Metric-agnostic Ranking Optimization

Qingyao Ai
Dept. CS&T, Tsinghua University
Zhongguancun Laboratory
Beijing, China
aiqy@tsinghua.edu.cn

Xuanhui Wang
Google Research
Mountain View, CA, USA
xuanhui@google.com

Michael Bendersky
Google Research
Mountain View, CA, USA
bemike@google.com

ABSTRACT

Ranking is at the core of Information Retrieval. Classic ranking optimization studies often treat ranking as a sorting problem with the assumption that the best performance of ranking would be achieved if we rank items according to their individual utility. Accordingly, considerable ranking metrics have been developed and learning-to-rank algorithms that have been designed to optimize these simple performance metrics have been widely used in modern IR systems. As applications evolve, however, people's need for information retrieval have shifted from simply retrieving relevant documents to more advanced information services that satisfy their complex working and entertainment needs. Thus, more complicated and user-centric objectives such as user satisfaction and engagement have been adopted to evaluate modern IR systems today. Those objectives, unfortunately, are difficult to be optimized under existing learning-to-rank frameworks as they are subject to great variance and complicated structures that cannot be explicitly explained or formulated with math equations like those simple performance metrics. This leads to the following research question – how to optimize result ranking for complex ranking metrics without knowing their internal structures? To address this question, we conduct formal analysis on the limitation of existing ranking optimization techniques and describe three research tasks in *Metric-agnostic Ranking Optimization*: (1) develop surrogate metric models to simulate complex online ranking metrics on offline data; (2) develop differentiable ranking optimization frameworks for list or session level performance metrics without fine-grained supervision signals; and (3) develop efficient parameter exploration and exploitation techniques for ranking optimization in metric-agnostic scenarios. Through the discussion of potential solutions to these tasks, we hope to encourage more people to look into the problem of ranking optimization in complex search and recommendation scenarios.

KEYWORDS

Information Retrieval, Learning to Rank, Ranking Optimization

1 INTRODUCTION

Ranking is at the core of many Information Retrieval (IR) applications. Examples include but not limited to web page ranking on search engine result pages (SERPs), item ranking for recommendation, question/answer ranking for conversational search, etc. Ranking performance affects not only the quality and efficiency of information access, but also high-level user experience such as engagement and satisfaction. Therefore, ranking optimization has long been considered as a key component of modern IR systems and a core problem in IR research community.

Classic ranking optimization studies often treat ranking as a sorting problem. Based on the Probability Ranking Principle (PRP) [71], they assume that the utility of a ranking candidate can be quantified as a scalar, and the quality of ranking can be maximized by showing results following the decreasing order of their utility scores. This assumption essentially indicates that ranking performance can be measured by the individual utility of items on each position (with or without weighting schemes), and ranking tasks can be formulated as a set of partial order prediction problems. Therefore, learning-to-rank algorithms that build machine learning (ML) models to directly optimize training loss constructed from the weighting schemes of such ranking metrics can achieve the best performance in theory and have been shown to be effective in many ranking applications. For simplicity, we refer to the ranking metrics built from PRP as *simple ranking metrics* and learning-to-rank methods designed for them as *simple ranking optimization* algorithms.

As applications evolve, however, recent studies have found a significant gap between the performance scores computed with simple ranking metrics and the actual ranking quality perceived by users [7, 45, 98]. In practice, users do not evaluate result utility independently in ranking following the PRP assumption. Their satisfaction on rankings is often affected by complicated factors such as the relations between results, the local context of the request, personal preferences of the users, etc. Thus, research has been conducted in IR to provide better understanding on user behaviors in online systems, which results in numerous user-centric evaluation metrics such as satisfaction scores [23, 55] and engagements [52, 54, 102]. We refer to these as *complex ranking metrics*.

