Revisiting Two-tower Models for Unbiased Learning to Rank

Le Yan
Google
lyyanle@google.com

Zhen Qin
Google
zhenqin@google.com

Honglei Zhuang
Google
hlz@google.com

Xuanhui Wang
Google
xuanhui@google.com

Michael Bendersky
Google
bemike@google.com

Marc Najork
Google
najork@google.com

ABSTRACT

Two-tower architecture is commonly used in real-world systems for Unbiased Learning to Rank (ULTR), where a Deep Neural Network (DNN) tower models unbiased relevance predictions, while another tower models observation biases inherent in the training data like user clicks. This two-tower architecture introduces inductive biases to allow more efficient use of limited observational logs and better generalization during deployment than single-tower architecture that may learn spurious correlations between relevance predictions and biases. However, despite their popularity, it is largely neglected in the literature that existing two-tower models assume that the joint distribution of relevance prediction and observation probabilities are completely factorizable. In this work, we revisit two-tower models for ULTR. We rigorously show that the factorization assumption can be too strong for real-world user behaviors, and existing methods may easily fail under slightly milder assumptions. We then propose several novel ideas that consider a wider spectrum of user behaviors while still under the two-tower framework to maintain simplicity and generalizability. Our concerns of existing two-tower models and the effectiveness of our proposed methods are validated on both controlled synthetic and large-scale real-world datasets.

CCS CONCEPTS

• **Information systems** → **Information retrieval**;

KEYWORDS

Unbiased Learning to Rank; Expectation Maximization; Bias Factorization

ACM Reference Format:
Le Yan, Zhen Qin, Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork . 2022. Revisiting Two-tower Models for Unbiased Learning to Rank. In *Proceedings of the 45th Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3477495.3531837

1 INTRODUCTION

Learning to rank is an essential component for many real-world applications [16, 20, 27]. Unbiased Learning To Rank (ULTR) has drawn much attention recently due to their promises to mitigate biases in real-world user feedback signals. Two-tower *additive* models are popular in practice [7, 10, 11, 28] to debias user feedback data due to their simplicity and effectiveness. In these models, one Deep Neural Network (DNN) tower takes regular input features to model unbiased relevance predictions, and another tower takes bias-related features, such as position and platform (e.g., mobile vs desktop). The outputs of these two towers are added together to explain observed user feedback logs during offline training, and only the unbiased prediction tower is effectively used during online serving. Two-tower additive models are easy to implement, interpret, and are sound in theory by following the Position Based Model (PBM) click model [8] to model user behaviors.

Despite their popularity, additive two-tower models have their limitations due to the assumptions made. More specifically, by following PBM, they assume that relevance prediction and observation probability are completely factorizable without any confounding variables between them. They also assume the observed utility is a first-order multiplication between relevance and observation probability. One should realize the PBM is just one kind of click models [8] and may miss important patterns in real-world noisy datasets: first, users may follow different click patterns, depending on factors such as user preference and query type (e.g., navigational vs browsing). In fact, click models are still being actively studied [15]. Second, even if the factorization is preferred due to its simplicity and generalizability, the first-order multiplication between two towers may be too limited to model real-world datasets. These limitations are largely neglected in the research community, since it is a common practice to study unbiased learning algorithms after generating synthetic data by following such assumptions [3].

In this work, we revisit two-tower models for ULTR, and show that additive models are inherently incapable to model the spectrum of user behaviors. We investigate several novel methods that can fit more diverse user behaviors, including user-based Expectation-Maximization (EM) methods and embedding interaction methods, both of which are still under the two-tower framework to largely maintain their simplicity and generalizability. We use controlled, but more complex and realistic synthetic datasets than existing ones to show when additive two-tower models may fail, and further validate our findings on a large-scale real-world dataset.

2 PROBLEM FORMULATION

Consider an online search or recommendation service. Given a query/user q in session s, we rank n candidate documents $\{d_i\}_n$ and present this list of documents to the user at positions $\{k_i^s\}_n$

SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07.
https://doi.org/10.1145/3477495.3531837

with $k_i^s \in \{1, 2, ..., n\}$. We then obtain a list of user interactions, say clicks, on these documents $\{c_i^s\}_n$, where $c_i^s \in \{1, 0\}$ indicating either clicked or not. Our goal is to learn the click probability for each document at a position $p(c_i = 1|q, d_i, k_i = k)$ from a finite log of sessions.

