Debiasing Grid-based Product Search in E-commerce

Ruocheng Guo*
Arizona State University
rguo12@asu.edu

Xiaoting Zhao
Etsy Inc.
xzhao@etsy.com

Adam Henderson
Etsy Inc.
ahenderson@etsy.com

Liangjie Hong*
LinkedIn Inc.
liahong@linkedin.com

Huan Liu
Arizona State University
huan.liu@asu.edu

ABSTRACT

The widespread usage of e-commerce websites in daily life and the resulting wealth of implicit feedback data form the foundation for systems that train and test e-commerce search ranking algorithms. While convenient to collect, implicit feedback data inherently suffers from various types of bias since user feedback is limited to products they are exposed to by existing search ranking algorithms and impacted by how the products are displayed. In the literature, a vast majority of existing methods have been proposed towards unbiased learning to rank for list-based web search scenarios. However, such methods cannot be directly adopted by e-commerce websites mainly for two reasons. First, in e-commerce websites, search engine results pages (SERPs) are displayed in 2-dimensional grids. The existing methods have not considered the difference in user behavior between list-based web search and grid-based product search. Second, there can be multiple types of implicit feedback (e.g., clicks and purchases) on e-commerce websites. We aim to utilize all types of implicit feedback as the supervision signals. In this work, we extend unbiased learning to rank to the world of e-commerce search via considering a grid-based product search scenario. We propose a novel framework which (1) forms the theoretical foundations to allow multiple types of implicit feedback in unbiased learning to rank and (2) incorporates the *row skipping* and *slower decay* click models to capture unique user behavior patterns in grid-based product search for inverse propensity scoring. Through extensive experiments on real-world e-commerce search log datasets across browsing devices and product taxonomies, we show that the proposed framework outperforms the state of the art unbiased learning to rank algorithms. These results also reveal important insights on how user behavior patterns vary in e-commerce SERPs across browsing devices and product taxonomies.

CCS CONCEPTS

• **Information systems** → **Learning to rank**.

*This work was conducted while the first and forth author were working at Etsy Inc.

KDD '20, August 23–27, 2020, Virtual Event, USA

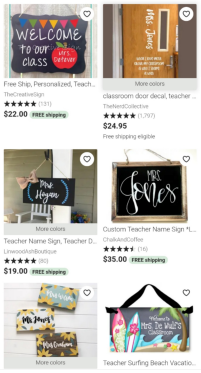
ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403336

ACM Reference Format:
Ruocheng Guo, Xiaoting Zhao, Adam Henderson, Liangjie Hong, and Huan Liu. 2020. Debiasing Grid-based Product Search in E-commerce. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining USB Stick (KDD '20), August 23–27, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403336

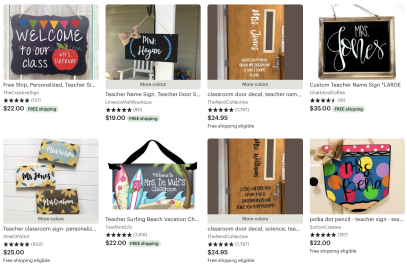
1 INTRODUCTION

Recently, large-scale search ranking systems have been deployed for a variety of e-commerce websites such as Amazon, Ebay, JD, Taobao and Walmart. Different from the traditional list-based web search engines such as Google and Baidu which display search engine result pages (SERPs) in the manner of 1-dimensional lists with textual information, e-commerce websites show SERPs in 2-dimensional grids along with images and meta information of the products. Such a difference in display can significantly change the way users interact with SERPs. In a previous study [27], researchers observed several unique user behavior patterns in grid-based SERPs with images: (1) users may scroll down to browse more products by skipping some rows in the middle of each SERP, and (2) the decay of users' attention is often slower than that in list-based web search. In addition, such grid-based display fashion can also amplify the differences in user behavior patterns due to different browsing devices. For example, in Fig 1a, we can observe that, limited by the width of screen of mobile devices, in the SERPs of e-commerce data, products are organized in two columns for mobile device users. In contrast, on desktops, SERPs display products in four columns (See Fig. 1b). This implies that on mobile devices, users need to scroll more to reach the same position of a SERP, which can also influence user behavior patterns.

Search logs with implicit feedback are widely adopted in training and testing learning to rank algorithms. Such log data of e-commerce search often consists of three main components: (1) **Features** describing a query and a potentially relevant product, (2) **SERPs** presented as 2-dimensional grids of products ranked by existing search algorithms based on the features, and (3) **Implicit Feedback** showing the users' opinions on the products shown in SERPs (e.g., clicks and purchases). Given search logs with implicit feedback, we are only able to observe the implicit feedback of users to those products shown to and examined by them in the SERPs. In other words, we cannot observe the *counterfactuals*, i.e., how feedback would have been if the SERPs, i.e., products and their positions, had been different. This inherent bias of implicit feedback data presents challenges to the development of learning to rank algorithms. Position bias is one of the most significant sources



(a) An e-commerce SERP on mobile device



(b) An e-commerce SERP on desktop

Figure 1: E-commerce SERPs on mobile devices and desktops: products are shown in two-column and four-column grids along with images and meta information.

of bias in implicit feedback search log. It describes the phenomenon that items ranked higher are more likely to be examined than others, and therefore, are more probable to observed with implicit feedback. Position bias confounds the causal effect of SERPs on users' implicit feedback. Unbiased learning to rank methods have been proposed to mitigate this bias. However, these methods focus on list-based web search.

As mentioned before, the way e-commerce websites display SERPs can lead to different user behavior patterns from those in SERPs of traditional list-based web search. In addition, the implicit feedback data from e-commerce websites often comes with more than one type of user feedback (e.g., clicks and purchases). Such observations raise the following new research questions in mitigating position bias in e-commerce search engines: (1) How can we leverage the unique patterns of user behavior in grid-based product search to mitigate the position bias? (2) How can we handle the difference of user behavior patterns caused by browsing devices and other unique factors like product taxonomies in the context of e-commerce? We are also interested in understanding how users' behaviors vary across different browsing devices and product taxonomies.

In this work, we propose a novel framework to address the unique challenges of debiasing grid-based product search for e-commerce. Our contributions can be summarized as below:

- We formulate the problem of unbiased learning to rank for grid-based product search in the context of e-commerce. In short, this problem can be distinguished from the existing unbiased learning to rank problem by two properties: (1) There can be multiple types of implicit feedback. (2) As SERPs are shown in a grid-based fashion, each position i comes along with its row number and column number.
- We propose the *joint examination hypothesis*, which extends the original examination hypothesis widely used in list-wise web search to handle multiple types of implicit feedback in the context e-commerce.
- With the joint examination hypothesis, we propose a novel unbiased learning to rank framework which has three unique properties: (1) It offers an unbiased estimate of the original loss function under mild assumptions. (2) It handles multiple types of user feedback. (3) It incorporates inverse propensity scoring models for unique user behavior patterns in grid-based product search.
- We perform extensive experiments in real-world e-commerce search log data across browsing devices and product taxonomies, and the proposed framework demonstrates significant improvement over the state-of-the-art baselines.
- We provide insights of the difference in user behavior patterns across browsing devices as well as product taxonomies through comparison studies.