Despite the extensive studies on ranking evaluation in complex interaction scenarios, the techniques to support ranking optimization for complex ranking metrics are very much under development. Unsurprisingly, it is non-trivial, if not impossible, to apply existing simple ranking optimization techniques to optimize complex ranking metrics. First, complex ranking metrics, especially those computed with user logs sampled from online systems, are subject

to extensive noise produced by systematic bias or unpredictable user behaviors. Because simple ranking optimization algorithms assume that each item have unique and fixed utility following the PRP assumption, the uncertainty in complex ranking metrics could easily confuse the algorithms and produce unreliable models. Second, most existing complex ranking metrics are constructed on list levels instead of item levels. It is often difficult to decompose them into an aggregation of partial orders or pointwise losses on items, which are necessary for existing simple ranking optimization techniques. Third, and most importantly, the modeling of complex ranking metrics itself is an unsolved research problem in IR and it is widely believed that it is impossible to derive a single mathematical framework to explicitly describe or interpret user-related ranking metrics [60]. In other words, the structures of complex ranking metrics could be *agnostic*. This is unacceptable in existing ranking optimization frameworks as they need explicit loss functions constructed based on target ranking metrics and the predicted utility (i.e., the ranking score) of each item in order to apply well-established algorithms (e.g., gradient descent) to optimize model parameters. This leads to the following question: *How to optimize ranking without knowing the internal structures of ranking metrics?*

We believe that the answer to this question, which we refer to as *Metric-agnostic Ranking Optimization*, would be to develop new sets of ranking theory and techniques on metric modeling, parameter optimization, and ranking construction. Those techniques would not only provide new ideas for ranking formulation, but also serve as the foundation of ranking optimization for complicated IR problems beyond the PRP assumptions and frameworks. More specifically, to address metric-agnostic ranking optimization, we propose and discuss three key tasks in this paper, which are (1) *structure-agnostic metric modeling and simulation*: how to build offline surrogate models for complex ranking metrics without extensive online experiments; (2) *differentiable ranking process with coarse-grained reward*: how to make ranking functions or processes differentiable to coarse-grained reward on list or session level. and (3) *efficient ranking exploration in parameter space*: how to conduct efficient and effective ranking exploration in parameter space without the PRP assumption. Through the formulation and discussion of potential solutions to these tasks, we hope to inspire novel ideas and encourage more people to look into the problem of ranking optimization with complex ranking metrics.

2 STRUCTURE-AGNOSTIC METRIC MODELING AND SIMULATION

While there is no universal definition on what complex ranking metrics should look like, three characteristics are widely acknowledged to be key factors that make a ranking metric complex for ranking optimization: (1) complex ranking metrics are usually noisy and subject to large variance; (2) complex ranking metrics are usually computed on list or session level, and (3) complex ranking metrics are often collected online with significant cost. Take user engagement as an example. The time people spend on a search/recommendation result pages varies greatly depending on their personalities. This means that no universal threshold can be defined to distinguish good rankings from bad ones according to engagement time. Also, user engagement time on each result is often difficult to collect. In many cases, we only know how much time the user has spent

on the whole result page but not those on each individual result. Last but not least, while engagement time has been shown to be a valuable indicator of ranking performance, it's also expensive to get because it requires real users to interact with the online systems, which restricts both the size and the efficiency of data collection.

Due to these characteristics, direct optimization of complex ranking metrics is mostly impossible under existing learning-to-rank frameworks. To the best of our knowledge, most existing studies on learning to rank formulate ranking as a sorting task where the best ranking is deterministic given the partial orders of item pairs. The noisy nature of complex ranking metrics, however, indicates that the same ranking could receive different rewards in practice, which can easily confuse existing ranking optimization algorithms. Also, existing learning-to-rank algorithms require explicit labels, either gathered from user behaviors or domain experts, on each individual result for partial order prediction. These labels are mostly unavailable for complex ranking metrics because the metric scores are computed on list or session level. Further, when ranking metrics are computed on list levels, the training data built from online or offline search logs would be extremely sparse as observed data are only a tiny portion of all possible rankings.