General assumption for two-tower models. A well established and intuitive assumption [21] allowing us to learn effectively with limited user interaction sessions and improve generalization capability is that, in a session, the click is conditioned on the observation at a given position; upon observation, the click only depends on the relevance between query and document, but not the position. Formally, the click probability of a document d_i at position k in a session s is,

$$p(c_i^s = 1|q, d_i, k_i^s = k) = p_{rel}(c_i^s = 1|q, d_i, s) \times p_{obs}(k_i^s = k|s). \quad (1)$$

Under this assumption, two-tower models, with one position bias tower to model the observation probability $p_{obs}(k)$ and one relevance tower to model the click probability given observation $p_{rel}(q, d_i)$, are widely adopted in ULTR. Note that Eq.(1) is not factorizable since both terms depend on the confounder s.

Additive model and its assumptions. The popular additive model [10, 26, 28, 29] is an instantiation of the general two-tower model family with extra assumptions. It assumes that the observation probability function $p_{obs}(k)$ and the click probability function $p_{rel}(q, d)$ are universal over sessions, thus ignoring the dependence on s in Eq.(1) and making it factorizable. In these models, the predictions of the relevance tower $r(q, d_i|\Theta_{rel})$ and the position bias tower $e(k|\Theta_{obs})$ with DNN parameters Θ_{rel} and Θ_{obs} correspondingly are usually combined in an additive way, thus the name of additive two-tower models, to predict the click probability:

$$p(c_i = 1|q, d_i, k) = f(g(r(q, d_i|\Theta_{rel})) + h(e(k|\Theta_{obs}))), \quad (2)$$

where $f(\bullet)$, $g(\bullet)$, and $h(\bullet)$ are model-dependent functions. For example, Regression Expectation-Maximization [26] and Position-bias Aware Learning [10] have $f = \exp$ and $g = h = \ln$, so that the relevance tower directly predicts $p_{rel}(c|q, d_i)$ and the position bias tower predicts the observation probability $p_{obs}(k)$. Another common practice [11, 28] uses the sigmoid function $\sigma(x) = \frac{\exp(x)}{1+\exp(x)}$ for f, and the logit function, i.e., the inverse of sigmoid, for g and h. Note that as most existing work only considers position in $e(k|\Theta_{obs})$, additive model directly follows the popular PBM click model.

When do additive models fail? In real-world data, the assumption to ignore dependence on s could be too strong, and PBM is just one click model in the rich literature. Intuitively, different sessions could have very diverse characteristics depending on factors such as user behavior and nature of the query. For example, a user who is patient would check through the entire ranking list, i.e., $p_{obs}(k_i = k|s) = 1$, so a click only depends on relevance, following the Document-based CTR model (DCTR). Even for the same user, depending on the time and the purpose of the query, the behaviors could differ. For example, a user issuing a browsing query might click purely based on positions, but not the content of the item, following the Rank-based CTR model (RCTR). In other words, real-world click data is likely generated under a mixture of different click behaviors conditioned on observation and relevance. Thus, a more general approach is to instantiate with some hidden parameters u^s and v^s, (for instance, u quantifies the browsy nature of the queries and v measures user patience in above examples),

$$p(c_i^s = 1|q, d_i, k_i^s = k) = p_{rel}(c_i^s = 1|q, d_i, u^s) \times p_{obs}(k_i^s = k|v^s),$$

and the overall click probability will be

$$p(c_i = 1|q, d_i, k_i = k) = \int dP(u^s, v^s) p_{rel}(c_i^s = 1|q, d_i, u^s) \times p_{obs}(k_i^s = k|v^s), \quad (3)$$

where $P(u^s, v^s)$ is the distribution of the hidden variables.

