the proposed method comparing with existing biased and unbiased product ranking methods?

Table 2: Statistics of the search log dataset for training.

	Impressions	Items	P:N Ratio
Logged-in	40,901,611	90,000	20%: 80%
Logged-out	47,187,300	90,000	9%: 91%

Table 3: Statistics of the unbiased dataset for evaluation.

	Impressions	Items	P:N Ratio
Logged-in	6,130	1,839	35%: 65%
Logged-out	7,818	2,218	11%: 89%

Table 4: Methods implemented for comparison in this work. Traditional methods of randomized experiments, Dual Learning Algorithm, traditional natural experiments, and the proposed method of natural experiments with calibration are adopted.

	Method	Acronym
Conventional Ranking	SVMRank [14]	SR
	Gradient Boosting Decision Tree [10]	GBDT
	LambdaMART [6]	LM
Unbiased Ranking	Dual Learning Algorithm [3]	DLA
	SVMRank-IPS [16]	SRI
	Trust Unbiased Ranking [1]	TU
	Unbiased LambdaMART [13]	UL
	2-D Dual Learning Algorithm	DLA^2
	2-D Trust Unbiased Ranking	TU^2
	2-D Unbiased LambdaMART	UL^2
Proposed Methods	2-D Unbiased Ranker	UR
	2-D Unbiased Ranker with Initialization	UI
	2-D Unbiased Ranker with Calibrated Initialization	UC

(2) How is product ranking model working differently by integrating 2-dimensional propensity and initialization?

4.1 Datasets

The search log data is obtained from Airbnb, an online marketplace. We randomly subsample data to build a dataset with both logged-in users and logged-out-users. There are in total 731, 869 search sessions with 90, 000 items being displayed. We also calculate the ratio of positive versus negative (P:N ratio) impressions where a positive impression leads to a conversion while a negative one does not. Detailed statistics are shown in Table 2.

We choose to differently model **logged-in** users and **logged-out** users due to their distinct browsing behaviors. Generally speaking, logged-in users are users with higher intents to booking listings, and their behaviors are more straightforward towards conversion. On the other hand, logged-out users are usually of lower intents. It is a conventional practice in industry to separately model them when they have very distinct behavioral patterns [28]. In our experiment, logged-in users and logged-out users will have two independent models, and logged-in users may have some additional features than loggeed-out users.

In order to evaluate performance, we will also need an unbiased dataset that is different from the one for training the model. A main assumption of presentation bias research is an unclicked item might have not been checked/seen by a user. Therefore, we build a dataset by extracting items that are ranked higher than the lowest clicked position in a search session. The resultant dataset is relatively small and is only used for testing a model. Detailed statistics about the dataset are shown in Table 3. Due to the small size of the unbiased dataset, it is usually only used for evaluation instead of training. We are aware that there are public datasets with unbiased labels and they have been used in previous research. The reason we do not include such datasets is that they lack the online scoring information for conducting a natural experiment.

4.2 Experimental Settings

We follow classic IR research and adopt Normalized Discounted Cumulative Gain (NDCG) to evaluate the ranking results. Since we focus on individual search sessions where the label information is likely to be binary, we also adopt Area Under Curve (AUC) of ROC which is widely used for classification tasks with skewed datasets. In addition, we also propose to measure the diversity of results as an offline experiment. We posit that ignoring the influence of presentation layout may unfairly bias the production model to focus on the few top items. Therefore, we aim to compare the variety of positive results of the production model with the proposed approach. A metric for measure diversity is defined as Normalized Diversity at K, $NDiv@K$. We assume that a positive instance will be booked under the new ranking order if it is ranked among top K. Therefore, the Diversity at K ($Div@K$) can be estimated as

$$Div@K = \sum_{i \in \mathbf{TP@K}} -\log(\frac{|i \in \mathbf{TP@K}|}{|\mathbf{P}|}) \times \frac{|i \in \mathbf{TP@K}|}{|\mathbf{P}|}, \qquad (26)$$

where $\mathbf{P}$ is the set of items that have been positive in any search session and $\mathbf{TP}$ is the set of true positives of an algorithm at top K positions. Given i is an item in $\mathbf{TP}$, $\frac{|i \in \mathbf{P}|}{|\mathbf{P}|}$ calculates the probability of the item and $Div@K$ estimates the diversity of results using information entropy. We smooth the probability of each item by setting the initial probability as $\frac{1}{|TP|}$ instead of *0*. We then calculate the maximum entropy with the ground truth data $Div_{max}@K$ and obtain the Normalized Diversity at K ($NDiv@K$) by $NDiv@K = Div@K/Div_{max}@K$.