To address these problems, we need to bridge or alleviate the gap between the noisy and sparse nature of complex ranking metrics and the need of reliable and extensive training data for ranking optimization. We believe that one promising direction is to construct surrogate models that can predict complex ranking metric scores on any rankings without online experiments or hypothesizing on metric structures. Formally, let $\mathcal{I}$, I, and i be the universal set of ranking sessions, the set of items in a session, and an item to be ranked, respectively. Let $\pi(I)$ be a specific ranking of items $i \in I$. For a complex ranking metric $m(\pi(I))$, the goal of metric modeling is to construct a surrogate model $g(\pi(I), \theta^*)$ that could serve as a reliable approximation of $m(\pi(I))$ with parameter θ^*:

$$\theta^* = \arg\min_{\theta} \sum_{I \in \mathcal{I}} \sum_{\pi(I) \in \mathcal{R}(I)} l\big(g(\pi(I)|\theta), m(\pi(I))\big) \qquad (1)$$

where $\mathcal{R}(I)$ is the universal set of all possible rankings for I, and l can be defined as any types of errors function such as pairwise loss (i.e., for two rankings $\pi(I)$ and $\pi'(I)$, the loss is 0 if m and g have same preferences on them, and is positive otherwise). In the rest of this section, we describe how to solve this metric modeling problem offline. Specifically, we start from proposing new data sampling/weighting scheme for noise and variance quantification, and then move to the development of context-aware surrogate models for offline metric prediction. After that, we describe an offline reinforcement learning framework to enhance the generalizability of metric surrogate models for training data generation.

2.1 Noise and Variance Quantification

Noise and variance in ranking evaluation is a common problem caused by multiple factors such as restrictive annotation process [8, 15], bias in user behaviors [1, 49, 99], etc. Existing studies usually apply two types of methods to deal with noise in ranking metrics. The first one is to build user hypothesis through lab or field studies and use them to explicitly construct metric models to explain and eliminate noise in ranking evaluation [29, 47]. These methods are particular useful for specific types of noise, but their generalizability

is limited as user hypothesis could be inaccurate or out-of-date. The other type of methods is to build ML models and use large-scale training data to learn and calibrate ranking metrics [24]. While these methods are attractive due to their theoretical robustness, they are extremely data hungry. Unfortunately, none of these existing methods is applicable to metric modeling in metric-agnostic ranking optimization because we neither have enough knowledge about the metric structure nor large-scale training data with annotations.

In fact, the problem of noise quantification with limited data is not unique to metric modeling. For example, recent studies on unbiased learning and data sampling has received significant attention in the IR community. Though designed for simple ranking optimization scenarios (i.e., optimize ranking metrics built on PRP), these studies show the possibility of combining the modeling of data noise with the learning of ranking models for effective noise quantification and reduction with limited data. For instance, previous studies on unbiased learning to rank [3, 96] show that effective user bias and examination propensity models can be automatically built together with ranking models on search logs without online user experiments. More recently, studies on learning to rank with partial labeled data [19] also show that by combining the learning of negative sampling strategies with ranking models, we can effectively discover noisy training samples in ranking optimization.

Based on these observations, we propose to explore the possibility of collaboratively learning noise quantification and surrogate metric models for complex ranking metrics. Formally, given a complex ranking metric $m(\pi(I))$, we collect a set of training data $\mathcal{T} = \{\pi(I), m(\pi(I)) | I \in \mathcal{T}_\mathcal{I}, \pi(I) \in T_I\}$ with ranking sessions $\mathcal{T}_\mathcal{I} \subset \mathcal{I}$ where each session contains multiple rankings T_I. As discussed previously, $m(\pi(I))$ is not noise-free, and a direct quantification of noise in $m(\pi(I))$ is difficult due to the unknown structure of m and limited data in $\mathcal{T}$. Thus, instead of training a separate model to predict noise, we propose to construct a noise-observation model $o(m, \pi(I))$ together with the surrogate model $g(\pi(I), \theta)$. The key observation is that a good noise-observation model can help us identify noisy data in training and improve the metric surrogate model, and a good metric surrogate model can help the noise-observation model better estimate noise in data. Formally, let $(\pi^+(I), \pi^-(I))$ be a pair of rankings over I where $m(\pi^+(I)) > m(\pi^-(I))$. An example training loss of $g(\pi(I))$ (short for $g(\pi(I), \theta)$) can be defined as