As the hidden variables u^s and v^s are in general not independent ($dP(u^s, v^s) \neq dP_u(u^s)dP_v(v^s)$), the click probability are *no longer factorizable* as in Eq.(2). As a result, the additive models in Eq.(2) relying on the factorization of position-dependent observation and relevance-dependent click are likely not optimal for the real-world biased data. To verify this, in the following sections, we will propose two more general two-tower based methods and then test their superiority in synthetic and real-world user click data with diverse user behaviors.

3 METHODS

We introduce two new approaches to address the limitations of existing methods.

3.1 Mixture Expectation-Maximization

To incorporate a wide spectrum of click behaviors, we design a general Expectation-Maximization (EM) algorithm to automatically infer the hidden user behaviors during learning from complex real-world click data. Consider that we have a *set* of two-tower models that potentially capture different user patterns. In the EM-algorithm, we first assign data points from different sessions to one of the models, and then train the model parameters with corresponding data, and iterate till convergence. Specifically, to assign the data points, we compute the likelihoods of models for a session, $p(\alpha|s)$, where α indexes the models in the set with a normalization constraint $\sum_\alpha p(\alpha|s) = 1$.

In the M-step, we train the model α, with loss,

$$\mathcal{L}^\alpha = \sum_s p(\alpha|s)\ell_\alpha(s),$$

where $p(\alpha|s)$ is fixed and serves as the weight for session s and $\ell_\alpha(s)$ is the loss of model α for a session s. ℓ_α can be instantiated as the sigmoid cross-entropy loss:

$$\ell_\alpha(s) = -\sum_{i \in D_q} \left(c_i^s \ln \frac{e^{f_\alpha(q,d_i,k_i)}}{1 + e^{f_\alpha(q,d_i,k_i)}} + (1 - c_i^s) \ln \frac{1}{1 + e^{f_\alpha(q,d_i,k_i)}} \right),$$

where $f_\alpha(q^s, d_i^s, k_i^s)$ is the logit prediction of model α for query q and document d_i at position k_i in session s.

In the E-step, we estimate the probability $p(\alpha|s)$, using the cross-entropy as a good evaluation of the likelihood,

$$p(\alpha|s) = \frac{1}{Z_s} \exp(-\ell_\alpha(s)/T) = \frac{1}{Z_s} \prod_i \left(\left[\frac{e^{f_\alpha(q,d_i,k_i)}}{1 + e^{f_\alpha(q,d_i,k_i)}} \right]^{c_i^s} \left[\frac{1}{1 + e^{f_\alpha(q,d_i,k_i)}} \right]^{1-c_i^s} \right)^{1/T},$$

Table 1: Factorization ability of clicks from different mixture click models

Ratio	Mixture Click Model	Factorizable	Position Bias
0:0:0:1	PBM	✓	✓
0:1:0:1	RCTR+PBM	✓	✓
1:0:1:0	RCM+DCTR	✓	✗
1:1:1:1	RCM+RCTR+DCTR+PBM	✗	✓
0:1:1:0	RCTR+DCTR	✗	✓

where $Z_s = \sum_\alpha \exp(-\ell_\alpha(s)/T)$ is the normalization factor, and T is a hyperparameter controlling model confidence assignment. When $T \to 0$, we always assign the data point to the model with the minimal loss ℓ with 100% confidence. When $T \to \infty$, we always mix and train all models equally on each data point.

At serving, we always rank based on the query document relevance prediction by the unbiased tower of models.

3.2 Embedding-based Interaction

A more generic technique than first-order multiplication to model the complex non-factorizable interactions of relevance and observation bias is to use higher-order interactions based on embeddings. Instead of predicting a logit or probability from each of the two towers, embedding-based methods leverage the embedding vectors from the DNN towers and dot product interactions of embeddings to niche the complex interaction beyond the additive models. In this work, we consider two specific embedding-based models: Embedding Dot-product model and Embedding Interaction model.