We organize the rest of this paper as follows. The problem statement of unbiased learning to rank for grid-based product search is defined in Section 2. Section 3 presents the proposed framework. Experiments on real-world e-commerce search log datasets are presented in Section 5 with discussions. Section 6 reviews related work. Finally, Section 7 presents conclusions and future work.

2 PROBLEM STATEMENT

In this section, we introduce the technical preliminaries and present the problem statement. We start with an introduction of the technical preliminaries. Then we introduce the settings of unbiased learning to rank in grid-based product search.

2.1 Technical Preliminaries

Generally, boldface uppercase letters (e.g., $\boldsymbol{X}$), boldface lowercase letters (e.g., $\boldsymbol{x}$) and normal lowercase (e.g., x) letters denote matrices, vectors and scalars, respectively. Let $\boldsymbol{x}_q^i$ denote the feature vector of the query-product pair in the i-th position and $\boldsymbol{X}_q$ signify the feature matrix of all query-product pairs in the SERP of the query q. $\bar{\boldsymbol{y}}_q$ signify the vector of product indexes in the search session corresponding to the query q in the observed search log data. $\boldsymbol{o}_q$ denotes the binary vector corresponding to whether a product in q is examined. For example, $o_q^i = 1$ (0) means the product ranked in the i-th position has been examined (not examined). $\boldsymbol{c}_q$ and $\boldsymbol{p}_q$ are the vectors of clicks and purchases of the products $\bar{\boldsymbol{y}}_q$ in the SERP of the query q. $c_q^i = 0, 1$ means the i-th product is not clicked and clicked. $p_q^i = 0, 1$ means the product is not purchased and purchased, respectively. Then the training set containing n queries and their search result sessions can be denoted by $\{\boldsymbol{X}_q, \bar{\boldsymbol{y}}_q, \boldsymbol{c}_q, \boldsymbol{p}_q\}_{q=1}^n$. We define a ranker as a function $f : \mathcal{X} \rightarrow \mathbb{R}$ mapping the features of a query-product pair to a real number standing for its ranking score.

2.2 Problem Statement

In this work, we focus on the offline setting where randomized experiments are not available. In contrast to [16, 22] where randomized experiments are performed, we can neither obtain user feedback to SERPs with randomized ranking nor ground truth of propensity scores. This requires us to estimate propensity scores along with train the ranker as in [2, 14].

Definition 1. ***Unbiased Learning to Rank for Grid-based Product Search.*** *Given search log data* $\{\boldsymbol{X}_q, \bar{\boldsymbol{y}}_q, \boldsymbol{c}_q, \boldsymbol{p}_q\}_{q=1}^n$ *and the*

number of columns and rows of e-commerce SERPs, we aim to learn the propensity score model(s) which would be used to reweigh products for unbiased estimate of rankers' loss and train unbiased rankers with inverse propensity scoring to maximize e-commerce search metrics (e.g., purchase NDCG@K) on held-out test data.

3 INVERSE PROPENSITY SCORING FOR GRID-BASED PRODUCT SEARCH

In this section, we start with a brief introduction of background knowledge. Then we provide descriptions of the proposed framework including the loss function and the propensity score models.

3.1 Background

Cascade Click Models. Click models have been used to connect user behavior patterns (e.g., click rate) to the evaluation metrics of learning to rank algorithms [18]. The cascade model [7] is one of the most widely adopted click models which can quantify the probabilities of multiple types of users' behaviors (e.g., click, stop and examination) in list-based web search SERPs. In particular, let α describes the how likely users continue to browse the next product, then the probability that users stop and leave the search results page at position i can be formulated as $\beta(i) = (1-\alpha)\prod_{j=0}^{i-1}\alpha$. In a series of randomized controlled trial [7], the cascade click model has been shown to outperform others in click prediction tasks.

Propensity Score Estimation from Observational Data. Generally, unbiased estimation of propensity scores requires randomized experiments [16, 23]. However, randomized experiments can be expensive, time consuming and can hurt users' experience. In [2, 14, 23], Expectation Maximization (EM) style optimization algorithms have been proposed to learn propensity models without randomized experiments. These methods are based on the intuition that the joint optimum of the ranker and the propensity model leads to unbiased estimates of propensity scores. But these algorithms can be trapped in local joint optimum. Based on the same intuition, in our proposed framework, we aim to find the joint optimum of the two models by minimizing the loss function through grid search on hyperparameters.

3.2 Pairwise Unbiased Learning to Rank for Multiple Types of Feedback

3.2.1 Joint Examination Hypothesis. The *examination hypothesis* is a widely adopted assumption in the literature of unbiased learning to rank [2, 14, 16], which postulates that a user clicks a document iff the document is examined and relevant. Only considering click and the attractiveness of products (similar to relevance of documents), we can rewrite the straightforward counterpart of the original examination hypothesis in the context of e-commerce as:

$$P(c_q^i = 1|\boldsymbol{x}_q^i) = P(o_q^i = 1|\boldsymbol{x}_q^i)P(a_q^i = 1|\boldsymbol{x}_q^i), \tag{1}$$

where a_q^i is the binary variable representing attractiveness of the product at position i of the search results page of query q. We define attractiveness of a product as how attractive it appears in SERPs.

However, in the context of e-commerce search, we need to adapt this hypothesis such that we can take multiple types of user feedback into consideration. For simplicity, in this work, we only consider two types of feedback: clicks and purchases. Nevertheless, the proposed hypothesis as well as the other components of the proposed framework can be extended to account for more types of feedback (e.g., favorite and add-to-cart). To consider both clicks and purchases, we propose the *joint examination hypothesis*, a novel extension of the examination hypothesis, which is defined as:

Joint Examination Hypothesis. No matter if a user eventually does purchase or not purchase a product, she clicks a product iff the product is examined and attractive. The joint examination hypothesis can be formulated as:

$$P(p_q^i, c_q^i = 1|\boldsymbol{x}_q^i) = P(o_q^i = 1|\boldsymbol{x}_q^i)P(p_q^i, a_q^i = 1|\boldsymbol{x}_q^i) \tag{2}$$

In short, the joint examination hypothesis extends the examination hypothesis to the context of e-commerce where multiple types of feedback exist. We are aware of that the joint examination hypothesis is a stronger assumption than the original examination hypothesis as we can recover the original examination hypothesis (Eq. 1) by marginalizing the joint examination hypothesis (Eq. 2) over $P(p_q^i)$. Note that this assumption can be relaxed when noisy clicks are taken into consideration, which is similar to that in [16]. We do *not* model purchase as a function of attractiveness because we define attractiveness of a product as how attractive it appears in SERPs for a user to start engaging (i.e, click). It is natural to consider users' shopping journey as a two-stage process illustrated in [24], where at first users search for a query and decide to click on a product displayed by SERPs when found it attractive. Then, the user makes purchase decision after examining the detail catalog on the product landing page.