$$L(g, \mathcal{T}) = -\sum_{I \in \mathcal{T}_\mathcal{I}} \sum_{\pi^+(I), \pi^-(I) \in T_I} \log Pr(\pi^+, \pi^- | g),$$
$$Pr(\pi^+, \pi^- | g) = \frac{e^{g(\pi^+(I))}}{e^{g(\pi^+(I))} + e^{g(\pi^-(I))}} \quad (2)$$

where we define $Pr(\pi^+, \pi^- | g)$ as the probability of $\pi^+(I)$ being better than $\pi^-(I)$ given $g(\pi(I))$. Intuitively, when there is a higher probability of observing noise on $m(\pi^+(I))$ than $m(\pi^-(I))$, the reliability of $(\pi^+(I), \pi^-(I))$ should be discounted and reflected in the training of $g(\pi(I))$. Inspired by counterfactual learning [3, 49], we propose to revise Eq. (2) by weighting data with

$$L(g, \mathcal{T}) = -\sum_{I \in \mathcal{T}_\mathcal{I}} \left(\sum_{\pi^+(I), \pi^-(I) \in T_I} (1 - Pr(\pi^+, \pi^- | o, m)) \log Pr(\pi^+, \pi^- | g) \right) \quad (3)$$

where we define $Pr(\pi^+, \pi^- | o, m) = \frac{e^{o(m, \pi^+(I))}}{e^{o(m, \pi^+(I))} + e^{o(m, \pi^-(I))}}$ as the probability of observing inaccurate data on $(\pi^+(I), \pi^-(I))$, and use

$1 - Pr(\pi^+, \pi^- | o, m)$ to weight the data pair. Similarly, we could also use $g(\pi(I))$ to revise our noise-observation model following the same weighting scheme and build training loss for $o(m, \pi(I))$ as

$$L(o, \mathcal{T}) = -\sum_{I \in \mathcal{T}_\mathcal{I}} \left(\sum_{\pi^+(I), \pi^-(I) \in T_I} (1 - Pr(\pi^+, \pi^- | g)) \log Pr(\pi^+, \pi^- | o, m) \right) \quad (4)$$

where the probability of observing noise should be low when the surrogate model g and the target metric m agree with each other. By iteratively optimize Eq. (3)&(4), the two models help each other and efficiently extract noise and build surrogate model together.

2.2 Metric Prediction with Local Context

The nature of metric-agnostic ranking optimization creates significant difficulty in training data collection as we can only obtain metric scores on a whole ranked list that has been shown to users with no fine-grained feedback on each item. The number of possible rankings are large in which only one or a tiny proportion of them has been shown to users. Since real-time ranking optimization with online user experiments are mostly prohibitive in practice, it's important to predict the metric scores, which we refer to as the *ranking rewards*, for rankings without involving real users. The most related studies on this direction are query performance prediction, which tries to predict the ranking performance of a retrieval system without actually showing the results to users [40, 101]. Methods proposed by these studies, however, are tailored for ad-hoc retrieval tasks [103] and can only predict the performance of a query given a static ranking system. This makes them inapplicable for our task as the rankings produced by ranking models are subject to significant variances during the training process. In fact, how to predict the performance of a learning-to-rank system in general is an underexplored topic in the research community.

In order to predict ranking rewards on unobserved rankings without online experiments, we need to construct metric surrogate models purely based on the data available in offline ranking optimization. Among different learning-to-rank applications, one type of information is universally available in all ranking optimization scenarios, that is the *local ranking context*. Local ranking context refers to structure and feature information of a ranking and the items in it. Given a reasonable ranking model, the local context of a ranking produced by this model often contains valuable information including session-specific feature distributions and inter-relations between items to rank. Previous studies on context-aware learning-to-rank [2, 5, 68] have shown that, by reading and utilizing the local context of an initial ranking, learning-to-rank models can better understand the meaning of relevance in current sessions and produce tailored rankings that better satisfies user's need. We believe that such information could be a key for offline metric modeling in metric-agnostic ranking optimization.