The methods we adopted in the experiments fall into three categories: classic ranking methods that do not consider the influence of presentation bias, state-of-the-art unbiased ranking methods that debias the search log data in a 1-dimensional space, the proposed methods with two of its variants that remove the initialization module. All methods with their acronyms are illustrated in Table 4. For the first category, we include three ranking methods including SVMRank, Gradient Boosting Decision Tree (GBDT), and LambdaMART. These three methods are based on comparing pairs of items in a ranked list where SVMRank uses Support Vector Machines as a base classifier, and GBDT uses decision trees with gradient boosting, and LambdaMART adopts the λ trick to weight data pairs. These three methods have been popular in the both academia and industry and have been winners in several competitions of search ranking.

The second category of methods are recently proposed approaches that aim to estimate propensity scores with search log data. We include Dual Learning Algorithm (DLA), SVMRank with Inverse Propensity Scoring (SRI), Trust-Unbiased Ranking (TU), and Unbiased LambdaMART (UL) in this work. Since user feedback signals can be regarded as a product of presentation bias and

relevance, they iteratively optimize the two, *i.e.,* updating one with the other being fixed. Relevance is modeled as a classic ranking problem while presentation bias is modeled with only positions. These methods require both of the dual problems to converge in the early stage, which cannot be theoretically guaranteed. The only exception here is SRI, which requires an intervention experiment to be conducted for estimating propensity scores beforehand. Therefore, we collect propensity estimation results according to conventional practices [8], where randomly shuffled results are used to estimate propensity scores for SRI. This is different from the natural experiment we propose and it won't be updated.

We also include the 2-dimensional variants of methods in the second group to validate the effectiveness of the proposed method. Instead of learning a 1-dimensional propensity scores for DLA, UL and TU, we assume the position bias is the product of horizontal bias and vertical bias. Therefore, each algorithm will learn two propensity score vectors, and the final propensity is estimated with the two independent vectors.

The third group of methods include the proposed approach along with its variants. 2-D Unbiased Ranker is the proposed method without using initialization based on causal inference, where all propensity scores are initially as 1 for both row- and column-wise propensity scores. 2-D Unbiased Ranker with Initialization directly uses the propensity estimation results based on the strict natural experiments, where only special cases are included. 2-D Unbiased Ranker with Calibrated Initialization refers to the proposed method that adopts calibration for parameter estimation, where more data are included in the natural experiment. For all offline experiments, we adopt *10*-fold cross-validation where all data are randomly split into ten folds and in each round one fold is used for testing and the rest for training. All reported results are the average of ten folds.

4.3 Experimental Results

Table 5 and Table 6 illustrate the performance for different methods on NDCG@3, NDCG@5 and AUC for logged-in and logged-out users, respectively. We report all results on training (Train), validation (Vali) and test (Test) datasets. By observing the experimental results, we make following observations.

The proposed method UC achieves the best NDCG@3 among all methods for looged-in data and UI achieves the best result for logged-out data. The results prove that explicitly modeling the 2-dimensional presentation bias enable us to improve the relevance between queries and items. The proposed calibration method for natural experiments seems to be less effective for logged-out users, where the margin between UI and the runner-up method UR is not significant, and the result of UC falls behind both UI and UR. Unbiased LambdaMART also performs well on logged-out users, which proves that presentation bias exist in the search log data. Note that in the experiment for logged-in users, after adopting propensity estimation results, the performance of SRI is less competitive than that of the original SVMRank algorithm. This may be caused by the high variance of propensity estimation with randomized experiments, which further highlights the importance of natural experiments, where a massive amount of search log data can be used to reduce the variance.

The NDCG@5 results extend the items for evaluation from top 3 to top 5. The proposed method UC achieves the best results on both of the logged-in user data and logged-out user data. The runner-up for logged-in data is UR, and the runner-up for logged-out data is UI, which are variants of the proposed method UC by dropping off initialization with causal inference, and the calibration of initialization. Note that the performance of unbiased methods is very similar to that of conventional learning-to-ranking algorithms. This proves that directly applying existing unbiased methods may not solve the problem of 2-dimensional product search in E-Commerce. The proposed approach also outperforms the 2-dimensional variants of existing unbiased ranking methods. Note that the 2-D variants mostly underperform comparing with the original methods. The results reveal that classic unbiased models cannot be easily extended to deal with the novel challenge.