Less Clicks for Less Attractive Products. We add a mild assumption $P(a_q^i = 0|\boldsymbol{x}_q^i) = \zeta P(c_q^i = 0|\boldsymbol{x}_q^i)$ where we let $\zeta \in (0, 1]$ such that the assumption is coherent with Eq. 1. Intuitively, this means a less attractive product would receive less clicks.

3.2.2 The Loss Function. Let $\mathcal{I}_q$, $\mathcal{I}_q'$ and $\mathcal{I}_q''$ denote three types of pairs: *(click, no feedback), (purchase, no feedback) and (purchase, click)*, respectively. Then, the loss function of mis-ranking (as well as the gradients) can be reduced to an aggregation of losses defined over these three types of pairs. Note that the main task of e-commerce search engines is to maximize purchase or revenue of the website. But users would *unlikely* be able to make purchase decisions based on product images (and limited information) displayed on SERPs, instead, the product images shown on SERPs need to first attract them to click on products first, which then lead them to the product landing pages and help them to inform purchase decision after examining the product details. Therefore, in SERPs, we also aim to maximize the attractiveness of products shown in top positions such that purchase decisions can be triggered later after clicking. Based on this intuition, we first formulate the loss function based on purchases and attractiveness by adopting the fashion of pairwise ranking algorithms and then propose an unbiased estimate of it using implicit feedback data as:

$$\begin{aligned}\mathcal{L} = &\int \mathbb{1}(p_q^i = 0)\ L\ dP(\boldsymbol{x}_i, a_q^i = 1, \boldsymbol{x}_j, a_q^j = 0)\\ &+ A\int \mathbb{1}(p_q^i = 1)\ L\ dP(\boldsymbol{x}_i, a_q^i = 1, \boldsymbol{x}_j, a_q^j = 0)\\ &+ B\int L'\ dP(\boldsymbol{x}_i, p_q^i = 1, a_q^i = 1, \boldsymbol{x}_j, p_q^j = 0, a_q^j = 1),\end{aligned} \tag{3}$$

where the function $L = L(\boldsymbol{x}_i, a_q^i, \boldsymbol{x}_j, a_q^j)$ denotes the pairwise loss penalizing mis-ranking of *(click, no feedback)* or *(purchase, no feedback)* pairs. Similarly, the function $L' = L'(\boldsymbol{x}_i, a_q^i, p_q^i, \boldsymbol{x}_j, a_q^j, p_q^j)$ signifies the penalty for mis-ranking on *(purchase, click)* pairs. Note that the parameterization of the functions L and L' can be flexible. The details of how the loss functions L and L' are defined and optimized can be found in Section 4. Note that under Assumption Eq. 2, both click and purchase imply attractiveness. $\mathbb{1}(\cdot)$ is the indicator function. The hyperparameters $A, B \geq 0$ control the trade-off of penalizing the mis-ranking (purchase, no feedback) and (purchase, click) with respect to the pairs on (click, no feedback). Therefore, the loss of (purchase, no feedback) and (purchase, click) pairs are multiplied with A and B, respectively.

3.2.3 Unbiased Estimate of the Loss Function. However, we are not capable to evaluate this loss function (Eq. 3) with implicit feedback data because the ground truth of attractiveness cannot be observed. Alternatively, we aim to infer the attractiveness through the observed user feedback including clicks and purchases. This can be done by replacing the loss functions and probabilities relevant to attractiveness with the counterparts of user feedback with the following assumptions:

$$L(\boldsymbol{x}_q^i, \mathrm{a}_\mathrm{q}^\mathrm{i}, \boldsymbol{x}_q^j, \mathrm{a}_\mathrm{q}^\mathrm{j}) = L(\boldsymbol{x}_q^i, \mathrm{c}_\mathrm{q}^\mathrm{i}, \boldsymbol{x}_q^j, \mathrm{c}_\mathrm{q}^\mathrm{j}) \quad (4)$$

$$L'(\boldsymbol{x}_q^i, \mathrm{a}_\mathrm{q}^\mathrm{i}, p_q^i, \boldsymbol{x}_q^j, \mathrm{a}_\mathrm{q}^\mathrm{j}, p_q^i) = L'(\boldsymbol{x}_q^i, \mathrm{c}_\mathrm{q}^\mathrm{i}, p_q^i, \boldsymbol{x}_q^j, \mathrm{c}_\mathrm{q}^\mathrm{j}, p_q^j) \quad (5)$$

$$L(\boldsymbol{x}_q^i, \mathrm{c}_\mathrm{q}^\mathrm{i}, \boldsymbol{x}_q^j, \mathrm{c}_\mathrm{q}^\mathrm{j}) \neq 0 \;\; iff \;\; \mathrm{c}_\mathrm{q}^\mathrm{i} \neq \mathrm{c}_\mathrm{q}^\mathrm{j}. \quad (6)$$

$$L'(\boldsymbol{x}_q^i, \mathrm{c}_\mathrm{q}^\mathrm{i}, p_q^i, \boldsymbol{x}_q^j, \mathrm{c}_\mathrm{q}^\mathrm{j}, p_q^j) \neq 0 \;\; iff \;\; (c_q^i = c_q^j = 1) \cap (p_q^i \neq p_q^j). \quad (7)$$

We propose a loss function $\mathcal{L}_{imp}$ that can be evaluated on implicit feedback data. The subscript *imp* means implicit feedback. With inverse propensity scoring, we show below in Theorem 3.1 that the new loss function $\mathcal{L}_{imp}$ is an unbiased estimate of the original loss. In particular, the proposed loss can be formulated as:

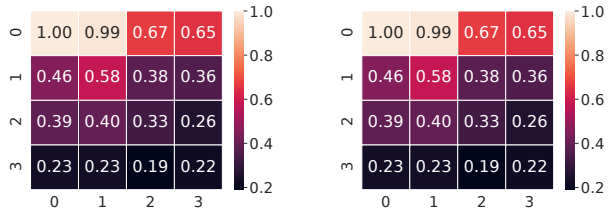
$$\begin{aligned}\mathcal{L}_{imp} = &\int \mathbb{1}(p_q^i = 0)\, L\, \frac{dP(\boldsymbol{x}_q^i, \boldsymbol{c}_q^i = 1, \boldsymbol{x}_q^j, \boldsymbol{c}_q^j = 0)}{P(o_q^i = 1|\boldsymbol{x}_q^i)} \\ &+ A' \int \mathbb{1}(p_q^i = 1)\, L\, \frac{dP(\boldsymbol{x}_q^i, \boldsymbol{c}_q^i = 1, \boldsymbol{x}_q^j, \boldsymbol{c}_q^j = 0)}{P(o_q^i = 1|\boldsymbol{x}_q^i)} \\ &+ B' \int L' \frac{dP(\boldsymbol{x}_q^i, \boldsymbol{c}_q^i = 1, \boldsymbol{x}_q^j, \boldsymbol{c}_q^j = 1)}{P(o_q^i = 1|\boldsymbol{x}_q^i) P(o_q^j = 1|\boldsymbol{x}_q^j)},\end{aligned} \quad (8)$$

where $A' = \zeta A$ and $B' = B$.