In this task, we propose to construct offline surrogate models for complex ranking metrics with local ranking context. Formally, given the training data $\mathcal{T} = \{\pi(I), m(\pi(I)) | I \in \mathcal{T}_\mathcal{I}, \pi(I) \in T_I\}$, our goal is to construct function g to predict $m(\pi(I))$ for any $\pi(I) \notin T_I$ with the local context of $\pi(I)$. Let $\vec{x}_i$ be a feature vector (pre-computed with hand-crafted algorithms or learned together with ranking optimization) of item $i \in I$ in the current session, then the local context of $\pi(I)$ can be represented as $X^g_{\pi(I)} = [\vec{x}_1, \vec{x}_2, ..., \vec{x}_{|\pi(I)|}]$.

Based on Eq. (1), the surrogate model g can be learned by minimizing the empirical loss as:

$$L(g, \mathcal{T}) = \sum_{I \in \mathcal{T}_I} \sum_{\pi(I) \in T_I} l\big(g(X^g_{\pi(I)}|\theta), m(\pi(I))\big) \qquad (5)$$

where θ is the model parameter of g. The key idea is to use the local context $X_{\pi(I)}$ as inputs to g so that, for any $\pi(I) \notin T_I$, we can predict its ranking reward without information from online experiments. Note that the design of g could be fairly flexible. It could be as simple as a linear regression or as complicated as a self-attention network [85]. Also, we can easily adapt the noise reduction framework proposed in Section 2.1 by jointly optimizing $g(X^g_{\pi(I)}|\theta)$ with a noise-observation model $o(X^o_{\pi(I)}|\theta_o)$ where $X^o_{\pi(I)}$ is a set of factors that potentially create noise in $m(\pi(I))$.

2.3 Offline Reinforcement Learning for Surrogate Metric Modeling

The most straightforward method to optimize the parameters of surrogate metric function g is to directly apply supervised learning algorithms using the loss functions proposed above. Yet, the generalizability of a supervised learning algorithm highly depends on the quality and quantity of training data. In practice, numerous training data may not be available due to the difficulty of collecting complex ranking metrics online. This jeopardizes the reliability and generaliability of metric modeling with supervised learning.

Fortunately, recent advances on reinforcement learning could provide some new insights on how to solve the data sparsity problem. Reinforcement learning is originally designed for ML applications where an agent policy is built to interact with an environment in order to obtain certain rewards pre-defined for the task. Similar to the problem of online experiments in IR, learning such agents by trial-and-error with real users is often detrimental to the system as it could significant affect user experience. Thus, an appealing solution is to construct agents purely based on logged transitions collected from the experiences of a previous policy that has already interacted with the environment for a while, which is referred to as *offline reinforcement learning*. Offline reinforcement learning has attracted considerable attention in the ML community [16, 93, 95]. Dadashi et al. [30] have conducted a theoretical analysis of offline reinforcement learning for pseudometric learning and show that it can help the agent achieve significantly better performance than traditional supervised learning methods. This gives us new inspirations on how to enhance model generaliability with limited data.

Therefore, we propose to adapt offline reinforcement learning algorithms to improve the generaliability of surrogate metric modeling. In general, the key to offline reinforcement learning is a distance function for action states/trajectories and a pseudometric defined over the distance function. The intuitive idea is that the reward of an unobserved state-action pair should be similar to an observed state-action pair in the logged transactions if their distance is small. Formally, let (s, a) be a state-action pair with observed reward $r(s, a)$. Let $\pi_\omega(s)$ be a policy parameterized by ω that takes the current state as input to predict the next action, and $Q_\theta(s, a)$ be the action value model parameterized by θ that tries to predict the true reward of (s, a). Assuming that there is a distance function $d(s_1, a_1; s_2, a_2)$ that measures the distance between two pairs (s_1, a_1) and (s_2, a_2), we define $d_\mathcal{T}(s, a) = \min_{(s', a') \in \mathcal{T}} d(s, a, s', a')$

as the distance function between (s, a) to observed data in $\mathcal{T}$. Then, following a popular actor-critic framework [30, 93], we can learn $\pi_\omega(s)$ and $Q_\theta(s, a)$ together by sampling transitions $(s, a, r, s') \in \mathcal{T}$ (where s' is the state after action a on s) and optimizing