For the embeddings $\vec{r}(q,d)$ by the relevance tower and $\vec{e}(k)$ by the position bias tower with D_{emb} embedding dimension, the embedding dot-product model makes a dot-product of the embeddings to predict logit of click probability,

$$f^{\text{EDot}}(C=1|q,d,k) = \vec{r}(q,d)\cdot\vec{e}(k), \tag{4}$$

and the embedding interaction model leverages a quadratic interaction of embeddings,

$$f^{\text{EInter}}(C=1|q,d,k) = \vec{r}(q,d)\cdot B\cdot\vec{e}(k)+\vec{b}_r\cdot\vec{r}(q,d)+\vec{b}_e\cdot\vec{e}(k)+b, \tag{5}$$

where $B, \vec{b}_r, \vec{b}_e$, and b are trainable parameters with B a $D_{emb}\times D_{emb}$ matrix, $\vec{b}_r$ and $\vec{b}_e$ D_{emb}-dim vectors, and b a scalar.

Different from the additive models and the mixture EM model above, as the position dependence cannot be factorized in Eqs.(4, 5), these embedding-based models require canonical position features fed in to make the predictions at serving. For example, canonical position 1 is commonly used [4].

4 EXPERIMENTS

In this section, we conduct experiments on both synthetic dataset and real-world click dataset.

4.1 Synthetic Dataset

4.1.1 Yahoo LTR Dataset. We create a click dataset with synthetic clicks using Yahoo Learning to Rank Set1 [5]. To generate synthetic clicks in the training set, we first train a Ranking SVM as the initial ranker based on 1% of the labeled training data, similar to previous studies [3]. We then use the initial ranker to rank all the items in each query and use the ranking starting from 1 as the synthetic serving position k_i for document i. Next, we generate synthetic clicks with the mixture click model to simulate the diverse user behavior in synthetic clicks.

4.1.2 Mixture click model. We consider a random mixture of the four most fundamental click models [8] (our proposed methods generalize to more click models): Random Click Model (RCM):

$$p(C=1) = \rho, \tag{6}$$

Rank-based CTR Model (RCTR):

$$p(C=1) = \rho_k, \tag{7}$$

Document-based CTR Model (DCTR):

$$p(C=1) = \rho_{q,d}, \tag{8}$$

and Position Based Model (PBM):

$$p(C=1) = \omega_{q,d}\gamma_k. \tag{9}$$

In a given session, we randomly choose one of the click models according to the predefined weights and generate synthetic clicks from the chosen model. Thus, in a training batch, we would see clicks generated from different click models. In this work, we will use the weight ratios as a shorthand name of the mixture click model: for example, 0:1:1:0 stands for a mixture of RCTR and DCTR with equal occurrence rate. All the click models studied in this work are summarized in Table 1. In particular, we use $\rho = 0.1$ for RCM Eq.(6), $\rho_k = 0.5\gamma_k$ for RCTR Eq.(7), $\rho_{q,d_i} = 0.5\omega_{q,d_i}$ for DCTR Eq.(8), $\gamma_k = \frac{1}{k}$ and $\omega_{q,d_i} = 0.1 + 0.9\frac{2^{y_i}-1}{2^{y_{\max}}-1}$ for PBM Eq.(9), where y_i is the relevance label of query document pair q, d_i. This label ranges from (0, 1, 2, 3, 4) with $y_{\max} = 4$ in Yahoo dataset. Under such simplification, clicks from some mixture models can then be factorized as,

$$p(c_i = 1|q,d_i,k_i = k) = E(k)R(q,d_i) \tag{10}$$

The possibilities to do such factorization of studied click models is also summarized in Table 1.

4.2 Real-World Dataset

We also run experiments on a real-world dataset collected from Google Chrome Web Store (CWS) user logs. Each session logs a set of extensions displayed to the user, their positions in the layout, and the interactions (including both clicks and installs). We extract several numerical and categorical features for each extension, such as number of impressions in the last two weeks. The same set of features are used for all compared methods. For each categorical feature, we learn an embedding vector. We concatenate all the embedding vectors with the numerical feature vector as the representation of the item. We collect 30 days of sampled logs with the first 28 days as training data and the following 2 days as test data. We only use clicks as labels in the training, and use installs in evaluation.

4.3 Metrics

To evaluate the model performance, we use Normalized Discounted Cumulative Gain (NDCG) on the ground truth relevance label for the synthetic dataset. For the real-world dataset, as the ground truth unbiased labels are missing, we evaluate the results using biased

Table 2: The methods we compared in our experiments.