Theorem 3.1. *With the assumptions in Eq. 2 and Eq. 4-7, $\mathcal{L}_{imp}$ is an unbiased estimate of the original loss function $\mathcal{L}$.*

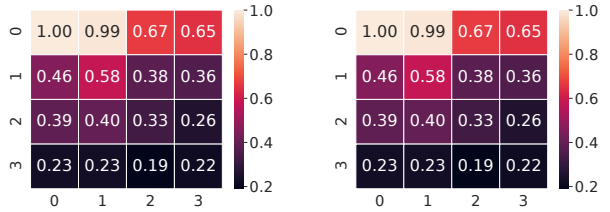
The proof of Theorem 3.1 can be found in Appendix[1]. With Theorem 3.1, we now know that the proposed loss function (Eq. 8) provides unbiased estimate of the original loss function (Eq. 3) given biased implicit feedback data. Following existing work [16, 23], we simplify the problem with the following assumption: The probability of examination only depends on the position, which can be formulated as $P(o_q^i = 1|\boldsymbol{x}_q^i) = P(o^i)$. As the focus of this work is to handle multiple types of user feedback and incorporate the unique user behavior patterns in grid-based product search for unbiased learning to rank, we leave modeling position bias with richer information (e.g., query-product features) as future work.

[1]The Appendix can be found at https://www.public.asu.edu/~rguo12/kdd20_appx.pdf.



(a) NCTR on mobile devices (b) NCTR on desktops

Figure 2: Normalized click through rate (NCTR) in the top 16 positions of the H&L dataset.



(a) NPR on mobile devices (b) NPR on desktops

Figure 3: Normalized purchase rate (NPR) in the top 16 positions of the H&L dataset.

3.3 Propensity Score Models for Grid-based Product Search

Here, we motivate the usage of two click models as propensity models through data analysis results which verify that they can capture unique user behavior patterns in grid-based product search. Then descriptions of the two propensity models are given below.

In the literature [27], variants of the cascade click model [7] have been proposed to capture the unique patterns of users' behaviors in grid-based search. These models can provide consistent probabilities of users' behaviors (e.g., examination, continuing to browse the next product and skipping a row) in such context. In eye tracking experiments of [27], three unique phenomena have been observed in grid-based search with images: row skipping, slower decay and middle bias. In this work, we propose to utilize the click models capturing the *row skipping* and *slower decay* phenomena as propensity score models for unbiased learning to rank. We also provide reasons why middle bias is not considered in this work through data analysis below.

To motivate the usage of the two propensity models, we show a series of data analysis results on real-world e-commerce search log data here while the detail description of data and experiment are further explained in Section 5. Limited by space, we only show the results on the *Home and Living* datasets, similar observations are also made on the *Paper and Party Supplies* datasets (see Section 5.1 for dataset description). In Fig. 2-3, we show the normalized click through rate (NCTR) and purchase rate (NPR) of the top 16 positions for data collected from both mobile devices and desktops. These NCTR and NPR are the click through rate and purchase rate of each

position divided by those of the first position. Different from the previous work [27] which focused on the development of novel evaluation metric for learning to rank, our target is to capture users' behavior patterns in grid-based product search for more accurate and interpretable propensity score modeling.

The *middle bias* model [27] is not considered in this work for two reasons: (1). The number of columns in the SERPs of our data is small. Specifically, the SERPs show products in 2 columns for mobile devices and 4 columns for desktops. (2). Further evident in our empirical analysis (Fig. 2-3), we also do not observe the middle bias phenomenon. In particular, the NCTR and NPR of products in the middle for desktops (4-column display) are not significantly higher than those of the other products.

Row Skipping. In our datasets, similar to [27], we observe the row skipping phenomena where users can skip some rows before they click, purchase or leave SERPs. As shown in Fig. 2-3, we can see that the click through rate and purchase rate are not monotonically decreasing from top to the bottom. For example, the last position of Fig. 2a has higher NCTR than the forth last position. Based on this observation, let $r(i)$ be the row number of the i-th product. We use the *row skipping* cascade model as a propensity score model to quantify $P(o^i)$:

$$P(o^i = 1) = \Big\{ \prod_{k=0}^{r(i)-1} (1-\gamma) \prod_{j=S(k)}^{S(k)+N(k)-1} \alpha + \prod_{k=0}^{r(i)-1} \gamma \Big\} \prod_{j=S(r(i))}^{i-1} \alpha$$

γ models the trend to skip a row. $S(k)$ and $N(k)$ are the number of items before and in the k-th row. Intuitively, in the row skipping cascade model, if a user reached position i, she must have gone through the k-th row before the row of position i ($k < r(i)$). There are two possible situations: she either skipped the k-th row with the row skipping probability γ or decided to continue browsing on every single position on that row with probability $\prod_{j=S(k)}^{S(k)+N(k)-1} \alpha$.

Slower Decay. Similar to what has been discovered by previous study [27], in grid-based product search, the decay of users' attention from top to bottom in each SERP is slower than that in list-based web search. In Fig. 2a and 2b, we can observe that the NCTRs on mobile devices and desktop take 10 positions to drop to 43% and 46% of the NCTR of the first positions, which is much slower than the drop of attention in list-based web search shown in Fig. 3 of [27]. We can specify the probability of examination at position i as:

$$P(o^i = 1) = \prod_{j=0}^{i-1} \min(\beta^{row(j)} \alpha, 1.0), \tag{9}$$

where $\beta \geq 1$ models the increased patience of users in grid-based product search compared to that in the original cascade model. When $\beta = 1.0$, $P(o^i = 1)$ of the slower decay model is the same as that in the cascade model.

Besides the these models, we encourage practitioners to design models of $P(o^i)$ based on a combination of domain knowledge and propensity scores estimated from online experiments.