$$\text{(critic)} \min_\theta ||Q_\theta(s, a) - r(s, a) - \alpha_r Q_\theta(s', \pi_\omega(s'))$$
$$- \alpha_c \mathcal{F}(d_\mathcal{T}(s', \pi_\omega(s')))||, \qquad (6)$$
$$\text{(actor)} \max_\omega Q_\theta(s, \pi_\omega(s)) + \alpha_a \mathcal{F}(d_\mathcal{T}(s, \pi_\omega(s)))$$

where α_r, α_c, and α_a are hyperparameters, and $\mathcal{F}(d_\mathcal{T}(s, a))$ is a pseudometric function defined over the distance between (s, a) and observed data in $\mathcal{T}$. Possible definition of $\mathcal{F}$ could be [30]

$$\mathcal{F}(d(s_1, a_1; s_2, a_2)) = |r(s_1, a_1) - r(s_2, a_2)| + \alpha_r \mathbb{E}_{a' \sim \mathcal{U}_\mathcal{A}} d(s'_1, a', s'_2, a') \qquad (7)$$

where s'_1 is the next state from s_1 after action a', and $\mathcal{U}_\mathcal{A}$ is a uniform distribution over action space $\mathcal{A}$.

Specifically for metric-agnostic ranking optimization, we can adapt the framework of offline reinforcement learning and define s as a specific ranking of items (i.e., $\pi(I)$) and a as an action of flipping the positions of two adjacent items in s. Further, we can replace $r(s, a)$ with $m(\pi(I), a) = m(\pi(I)')$ (where $\pi(I)'$ is the ranking after flipping action a on state $\pi(I)$) and define $d(\pi(I)_1, a_1; \pi(I)_2, a_2)$ as

$$d(\pi(I)_1, a_1, \pi(I)_2, a_2) = \text{Jaccard}_{flip}(\pi(I)'_1, \pi(I)'_2) \qquad (8)$$

where $\pi(I)'_1$ and $\pi(I)'_2$ is the rankings after action a_1 and a_2 on $\pi(I)_1$ and $\pi(I)_2$, respectively; and $\text{Jaccard}_{flip}(\pi(I)'_1, \pi(I)'_2)$ is the Jaccard distance between $(\pi(I)'_1, \pi(I)'_2)$ when the only actions are flipping adjacent items. With different sample and approximation techniques on pseudometric learning, we can directly construct $\mathcal{F}$ from the distance function above and learn a surrogate metric model $Q_\theta(s, a) = g(X^g_{\pi(I)'}|\theta)$ with the offline actor-critic algorithms described in Eq. (6). Such learning algorithms have potential to produce better surrogate metric model g as it can effectively regularize the prediction of g on unobserved rankings based on their similarities to observed $m(\pi(I))$. Also, depending on how we implement the action policy $\pi_\omega(s)$ (which is essentially a ranking model), we could improve the generaliability of g by letting it interact with different types of ranking models or even jointly optimize both the ranking model and the surrogate metric model together.

3 DIFFERENTIABLE RANKING WITH COARSE-GRAINED REWARD

Among all existing ML algorithms, gradient descent is one of the most popular methods that have been applied to a variety of parameter optimization problems, especially for deep learning [38]. Yet, unlike other tasks such as regression and classification, gradient descent is not directly applicable to ranking due to the discrete parameter space of ranking problems. For example, when we rank items by sorting them with ranking scores, a tiny perturbation on the score of an item wouldn't change the ranking and corresponding reward if it does not change the partial order between items. Thus, how to handle the discrete parameter space of ranking problems and make ranking models differentiable is a key research problem in learning-to-rank studies [56].

While there are many well-established optimization methods [17, 18, 20, 48] proposed for non-differentiable ranking problems, most of them are not applicable to model-agnostic ranking optimization.
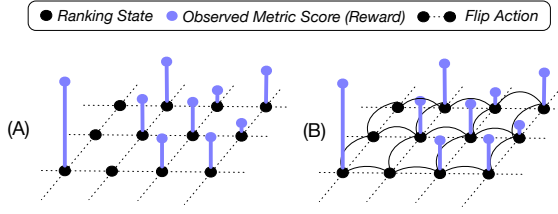
Figure 1: An illustration of latent ranking space.