Method	Description
Biased	Baseline model not using any bias-related features
REM	Regression EM, a commonly studied additive model [26]
Additive	Click logits are the sum of two tower logits [10, 28]
EDot	The Embedding Dot-product model in Eq.(4)
EInter	The Embedding Interaction model in Eq.(5)
MixEM	The EM method in Sec. 3

click labels and install labels, leveraging the rich counterfactual evaluation literature [22]. For click labels, we consider raw NDCG metric, NDCG metric corrected with Inverse Propensity Scores (IPS), which is computed with the average clicks at given position (not the exact IPS computed in the random experiments). We also report the NDCG metrics using installs as labels. Given that there are no ground-truth relevance labels, we intentionally report a wide range of metrics on the real-world dataset and assume a better model would perform better on consensus over these metrics.

4.4 Methods

The methods we studied are summarized in Table 2. Based on the mixture click model above, we apply the MixEM method to four click patterns in both synthetic and real-world datasets. Using richer click patterns might produce even better results on complex real-world dataset. Specifically, we have the $\alpha = 0$ model for RCM, $\alpha = 1$ for RCTR, $\alpha = 2$ for DCTR, and $\alpha = 3$ for PBM, whose logit predictions are,

$$
\begin{aligned}
f_0 &= \theta_0; \\
f_1 &= \theta_1 + e(k|\Theta_{obs}); \\
f_2 &= \theta_2 + r(q, d|\Theta_{rel}); \\
f_3 &= e(k|\Theta_{obs}) + r(q, d|\Theta_{rel}),
\end{aligned}
$$

where θ_0, θ_1, θ_2, and Θ_{obs}, Θ_{rel} are trainable parameters, $e(\bullet)$ and $r(\bullet)$ are DNN towers for the position bias and relevance correspondingly. For real-world data, through we don't know click patterns *a priori*, we still apply the mixture of the same four models, the linear combination of which covers sufficient range of click patterns.

We do a grid search over the hyperparameter T for MixEM, and D_{emb} for EDot and EInter. We present the results of the hyperparameters with the best validation performance on NDCG.

4.5 Results

The main results are summarized below. (1) Additive models work well when click models are factorizable, but are significantly less competitive when click models are not, see Table 3. (2) The proposed Mixture EM and Embedding-based methods can be less competitive when click model is isomorphic with the additive model (e.g. PBM), but are in general better than the additive models, especially significantly better when click model is exactly not factorizable (e.g. 0:1:1:0), see Table 3. (3) Mixture EM shows a significant better performance than the additive models on the CWS dataset, as in Table 4, indicates that the real-world clicks are more likely to be consistent with diverse user behaviors than a uniform click pattern assumed in existing works.

Table 3: NDCG@5 of relevance label on Yahoo. Up arrow ↑ and down arrow ↓ indicate statistical significant better and worse than Additive baseline, with p-value=0.01.

Click Model	Biased	REM	Additive	EDot	EInter	MixEM
0:0:0:1	0.6482$^{\downarrow}$	**0.6894**	0.6866	0.6325$^{\downarrow}$	0.6837	0.6620$^{\downarrow}$
0:1:0:1	0.6282$^{\downarrow}$	**0.6766**$^{\uparrow}$	0.6717	0.6191$^{\downarrow}$	0.6724	0.6410$^{\downarrow}$
1:0:1:0	0.6799$^{\downarrow}$	0.6864	0.6865	0.6782$^{\downarrow}$	**0.6904**$^{\uparrow}$	0.6773$^{\downarrow}$
1:1:1:1	0.6462$^{\downarrow}$	0.6818	0.6839	0.6551$^{\downarrow}$	**0.6867**	0.6834
0:1:1:0	0.6511$^{\downarrow}$	0.6768$^{\downarrow}$	0.6859	0.6676$^{\downarrow}$	**0.6914**$^{\uparrow}$	0.6883$^{\uparrow}$

Table 4: Ranking Metrics of clicks on CWS. Up arrow ↑ and down arrow ↓ indicate statistical significant better and worse than the Additive baseline, with p-value=0.01.