4 OPTIMIZATION

Without randomized experiments, we aim to achieve a joint optimum of both the propensity score models and the ranker with the implicit feedback data. Due to the simplicity of the propensity models, we consider parameters of the propensity models (α, γ, and β) as hyperparameters and adopt grid search along with minimizing the loss function $\mathcal{L}_{imp}$ to reach the joint optimum. Different from the existing ones [2, 14, 16, 23], the proposed propensity model leverages the user behavior patterns in grid-based product search.

Given propensity scores ($P(o^i)$) computed from either the row skipping or the slower decay model based on hyperparameters α, γ and β, we aim to learn a ranker f based on the unbiased loss function $\mathcal{L}_{imp}$. In particular, we adopt LambdaMART [25] where the ranker is the gradient boosting trees (GBDT) or MART [9]. In LambdaMART, instead of using a explicit loss function, we directly define the gradients of an implicit loss function, which are known as lambda gradients [5]. Toward unbiased learning to rank, similar to [14], we directly apply inverse propensity scoring to the lambda gradients. In addition, in e-commerce search, we need to consider multiple types of user feedback. We also assign different weights, A' and B', to the gradient components corresponding to trade-off in mis-ranking loss among three types of pairs. Therefore, we propose an extension of the original lambda gradient [5]. In particular, the lambda gradient of the k-th product (λ_k) can be written as:

$$\frac{\partial \mathcal{L}_{imp}}{\partial f(\boldsymbol{x}_k)} = \lambda_k = \sum_q \left(\sum_{\overline{y}_q^i = k \cap (i,j) \in \mathcal{I}_q} \frac{\lambda_{ij}}{P(o^i)} - \sum_{\overline{y}_q^i = k \cap (j,i) \in \mathcal{I}_q} \frac{\lambda_{ij}}{P(o^j)} \right)$$
$$+A' \sum_q \left(\sum_{\overline{y}_q^i = k \cap (i,j) \in \mathcal{I}'_q} \frac{\lambda_{ij}}{P(o^i)} - \sum_{\overline{y}_q^i = k \cap (j,i) \in \mathcal{I}'_q} \frac{\lambda_{ij}}{P(o^j)} \right)$$
$$+B' \sum_q \left(\sum_{\overline{y}_q^i = k \cap (i,j) \in \mathcal{I}''_q} \frac{\lambda_{ij}}{P(o^i)P(o^j)} - \sum_{\overline{y}_q^i = k \cap (j,i) \in \mathcal{I}''_q} \frac{\lambda_{ij}}{P(o^i)P(o^j)} \right),$$

where $\overline{y}_q^i = k$ means product k is at the i-th position in the SERP of query q. In addition, λ_{ij} is defined as:

$$\lambda_{ij} = \frac{-2}{1 + \exp(2(f(\boldsymbol{x}_q^i) - f(\boldsymbol{x}_q^j)))} |\Delta_{ij}|, \tag{10}$$

where $|\Delta_{ij}|$ denotes the absolute value of difference in a predefined metric (e.g., NDCG@K) if the ranking of item i and j are swapped. Note that product price is not directly involved in the lambda gradient to prevent bias towards expensive products.

5 EXPERIMENT

In this section, we start with data description followed by experimental settings. Then, through extensive experimental results, we aim to answer the following key research questions: (1). How effective is the proposed framework compared to the baselines in the task of reranking products in grid-based search? (2). How does user behavior patterns in grid-based product search vary across browsing devices and product taxonomies?

5.1 Dataset Description

Here, we provide a brief description of the product search log data used in experimental studies of this work. In addition to an introduction and a summary of statistics of the dataset, we also include details of the feature engineering procedure. Datasets are collected

Table 1: Data Statistics

Dataset	Sessions	Products	Clicks	Purchases	Features
Desktop PPS	15,360	734,289	19,241	1,913	213
Mobile PPS	12,777	611,304	14,861	1,446	215
Desktop H&L	24,905	1,184,454	29,446	2,436	195
Mobile H&L	24,208	1,148,804	26,851	2,287	195

Table 2: Feature Description

Feature Category	Examples
Product	Average historical rates of the product in last x days Price of the product Average processing and shipping time after purchases
Shop	Average rating of the products from the shop Decile of the shop's sale Top categories sold in the shop
Query	Average price of clicked products from the query Logarithm of purchase count from the query over x days Top buyer taxonomy purchased for the query
Interaction	BM25 of product's listing title and tags with query Ratio of a product's contribution to a shop's sale Difference in query average purchase price and product price

from the e-commerce website at Etsy, which is an international online marketplace for small businesses selling vintages, hand-crafted products and supplies. In particular, we pick two of the most popular product taxonomies: *Paper and Party Supplies* (PPS) and *Home and Living* (H&L). For understanding the difference in users' behaviors when they are browsing the search sessions via different devices, search logs from both desktop and mobile devices are collected. Therefore, we obtain 4 datasets: Desktop PPS, Mobile PPS, Desktop H&L and Mobile H&L. For each dataset, we only include the search result sessions with at least a click of those queries which triggered at least a click or a purchase. The statistics for these four datasets are then shown in Table 1. The number of features per dataset may vary because we remove the columns with incomplete values due to missing information in the data. For example, some of new sellers may not be familiar with the platform and therefore might forget to provide tags for some products.

Feature Engineering. The search log datasets are preprocessed to fit the format of $(\boldsymbol{X}_q, \overline{\boldsymbol{y}}_q, \boldsymbol{c}_q, \boldsymbol{p}_q)$ using the feature engineering tool Buzzsaw [20]. We summarize the features into the following four categories based on which subject they are related to: product, shop, query and interaction. In terms of how the features are computed, similar to [13], we consider features including raw features (e.g., content similarity matching between query and product, product or shop attributes such as price, title, materials, shipping time), ratio statistics (e.g., domestic sales ratio, the ratio of a product's contribution to a shop's sale), mean values over time windows (e.g., average CTR, purchase rate of the product or shop in search results in last x days) and composition features (e.g., the difference between product price and average clicked price for the query). Further descriptions of example features can be found in Table 2.

5.2 Experimental Settings

Here, we report the experimental settings. For unbiased learning to rank algorithms, in the offline settings, the most commonly adopted way of evaluation is via the task of reranking the products in SERPs of a hold-out test set [2, 14]. We randomly split the search sessions of each dataset into training (70%), validation (10%) and test sets (20%). We set $A = B = 50$ in accordance to the approximated ratio between clicks and purchases in our data after a global smoothing. We perform grid search to find optimal hyperparameter settings for the propensity score models. We search α in $\{0.8, 0.825, ..., 0.975\}$, β in $\{1.05, 1.1, 1.15, 1.2\}$ and γ in $\{0.8, 0.825, ..., 0.975\}$ to keep $P(o^i = 1)$ in a reasonable range. Algorithms that can achieve a global optimal w.r.t. parameters of both the ranker and the propensity model can also be used to obtain values of α, β and γ. For the LambdaMART ranker of the proposed framework, we search the number of leaves in $\{31, 127, 511\}$ for each tree and the number of trees in $\{100, 200, ..., 1,000\}$. Other parameters are adopted from the default setting of unbiased LambdaMART, while similar settings are also used for the baselines.