In existing literature, there are two types of methods to compute gradient from discrete rankings. The first one, which is directly built on the PRP assumption, is to decompose a ranking as a set of partial order relations where we can treat each pair of items as a separate training example and train ranking models for partial order predictions instead [18, 48]. The second one is to interpret sorting with a probability framework so that we can model the likelihood of creating the best ranking given certain ranking metrics and simply optimize ranking models with maximum likelihood estimation [59, 94]. Despite their different motivations and formulations, all these methods assume the accessibility of fine-grained partial-order labels or information about the structure of ranking metrics so that they build training losses and apply gradient descent accordingly. Unfortunately, in metric-agnostic ranking optimization, whether the best rankings can be achieved by correctly predicting partial item orders is questionable. And, in many cases, we have neither fine-grained labels on item pairs nor knowledge on how the best rankings should look like in each ranking session.

In this section, we discuss the possibility of constructing differentiable ranking and parameter optimization algorithms for metric-agnostic ranking optimization. Specifically, we propose to start from three directions: (1) Differentialize discrete and deterministic ranking states with random item flips; (2) Smooth the distribution of ranking rewards with stochastic ranking paradigms built on different sampling strategies; (3) Separate item scoring from item ranking and propose ranking modules that adaptively decide how to present items based on score distribution and target metrics.

3.1 Differentializing Ranking States through Random Flips

In general, ranking metrics or rewards are non-differentiable mainly because of the "loose" connection between ranking model outputs and actual item rankings. For simplicity, assuming that rankings are created by sorting items based on their ranking scores (i.e., we only need a scoring function to rank items). Then, no matter how we perturb item's ranking scores, the ranking of items, so as the ranking reward, would remain unchanged as long as the partial orders of their ranking scores are unchanged. This makes it impossible to compute parameter gradients directly from the reward we observed on each ranking.

More intuitively, let each possible item ranking be a state node in a latent space (Figure 1(A)). Let's define a *flip* as swapping the positions of two items in a ranked list π_a, then the state π_a is connected to a state π_b if π_a is same to π_b after the flip. Because the changes of ranking scores do not necessarily trigger a flip, the hyperplane of ranking metrics is non-differentiable with respect to the parameters of scoring functions. Therefore, the key of differentiable ranking optimization is how to make flip actions sensitive to perturbations

on ranking scores. The most popular way to solve this problem in existing literature is to collect annotations on the partial orders of items so that we know how to flip items in advance and only need to train scoring functions to achieve it [18]. This method, however, is not applicable to metric-agnostic ranking optimization as the assumption that correct partial orders can guarantee best rankings (i.e., the PRP assumption) may not hold for complex ranking metrics, and we do not have access to partial order annotations on items in many cases. Thus, new methodology and algorithms are needed for joint modeling flips and ranking scores.

We believe that the solutions of this problem lie in the modeling flip actions in ranking space and differentialization of ranking rewards on observed rankings. Specifically, we propose to view flip actions as translation processes through noisy channels. Let π_{ij} be the ranking of items when swapping the ith and jth items in ranking π, and let $m(\pi_{ij})$ be the observed metric score on π_{ij}. We can define a noisy channel $\Gamma_{\pi_{ij}} = P_{\pi}(i|j)$ that translates the item on position j to i (i.e., swapping items on i and j). Then, for next translation on the ith item, the expected metric score $M(\pi_i)$ can be

$$M(\pi_i) = \sum_{j \in \pi; \pi, \pi_{ij} \in \mathcal{T}} \frac{\Gamma_{\pi_{ij}}}{\sum_{k \in \pi, \pi_{ik} \in \mathcal{T}} \Gamma_{\pi_{ik}}} \cdot m(\pi_{ij}) \qquad (9)$$