Methods	NDCG@5	IPS-NDCG@5	NDCG@5 (install)	IPS-NDCG@5 (install)
Biased	0.4995$^{\uparrow}$	0.4835	0.3108	0.3050
REM	0.4847$^{\downarrow}$	0.4749$^{\downarrow}$	0.3060$^{\downarrow}$	0.3025$^{\downarrow}$
Additive	0.4920	0.4808	0.3104	0.3063
EDot	0.4967$^{\uparrow}$	0.4839$^{\uparrow}$	0.3120	0.3074
EInter	0.4968$^{\uparrow}$	0.4842$^{\uparrow}$	0.3119	0.3073
MixEM	**0.5030**$^{\uparrow}$	**0.4898**$^{\uparrow}$	**0.3206**$^{\uparrow}$	**0.3157**$^{\uparrow}$

From the experiments on the synthetic data of Yahoo, shown in Table 3, we have the following observations. (1) Good unbiased methods always perform better than the biased baseline, except for 1:0:1:0, which is RCM plus DCTR with no position bias. (2) Additive models, REM and Additive, perform very comparable on all different click patterns except for the 0:1:1:0 model, which is hardest in our experiments for the factorized models. (3) Both additive models work extremely well on PBM clicks and quite well on all the click patterns that are factorizable as Eq.(10). (4) Additive models work also reasonably well on one of non-factorizable click patterns (1:1:1:1), potentially because some patterns are nearly factorizable with just some constant differences. (5) MixEM is not as competitive as the additive models, especially when click patterns are factorizable, but becomes comparable for nearly factorizable clicks and significantly better than the additive models for the exactly non-factorizable case 0:1:1:0. (6) Among the embedding-based methods, EInter performs almost always better than EDot, potentially due to higher capacity in the interaction of embeddings. (7) EInter performs almost always the best regardless of factorization of the click patterns. It could potentially be explained by the fact that EInter will reduce to Additive when quadratic interaction vanishes, $B = 0$.

The experiment results on the real-world user interaction dataset CWS are summarized in Table 4. Compared with the synthetic data, a key observation is that the additive model can no longer beat even the biased baseline, which indicates that the factorizable model of click patterns may largely no longer holds in real-world datasets. On the other hand, we observe a significant better performance of MixEM on all metrics over all additive models and the biased baseline. These observations validate our concern on the unrealistic assumption made by existing methods, and more diverse user

behaviors should be considered for real-world unbiased learning to rank problems, where MixEM could be a promising method.

5 RELATED WORK

PAL [10] is the pivot work that introduces the two-tower model to the research community, but the methodology itself, which to the best of our knowledge, has been extensively used in the industry before the publication. Zhao et al. [28] uses the two-tower model for recommending which video to watch on YouTube. Haldar et al. [11] applies a two-tower model on Airbnb search. Huang et al. [13] directly models interactions between items and positions in one single model, which significantly complicates the learning space, leading to data hungry and generalization issues.

Another closely related family of methods, Inverse Propensity Weighting (IPW) based methods [1, 2, 12, 14, 18, 19, 24–26], follow the same Position Based Model assumption (except for very few recent work [23]). In this work, we focus on the discussion of two-tower models due to their popularity, but the discussed concerns and methodologies may generalize to IPW-based methods since they follow the same assumptions. We also note that more advanced click models [6, 9, 15, 17, 29] can be explored in future work.

6 CONCLUSION

In this work, we revisit the two-tower models for unbiased learning to rank. We first show that the commonly used additive models could fail due to over-simplified assumptions of the potentially diverse user behaviors. We then study two new classes of two-tower based methods, the mixture EM method and the embedding-based interaction method, and show that the two proposed methods can perform superior to the additive models and the biased baselines on both a synthetic dataset and a real-world user click dataset, especially when the user click patterns do not follow the assumptions in factorizable models. We hope our work could call up more inspirations on counting the diverse user behavior effect in unbiased learning-to-rank.

ACKNOWLEDGEMENT

We thank Po Hu, Xinyu Qian, and Janelle Lee for their expertise and assistance in experimenting on the Chrome Web Store dataset.

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A general framework for counterfactual learning-to-rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 5–14.