Baselines. We consider 6 baseline methods, including *four* classic learning to rank algorithms and *two* state-of-the-art unbiased learning to rank algorithms that can work without propensity scores estimated by randomized experiments. Because the implementation of Regression EM [23] is not available and empirical results in [14] also show that Unbiased LambdaMART outperform Regression EM, it is valid to skip Regression EM as a baseline in this work. Similar to the proposed model, we also enable every single baseline to handle multiple types of user feedback, by aggregating the loss function across different types with importance weights, i.e., 50:1 ratio between purchases and clicks. By doing so when compare performance, we can eliminate the influence of utilizing multiple types of feedback and safely claim the differences are caused by (1) the proposed propensity score estimation models and (2) the underlying learning to rank models. The baselines are:

MART [25] is a gradient boosting algorithm leveraging multiple additive regression trees as weak learners. It minimizes pairwise loss functions (e.g., cross entropy loss).

RankBoost [8] is a pairwise algorithm based on AdaBoost, which minimizes cross entropy loss.

LambdaMART [25] is an extension of MART which reweights each pair to optimize listwise ranking measures (e.g., NDCG@K).

Random Forests [4] is a variant of the classic machine learning algorithm which minimizes cross entropy loss.

Unbiased LambdaMART [14] is a variant of LambdaMART where each pair is reweighted by the product of their inverse propensity scores. Two propensity scores are estimated for each position along with the ranker: one for products that are clicked and purchased, and the other one for products with no feedback.

Dual Learning [2] performs joint optimization of two models. The first model is a neural network trained to maximize a listwise ranking measure. The second model is a neural network learned to optimize the likelihood of examinations on clicked products.

Evaluation Metrics. We then describe the evaluation metrics. In this work, we perform experiments in the offline setting. In particular, we evaluate the proposed framework and the baselines on the organic search logs obtained from the e-commerce website on Etsy. In organic search (non-sponsored search), the target is to maximize purchase and revenue of e-commerce websites, therefore, we adopt the three widely used metrics purchase NDCG@K, revenue NDCG@K and purchase mean average precision (MAP) [24]:

$$NDCG_{pur}@K = \frac{1}{IDCG_{pur}@K} \sum_{i=1}^{K} \frac{2^{p_q^i} - 1}{\log_2(i+1)}$$

$$NDCG_{rev}@K = \frac{1}{IDCG_{rev}@K} \sum_{i=1}^{K} \left(\frac{2^{p_q^i} - 1}{\log_2(i+1)} price_q^i \right),$$

$$MAP_{pur}@K = \frac{1}{K} \sum_{i=1}^{K} |\{j | p_q^j = 1, j = 1, ..., i\}|/i,$$

where $IDCG_{pur}@K$ and $IDCG_{rev}@K$ are the normalizers. Revenue NDCG@K is a variant of purchase NDCG@K by weighting the gain of each product with price. To consider slow decay of user attention, we set $K = 1, 2, 5, 10, 20$ for the NDCGs and $K = 20$ for MAP.

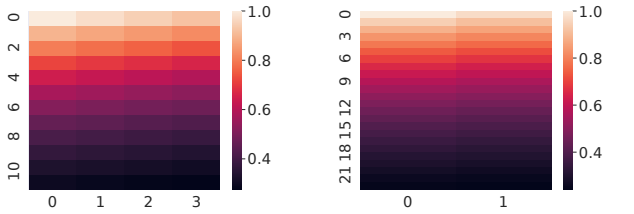
In the offline setting, we are not able to perform randomized experiments to obtain ground truth of propensity scores. Therefore, unlike the previous work relying on simulated propensity scores and relevance labels [2, 14], we could not obtain the attractiveness of products that received no feedback. To the best of our ability, we apply these evaluation metrics on hold-out test sets of search logs. This may not be the theoretically optimal strategy and we understand that there can exist attractive products which comes without user feedback. However, because of the unavailability of ground truth of attractiveness of products, we leave handling the attractive products without user feedback as a future work.

5.3 Experimental Results

Effectiveness. Here, we report the experimental results to show (1) how effective the proposed framework is in terms of improving e-commerce search results and (2) how users behavior patterns vary across different browsing devices and taxonomies. We show the results in Table 3 and make the following observations:

- At least one of the two proposed methods outperforms the baselines in almost all of the cases. This demonstrates the effectiveness of our proposed unbiased ranker, which is able to capture unique user behavior patterns in grid-based product search with these two simple propensity models.
- The proposed framework shows superior performance to unbiased LambdaMART. This corroborates the efficacy of the proposed propensity score models. This is because unbiased LambdaMART relies on a different pairwise inverse propensity scoring strategy but shares the same underlying ranker (LambdaMART). This observation can be attributed to incorporating prior knowledge of users' behavior patterns to guide the learning process of propensity score models.
- Row skipping performs better in the H&L datasets, this can be caused by the fact that users have more specific intent when they browse SERPs in this taxonomy, which means they would more likely to skip rows of products that do not look attractive. In addition, the price of products in this taxonomy has larger variance, users may skip those rows showing expensive products.
- On the mobile datasets, the performance of the best baseline, i.e. unbiased LambdaMART, is closer to the proposed framework than that on desktop datasets. This is because that the list-based web search is a better proxy for mobile devices with products displayed in 2 columns as comparing to those on desktops (4 columns).

We train separate models for different taxonomies to show that modeling different user behavior patterns across product taxonomies can be beneficial. In practical deployments, a single ranker is often trained and tested across all taxonomies. The model with highest purchases or revenue across taxonomies in randomized online experiments may be preferred in such a case.



(a) desktop H&L (b) mobile H&L

Figure 4: Propensity scores obtained through grid search that achieve the optimal performance.

Propensities. Next, we report the values of propensity scores and the hyperparameters α, β and γ that achieve the optimal performance for the proposed framework. For Desktop and Mobile PPS, the slower decay models with $\alpha = 0.95, \beta = 1.1$ and $\alpha = 0.925, \beta = 1.15$ outperform others. For Desktop and Mobile H&L, the row skipping model with $\alpha = 0.95, \gamma = 0.975$ and the slower decay model with $\alpha = 0.925, \beta = 1.1$. We show propensity scores estimated for the H&L datasets in Fig. 4 to draw connection with the earlier empirical results (Fig. 2-3). Although we cannot perfectly reconstruct the non-monotonically decreasing patterns in Fig. 2-3, in Fig. 4a, we can observe that positions at the left bottom can have higher estimated $P(o^i)$ than some positions. We can regard these propensity scores as upper bounds of the NCTR values observed in Fig. 2. This is because the NCTR values result from a combination of position bias (propensity scores) and effectiveness of the ranking algorithm(s) that generated the search logs. It can also be observed that users are more patient when they browse with desktops.