Since label information is mostly binary for search log data, the task can also be seen as a binary classification problem [16]. Therefore, we adopt AUC to measure the effectiveness. AUC is ideal for our task since it is robust to unbalanced dataset and our data is highly skewed. AUC ranges from 50% to 100% where 50% means the results are randomly chosen and 100% indicates the theoretically best performance. The AUC results show that the proposed methods with its variants achieve the best performance on both logged-in and logged-out data. Similarly, the results of conventional ranking methods are very close or even better than the results of unbiased 1-dimensional ranking methods, which indicate that directly modeling a 1-dimensional propensity score downgrades the performance of the ranking model.

4.4 Diversity

We also study the diversity of search results with $NDiv@K$. Table 7 illustrates the $NDiv@K$ for different methods with a varying K. We use the proposed method UC that has the best performance to compare against the production model that has not considered influence of presentation bias. We vary the range from 1 to 10 to focus on the top ranked items. According to the definition of $NDiv@K$, a larger K represents that an item is more likely to converge, and the corresponding diversity increases. Based on the experimental results we make following observations:

Not only improving effectiveness, UC methods also significantly increase the diversity of true positives by favoring items that were ranked lower by the production system, which are highly biased by positions. When K becomes larger, UC consistently outperforms the production model and maintains a large margin. We observe that the diversity of the production system increases faster when we vary K. This is because of the small diversity value. A large K would make all methods converge to 1. Since top ranked results are the most important in search and ranking applications, the superiority of UC on top 10 positions proves its utility in real applications.

5 RELATED WORK

The work is closely related to presentation/position bias in information retrieval. Though presentation bias has been intensively studied in the literature of information retrieval [9, 11, 19, 22, 30], learning unbiased ranking models with biased feedback data is

Table 5: Experimental results of different ranking methods for logged-in users in terms of NDCG@3, NDCG@5 and AUC on training, validation and test sets. *10*-fold cross-validation has been adopted and a holdout set is used as a validation set. All reported results are the average calculated based on the ten folds.

	Method	NDCG@3			NDCG@5			AUC		
		Train	Vali	Test	Train	Vali	Test	Train	Vali	Test
Conventional Ranking	SR	0.3817	0.3904	0.3916**	0.4043	0.3946	0.4292**	0.8494	0.8382	0.8205**
	GBDT	0.3653	0.3560	0.3245**	0.4174	0.4212	0.4433**	0.8330	0.8457	0.8298**
	LM	0.3490	0.3368	0.3588**	0.3717	0.4500	0.4129**	0.8017	0.7876	0.7714**
Unbiased Ranking	DLA	0.3508	0.3476	0.3497**	0.4236	0.4143	0.4385**	0.8060	0.8044	0.7883**
	SRI	0.3799	0.3700	0.3845**	0.3861	0.4033	0.4016**	0.8073	0.7949	0.7799**
	TU	0.3481	0.3702	0.3630**	0.4239	0.4084	0.4251*	0.8076	0.7847	0.7885**
	UL	0.3787	0.3447	0.3744**	0.4005	0.3839	0.3635**	0.7953	0.7768	0.7835**
2-D Unbiased Ranking	DLA^2	0.3500	0.3494	0.3399**	0.3825	0.4162	0.4087**	0.7749	0.8045	0.7725**
	TU^2	0.3143	0.3773	0.3560**	0.4007	0.3944	0.3979**	0.8237	0.7723	0.7122**
	UL^2	0.3649	0.3443	0.3384**	0.4064	0.3533	0.3457**	0.7826	0.7173	0.7824**
Proposed Methods	UR	0.3989	0.3971	0.4099**	0.4192	0.4182	0.4451**	0.8623	0.8649	0.8530**
	UI	0.4302	0.4003	0.4171**	0.4370	0.4177	0.4249**	0.8848	0.8736	0.8660**
	UC	0.4374	0.4326	**0.4401**	0.4930	0.4604	**0.4778**	0.8971	0.8825	**0.8833**

* indicates that the method is outperformed by the best one by 0.05 statistical significance level with paired t-tests, ** indicates 0.01.

Table 6: Experimental results of different ranking methods for logged-out users in terms of NDCG@3, NDCG@5 and AUC on training, validation and test sets. *10*-fold cross-validation has been adopted and a holdout set is used as a validation set. All reported results are the average calculated based on the ten folds.