[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*. 385–394.

[3] Qingyao Ai, Tao Yang, Huazheng Wang, and Jiaxin Mao. 2021. Unbiased learning to rank: online or offline? *ACM Transactions on Information Systems (TOIS)* 39, 2 (2021), 1–29.

[4] Alexey Borisov, Ilya Markov, Maarten de Rijke, and Pavel Serdyukov. 2016. A Neural Click Model for Web Search. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. 531–541.

[5] Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview. In *Proceedings of the Learning to Rank Challenge*. 1–24.

[6] Olivier Chapelle and Ya Zhang. 2009. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*. 1–10.

[7] Wenjie Chu, Shen Li, Chao Chen, Longfei Xu, Hengbin Cui, and Kaikui Liu. 2021. A General Framework for Debiasing in CTR Prediction. *arXiv preprint arXiv:2112.02767* (2021).

[8] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. *Click Models for Web Search*. Morgan & Claypool.

[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*. 331–338.

[10] Huifeng Guo, Jinkai Yu, Qing Liu, Ruiming Tang, and Yuzhou Zhang. 2019. PAL: a position-bias aware learning framework for CTR prediction in live recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 452–456.

[11] Malay Haldar, Prashant Ramanathan, Tyler Sax, Mustafa Abdool, Lanbo Zhang, Aamir Mansawala, Shulin Yang, Bradley Turnbull, and Junshuo Liao. 2020. Improving deep learning for airbnb search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2822–2830.

[12] 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*. 2830–2836.

[13] Jianqiang Huang, Ke Hu, Qingtao Tang, Mingjian Chen, Yi Qi, Jia Cheng, and Jun Lei. 2021. *Deep Position-Wise Interaction Network for CTR Prediction*. 1885–1889.

[14] 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*. 781–789.

[15] Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Shuai Li, Ruiming Tang, Xiuqiang He, Jianye Hao, and Yong Yu. 2021. A Graph-Enhanced Click Model for Web Search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1259–1268.

[16] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.* (2009).

[17] Jianling Sun Mouxiang Chen, Chenghao Liu and Steven C.H. Hoi. 2021. Adapting Interactional Observation Embedding for Counterfactual Learning to Rank. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[18] Harrie Oosterhuis and Maarten de Rijke. 2020. *Policy-Aware Unbiased Learning to Rank for Top-k Rankings*. 489–498.

[19] Zhen Qin, Suming J. Chen, Donald Metzler, Yongwoo Noh, Jingzheng Qin, and Xuanhui Wang. 2020. Attribute-Based Propensity for Unbiased Learning in Recommender Systems: Algorithm and Case Studies. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2359–2367.

[20] Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2021. Are Neural Rankers still Outperformed by Gradient Boosted Decision Trees?. In *International Conference on Learning Representations*.

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

[22] Yi Su, Lequn Wang, Michele Santacatterina, and Thorsten Joachims. 2019. Cab: Continuous adaptive blending for policy evaluation and learning. In *International Conference on Machine Learning*. 6005–6014.

[23] Ali Vardasbi, Maarten de Rijke, and Ilya Markov. 2020. Cascade Model-Based Propensity Estimation for Counterfactual Learning to Rank. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2089–2092.

[24] Nan Wang, Zhen Qin, Xuanhui Wang, and Hongning Wang. 2021. Non-Clicks Mean Irrelevant? Propensity Ratio Scoring As a Correction. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*.

[25] 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 (SIGIR '16)*. 115–124.

[26] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position bias estimation for unbiased learning to rank in personal search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 610–618.

[27] Le Yan, Zhen Qin, Rama Kumar Pasumarthi, Xuanhui Wang, and Mike Bendersky. 2021. Diversification-Aware Learning to Rank using Distributed Representation. In *The Web Conference*.

[28] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending What Video to Watch next: A Multitask Ranking System. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 43–51.

[29] Honglei Zhuang, Zhen Qin, Xuanhui Wang, Michael Bendersky, Xinyu Qian, Po Hu, and Dan Chary Chen. 2021. Cross-positional attention for debiasing clicks. In *Proceedings of the Web Conference 2021*. 788–797.