6 RELATED WORK

Here, we review the related work from the three subareas: unbiased learning to rank, grid-based search and e-commerce search.

Unbiased Learning to Rank is an area where causal inference [12] helps learning to rank. Given the same attractiveness (relevance), the probability of products (documents) being clicked may change significantly with many factors in SERPs of product (web) search. Position is one of the most significant factor. It has been studied in list-wise web search [2, 14, 16, 22, 23]. As the literature of unbiased learning to focuses on solving the problem of position bias in traditional information retrieval systems, here, we use the terms, document and relevance, instead of product and attractiveness. Joachims et al. [16] analyzed the inherent position bias in search log data with implicit feedback and proposed the Propensity SVM-Rank [15] algorithm which applies inverse propensity scoring to each clicked document to mitigate the position bias. In particular, the propensity scores of each position is estimated through an randomized experiment which randomly picks and swaps items at the i-th and j-th positions [16]. In [1], the authors extended the Propensity SVM-Rank model to directly optimize additive information retrieval metrics such as DCG and proposed to replace the SVM-Rank model with neural networks. However, such randomized experiments may degrade users' experience and would likely be

Table 3: Experimental results show comparison of model effectiveness using the held-out test set of the 4 datasets. Best results are highlighted in boldface. Significant improvements with respect to the best baseline are indicated with +.

Models	$NDCG_{pur}$					$NDCG_{rev}$					MAP_{pur}
	@1	@2	@5	@10	@20	@1	@2	@5	@10	@20	@20
Desktop PPS (Paper and Party Supplies)											
MART	0.082	0.121	0.181	0.232	0.291	0.078	0.126	0.184	0.234	0.289	0.079
RankBoost	0.087	0.117	0.184	0.243	0.303	0.087	0.110	0.182	0.241	0.303	0.084
LambdaMART	0.101	0.128	0.194	0.249	0.305	0.100	0.130	0.194	0.248	0.308	0.097
Random Forest	0.096	0.128	0.192	0.239	0.295	0.088	0.117	0.185	0.233	0.287	0.096
Unbiased LambdaMART	0.109	0.142	0.201	0.251	0.308	0.109	0.142	0.201	0.250	0.307	0.106
Dual Learning	0.098	0.136	0.211	0.277	0.327	0.098	0.136	0.211	0.277	0.327	0.094
Row Skipping	0.111	0.141	0.196	0.256	0.312	0.110	0.141	0.196	0.256	0.312	0.106
Slower Decay	**0.144**$^+$	**0.173**$^+$	**0.232**$^+$	**0.281**$^+$	**0.340**$^+$	**0.143**$^+$	**0.173**$^+$	**0.232**$^+$	**0.281**$^+$	**0.339**$^+$	**0.139**$^+$
Mobile PPS (Paper and Party Supplies)											
MART	0.154	0.197	0.236	0.294	0.347	0.148	0.184	0.227	0.289	0.343	0.154
RankBoost	0.067	0.116	0.181	0.232	0.286	0.085	0.135	0.201	0.252	0.300	0.067
LambdaMART	0.111	0.148	0.216	0.262	0.322	0.119	0.159	0.225	0.272	0.335	0.111
Random Forest	0.138	0.177	0.232	0.286	0.339	0.131	0.176	0.244	0.298	0.343	0.136
Unbiased LambdaMART	0.151	0.192	0.254	0.293	0.345	0.150	0.192	0.253	0.292	0.344	0.149
Dual Learning	0.102	0.144	0.235	0.291	0.340	0.100	0.143	0.235	0.290	0.339	0.101
Row Skipping	0.148	0.182	0.243	0.298	0.351	0.164$^+$	0.203$^+$	0.265$^+$	0.318$^+$	0.370$^+$	0.155
Slower Decay	**0.166**$^+$	**0.208**$^+$	**0.281**$^+$	**0.321**$^+$	**0.371**$^+$	**0.176**$^+$	**0.223**$^+$	**0.293**$^+$	**0.332**$^+$	**0.383**$^+$	**0.165**$^+$
Desktop H&L (Home and Living)											
MART	0.114	0.152	0.212	0.265	0.318	0.119	0.151	0.215	0.265	0.319	0.116
RankBoost	0.085	0.127	0.193	0.240	0.297	0.096	0.131	0.194	0.239	0.297	0.070
LambdaMART	0.107	0.135	0.210	0.266	0.323	0.109	0.138	0.213	0.263	0.321	0.101
Random Forest	0.109	0.164	0.228	0.275	0.325	0.102	0.145	0.212	0.260	0.307	0.112
Unbiased LambdaMART	0.148	0.184	0.243	0.284	0.340	0.148	0.185	0.243	0.284	0.340	0.145
Dual Learning	0.097	0.138	0.211	0.266	0.322	0.097	0.138	0.211	0.266	0.322	0.096
Row Skipping	**0.165**$^+$	**0.199**$^+$	**0.252**$^+$	**0.300**$^+$	**0.354**$^+$	**0.165**$^+$	**0.200**$^+$	**0.252**$^+$	**0.301**$^+$	**0.354**$^+$	**0.163**$^+$
Slower Decay	0.141	0.182	0.242	0.290	0.347	0.139	0.182	0.242	0.290	0.346	0.135
Mobile H&L (Home and Living)											
MART	0.147	0.200	0.261	0.306	0.350	0.159	0.216	0.274	0.322	0.369	0.147
RankBoost	0.084	0.117	0.169	0.227	0.291	0.094	0.131	0.181	0.234	0.295	0.083
LambdaMART	0.119	0.155	0.23	0.281	0.322	0.124	0.161	0.239	0.29	0.33	0.117
Random Forest	0.125	0.187	0.250	0.296	0.341	0.137	0.194	0.263	0.307	0.354	0.123
Unbiased LambdaMART	**0.181**	**0.237**	0.285	0.322	0.367	**0.181**	**0.237**	0.285	0.322	0.367	**0.180**
Dual Learning	0.116	0.154	0.221	0.288	0.331	0.116	0.154	0.221	0.288	0.331	0.114
Row Skipping	0.172	0.222	**0.287**	0.324	0.372	0.172	0.222	**0.287**	0.324	0.371	0.173
Slower Decay	**0.181**	0.233	0.282	**0.329**	**0.377**$^+$	**0.181**	0.233	0.282	**0.329**	**0.377**$^+$	**0.180**

time and labor consuming. Ai et al. [2] treated estimating propensity scores as a dual problem of unbiased learning to rank [16]. As the propensity scores can only be used to reweigh documents with clicks in their model and only relevant documents are clicked, so they reweigh each document with its probability to be relevant. Both the propensity model and the ranker are parameterized by neural networks. Then, listwise objectives [6, 26] are employed to train the two models alternatively. In [14], an unbiased learning to rank algorithm is proposed based on the pairwise ranking algorithm LambdaMART [25]. Similar to [2], in unbiased LambdaMART, the propensity score model is learned along with the ranker by an alternating optimization algorithm. However, none of the existing unbiased learning to rank algorithms takes the unique context of e-commerce into consideration. Different from them, in this work, the proposed framework is developed to handle multiple types of implicit feedback and incorporate the unique user behavior patterns in grid-based product search into inverse propensity scoring. In particular, compared to unbiased LambdaMART which also utilizes a pairwise debiasing strategy and adopts LambdaMART, the proposed framework incorporates prior knowledge of users' behavior patterns to guide the learning process of propensity score models.