	Method	NDCG@3			NDCG@5			AUC		
		Train	Vali	Test	Train	Vali	Test	Train	Vali	Test
Conventional Ranking	SR	0.3676	0.3820	0.3548**	0.4271	0.4356	0.4495**	0.7827	0.7922	0.7909**
	GBDT	0.3390	0.3328	0.3408**	0.4598	0.4666	0.4608**	0.7865	0.7933	0.7988**
	LM	0.3275	0.3105	0.3231**	0.4368	0.4496	0.4374**	0.7762	0.7817	0.7849**
Unbiased Ranking	DLA	0.3614	0.3226	0.3367**	0.4591	0.4746	0.4405**	0.7961	0.7993	0.7752**
	SRI	0.3543	0.3178	0.3274**	0.4727	0.4338	0.4496**	0.8027	0.7806	0.7756**
	TU	0.3756	0.3506	0.3543**	0.4575	0.4558	0.4240*	0.7869	0.7793	0.7797**
	UL	0.3814	0.4090	0.3915**	0.4379	0.4439	0.4304**	0.8027	0.7850	0.7859**
2-D Unbiased Ranking	DLA^2	0.3335	0.3146	0.3218**	0.4541	0.4456	0.4222**	0.7482	0.7238	0.7065**
	TU^2	0.3387	0.3331	0.3297**	0.4292	0.4473	0.3889**	0.7225	0.7854	0.7785**
	UL^2	0.3494	0.3964	0.3765**	0.4393	0.4219	0.4190**	0.7331	0.7538	0.7787**
Proposed Methods	UR	0.4177	0.4269	0.4071*	0.4603	0.4549	0.4366**	0.8357	0.8315	0.8247**
	UI	0.4204	0.4246	**0.4091**	0.4564	0.4678	0.4699**	0.8432	0.8301	0.8318**
	UC	0.4415	0.3979	0.4021**	0.4913	0.4890	**0.4816**	0.8423	0.8273	**0.8384**

* indicates that the method is outperformed by the best one by 0.05 statistical significance level with paired t-tests, ** indicates 0.01.

Table 7: Comparison of diversity of search results in terms of $NDiv@K$ with a varying K, from top 1 to top 10. For each search session, we calculate Normalized Diversity (NDiv) for the proposed UC and the production model CONTROL.

Method	NDiv@1	NDiv@2	NDiv@3	NDiv@4	NDiv@5	NDiv@6	NDiv@7	NDiv@8	NDiv@9	NDiv@10
UC	0.37	0.40	0.42	0.44	0.45	0.45	0.46	0.48	0.49	0.49
Production	0.21**	0.24**	0.27**	0.30*	0.32**	0.33**	0.35**	0.36**	0.37**	0.39**

Symbol * indicates that the method is outperformed by the best one by 0.05 statistical significance level with paired t-tests, ** indicates 0.01.

a relatively new problem. For many emerging areas, such as E-Commerce search and social media search, it is impractical to obtain large-scale manual annotation since there task is more personalized. Another example is email search, where it is not allowed to expose large-scale email data to manual annotators to label. Therefore, user feedback signals, such as clicks and conversions (purchases, downloads, *etc.*) are commonly used for training a model.

There are different ways of dealing with presentation bias. A straightforward way is to estimate the probability of being observed. A conversion on a lower position is assigned more weights than one a higher position, since its probability of being observed is relatively small [15, 16, 24, 25, 27]. They follow Inverse Propensity Score Weighting (IPSW) [5, 12, 18, 21], which has been widely used in the areas where statistics standardized to a population are different from where the observations were collected. All methods in this stream require a randomized experiment or intervention experiment to be conducted. The main intuition of our work is to avoid using randomized experiments which are expensive for many emerging applications. There have been efforts in studying automatic debiasing without using randomized experiments. Ai *et al.* propose to model this problem as a dual learning problem [3], its convergence cannot be guaranteed since it requires both of the dual problems to converge in the early stage. Our work is thus to find a way that is theoretically unbiased and guaranteed to converge.

6 CONCLUSION

In this work, we study the problem of 2-dimensional product search. We present an end-to-end framework that jointly estimates propensity scores and learns a ranking model. In order to facilitate the iterative optimization of the nested problem, we introduce a novel way to initialize propensity scores with causal inference. In particular, we propose a natural experiment and introduce how we calibrate the estimation. We present experimental results based on real-world data.