Grid-based Search. Nowadays, various types of websites including e-commerce, video and music streaming services show SERPs in a grids. Recently, in eye-tracking experiments, Xie et al. [27] observed three unique properties of users' behaviors in grid-based image search: middle bias, slower decay and row skipping. Based on the observations, for the sake of developing better evaluation metrics for grid-based search, they propose three novel click models to quantify how users' attention decays in such scenarios. We did not adopt these new evaluation metrics because without eye-tracking experiments we cannot obtain ground truth for the parameters of these evaluation metrics which quantify the decay of attention. Different from their focus, we propose to incorporate the row skipping and slower decay click models for propensity score modeling toward unbiased learning to rank. At the same time, grid-based search is still an open question for many other research problems like grid-based sponsored search.

E-commerce Search. Compared to traditional information retrieval, e-commerce search is confronted with some unique challenges such as its multi-objective nature and the need to explore new items for fairness among sellers as well as long-term user engagement [11, 24]. E-commerce search logs come with multiple types of implicit feedback (e.g., purchase and click). The target of e-commerce search is to maximize purchases or revenue of the website, however, due to the fact that purchases are much less frequently observed than other types of feedback such as clicks, it has been proposed to combine different types of feedback in the training objective [17, 19, 24]. In [19], authors found such hybrid objectives

help improve the search performance of fashion products on Amazon. In [24], a two-stage algorithm is proposed to integrate clicks and purchases through two separate machine learning models. In e-commerce search, we aim to help buyers explore unseen items, in [11], authors proposed a multi-armed bandit (MAB) method which allows exploration of items that are shown less than a certain times in a time interval. In terms of feature engineering, besides manually engineered features, recently, representation learning has been incorporated in e-commerce search [3, 21]. Regarding other aspects, Goswani et al. [10] also found that e-commerce search log data helps quantify the gap between customer demands and supplies. Different from them, our work is the first to develop a framework for unbiased learning to rank for e-commerce search.

7 CONCLUSION

In this work, we study the novel problem unbiased learning to rank algorithms in grid-based product search for e-commerce. This work is the first step toward handling the special challenges in this problem. In particular, the proposed framework utilizes multiple types of feedback and leverages users' behavior patterns in grid-based product search for propensity score modeling. We prove that the proposed loss function evaluated on implicit feedback data provides unbiased estimate of the ideal loss. We then motivate the usage of the row skipping and slower decay models for inverse propensity scoring justified through empirical evidence from data analysis. Finally, extensive experimental results show the effectiveness of the proposed framework across browsing devices and product taxonomies in datasets collected from a real-world e-commerce website. Future work includes (1) modeling propensity with meta information from SERPs, (2) relaxation of the joint examination hypothesis to handle multiple types of feedback, and (3) strategies to address products with low or no feedback in evaluation metrics.

ACKNOWLEDGEMENTS

This material is partially based upon work supported by the National Science Foundation (NSF) Grant #1614567 and #1909555.

REFERENCES

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

[2] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased Learning to Rank with Unbiased Propensity Estimation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 385–394.

[3] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W Bruce Croft. 2017. Learning a hierarchical embedding model for personalized product search. In *SIGIR*. ACM, 645–654.

[4] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

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

[6] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *ICML*. ACM, 129–136.

[7] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *WSDM*. ACM, 87–94.

[8] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *JMLR* 4, Nov (2003), 933–969.

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

[10] Anjan Goswami, Prasant Mohapatra, and Chengxiang Zhai. 2019. Quantifying and Visualizing the Demand and Supply Gap from E-commerce Search Data using Topic Models. In *Companion Proceedings of WWW*. ACM, 348–353.

[11] Anjan Goswami, ChengXiang Zhai, and Prasant Mohapatra. 2018. Towards Optimization of E-Commerce Search and Discovery. In *The 2018 SIGIR Workshop On eCommerce*.

[12] Ruocheng Guo, Lu Cheng, Jundong Li, P Richard Hahn, and Huan Liu. 2018. A survey of learning causality with data: Problems and methods. *arXiv preprint arXiv:1809.09337* (2018).

[13] Malay Haldar, Mustafa Abdool, Prashant Ramanathan, Tao Xu, Shulin Yang, Huizhong Duan, Qing Zhang, Nick Barrow-Williams, Bradley C Turnbull, Brendan M Collins, et al. 2019. Applying deep learning to Airbnb search. In *SIGKDD*. ACM, 1927–1935.

[14] 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.

[15] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *SIGKDD*. ACM, 133–142.

[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 *SIGIR*. ACM, 475–484.

[18] Alistair Moffat and Justin Zobel. 2008. Rank-biased precision for measurement of retrieval effectiveness. *TOIS* 27, 1 (2008), 2.

[19] Daria Sorokina and Erick Cantu-Paz. 2016. Amazon search: The joy of ranking products. In *SIGIR*. ACM, 459–460.

[20] Andrew Stanton, Liangjie Hong, and Manju Rajashekhar. 2018. Buzzsaw: A System for High Speed Feature Engineering. In *SysML*.

[21] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Learning latent vector spaces for product search. In *CIKM*. ACM, 165–174.

[22] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *SIGIR*. ACM, 115–124.

[23] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position bias estimation for unbiased learning to rank in personal search. In *WSDM*. ACM, 610–618.

[24] Liang Wu, Diane Hu, Liangjie Hong, and Huan Liu. 2018. Turning clicks into purchases: Revenue optimization for product search in e-commerce. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 365–374.

[25] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval* 13, 3 (2010), 254–270.

[26] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *ICML*. ACM, 1192–1199.

[27] Xiaohui Xie, Jiaxin Mao, Yiqun Liu, Maarten de Rijke, Yunqiu Shao, Zixin Ye, Min Zhang, and Shaoping Ma. 2019. Grid-based Evaluation Metrics for Web Image Search. (2019).