Several interesting directions remain to be studied in the future. Since we focus on implicit user feedback signals, it would be interesting to explore the possibility of integrating manual annotation like crowdsourcing results or the attention heatmap to supervise or facilitate the inference process. It would also be interesting to study how the proposed method can be adopted to improve other applications such as personalization and recommender systems.

REFERENCES

[1] Aman Agarwal, Xuanhui Wang, Cheng Li, Michael Bendersky, and Marc Najork. 2019. Addressing Trust Bias for Unbiased Learning-to-Rank. In *The World Wide Web Conference*. ACM, 4–14.

[2] Aman Agarwal, Ivan Zaitsev, and Thorsten Joachims. 2018. Counterfactual Learning-to-Rank for Additive Metrics and Deep Models. *arXiv preprint arXiv:1805.00065* (2018).

[3] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased Learning to Rank with Unbiased Propensity Estimation. (2018).

[4] Joshua D Angrist, Guido W Imbens, and Donald B Rubin. 1996. Identification of causal effects using instrumental variables. *Journal of the American statistical Association* 91, 434 (1996), 444–455.

[5] Peter C Austin and Elizabeth A Stuart. 2015. Moving towards best practice when using inverse probability of treatment weighting (IPTW) using the propensity score to estimate causal treatment effects in observational studies. *Statistics in medicine* 34, 28 (2015), 3661–3679.

[6] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[7] Thomas D Cook, Donald Thomas Campbell, and William Shadish. 2002. *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin Boston.

[8] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*. ACM, 87–94.

[9] Georges E Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations.. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 331–338.

[10] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[11] Sreenivas Gollapudi and Rina Panigrahy. 2010. System of ranking search results based on query specific position bias. (2010). US Patent App. 12/335,396.

[12] Keisuke Hirano and Guido W Imbens. 2001. Estimation of causal effects using propensity score weighting: An application to data on right heart catheterization. *Health Services and Outcomes research methodology* 2, 3-4 (2001), 259–278.

[13] Ziniu Hu, Yang Wang, Qu Peng, and Hang Li. 2019. Unbiased LambdaMART: An Unbiased Pairwise Learning-to-Rank Algorithm. In *The World Wide Web Conference*. ACM, 2830–2836.

[14] Thorsten Joachims. 2009. Svmrank: Support vector machine for ranking. *Cornell University* (2009).

[15] Thorsten Joachims, Adith Swaminathan, and Maarten de Rijke. 2018. Deep learning with logged bandit feedback. (2018).

[16] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 781–789.

[17] Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. 2017. On Application of Learning to Rank for E-Commerce Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 475–484.

[18] Brian K Lee, Justin Lessler, and Elizabeth A Stuart. 2010. Improving propensity score weighting using machine learning. *Statistics in medicine* 29, 3 (2010).

[19] Kristina Lerman and Tad Hogg. 2014. Leveraging position bias to improve peer recommendation. *PloS one* 9, 6 (2014), e98914.

[20] Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.

[21] Jared K Lunceford and Marie Davidian. 2004. Stratification and weighting via the propensity score in estimation of causal treatment effects: a comparative study. *Statistics in medicine* 23, 19 (2004), 2937–2960.

[22] Jiaxin Mao, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. Constructing Click Models for Mobile Search. (2018).

[23] Harrie Oosterhuis and Maarten de Rijke. 2018. Differentiable unbiased online learning to rank. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1293–1302.

[24] Tobias Schnabel, Adith Swaminathan, Peter I Frazier, and Thorsten Joachims. 2016. Unbiased comparative evaluation of ranking functions. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*. ACM, 109–118.

[25] Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. In *Advances in Neural Information Processing Systems*. 3231–3239.

[26] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 115–124.

[27] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search. (2018).

[28] Liang Wu and Mihajlo Grbovic. 2020. How Airbnb Tells You Will Enjoy Sunset Sailing in Barcelona? Recommendation in a Two-Sided Travel Marketplace. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2387–2396.

[29] Liang Wu, Diane Hu, Liangjie Hong, and Huan Liu. 2018. Turning Clicks into Purchases: Revenue Optimization for Product Search in E-Commerce. (2018).

[30] Yisong Yue, Rajan Patel, and Hein Roehrig. 2010. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *Proceedings of the 19th international conference on World wide web*. ACM, 1011–1018.