where $\mathcal{T}$ is the observed rankings in training data. Thus, given a set of observed rankings and corresponding metric scores, the goal of metric-agnostic ranking optimization is to construct a translation model that can translate any rankings into the ranking with highest reward. There have already been mature methods to solve such problems in markov random fields [61, 74], and we could easily adapt them to learn the noisy channel parameters. For instance, we can iteratively maximize the local expected value of $M(\pi_i)$ on all possible π_i and stop when $M(\pi_i)$ converges. To connect the noisy channel $\Gamma_{\pi_{ij}}$ with ranking models, we could apply different types of noise models parameterized by the ranking scores of items on i and j (i.e., s_i and s_j). For example, we could assume that $\Gamma_{\pi_{ij}}$ follows a Gaussian Distribution $\mathcal{N}(\mu, \sigma)$ where σ is a pre-defined variance and μ is computed as $(s_i - s_j) \cdot \text{sign}(i - j)$:

$$\Gamma_{\pi_{ij}} = P_{\pi}(i|j) = \int_{-\infty}^{0} \mathcal{N}(x; (s_i - s_j)\text{sign}(i - j), \sigma) dx \qquad (10)$$

so that we can directly optimize s_i by computing its gradient $\frac{\partial M(\pi_i)}{\partial s_i}$ through $\frac{\partial M(\pi_i)}{\partial \Gamma_{\pi_{ij}}}$ and $\frac{\partial \Gamma_{\pi_{ij}}}{\partial s_i}$. Note that the proposed noisy translation process is fundamentally different from existing studies that models ranking process as sequential item selection under probability frameworks [20] as the former directly optimize ranking models based on observed data while the later requires the access of explicit metric scores on all possible rankings in order to compute the global ranking losses. In other words, the proposed method is particularly suitable for metric-agnostic ranking optimization in which we have limited knowledge on ranking metric and can only collect reward scores on limited number of rankings.

3.2 Stochastic Ranking Paradigms

While ranking by sorting items with their ranking scores is a straightforward method, this paradigm oversimplifies ranking problems in practice and doesn't fit the needs of modern IR system today. For example, item or user exposure fairness is considered to be a key

problem for modern search and recommendation [9, 79]. To satisfy users or balance content in different ethnic groups, retrieval results need to be ranked according to the distribution of user demographics and available information. Such demands cannot be satisfied with rankings created by sorting because a deterministic ranking will inevitably create uneven exposure to items in it. Instead, recent studies on online learning to rank [65] and fairness [62] usually create multiple rankings for a set of items and sample based on pre-computed data distributions to balance the exploration and exposure of items. These methods provide a new perspective to view ranking problems and could potentially benefit our research on metric-agnostic ranking optimization.

The main difference between rankings in fairness studies and existing learning-to-rank algorithms is a random process that could produce non-deterministic rankings over the same set of items, which we refer to as the *stochastic ranking paradigm*. Intuitively, stochastic ranking paradigms do not guarantee that the final rankings shown to users would strictly follows the partial orders of items determined by their ranking scores. Different from the noisy channel modeling in Section 3.1, stochastic ranking paradigms directly model the marginal probability of a ranking following a pre-defined probabilistic ranking generation model so that we can compute the prior distribution of rankings and sample them directly. Thus, the key for ranking optimization with stochastic ranking paradigms is how to derive the prior distribution of ranking and how to compute or define loss functions on non-deterministic rankings.

Here, we propose to adapt stochastic ranking paradigms for metric-agnostic ranking optimization. Formally, let $s = f(x|\theta)$ be the ranking score of an item $x \in I$ and the scoring function (i.e., the ranking model) f is parameterized by θ. Assuming that the metric scores of all possible rankings (i.e., $\{m(\pi)|\pi \in \mathcal{R}(I)\}$) are available beforehand, we can directly conduct metric-agnostic ranking optimization by maximizing the expected metric reward as

$$\theta^* = \arg\max_{\theta} \sum_{\pi \in \mathcal{R}(I)} m(\pi) \cdot Pr(\pi|f, \theta) \tag{11}$$

where $Pr(\pi|f, \theta)$ is the prior probability of observing π given the ranking generation model parameterized by $f(x|\theta)$.

There are two key advantages of stochastic ranking comparing to traditional rank-by-sort paradigms. First, with the stochastic ranking process, we can compute the reward of a ranking policy without knowing the partial order labels on each item pairs. Second, by parameterizing the ranking sampling distribution with the scoring function $f(x|\theta)$, we are able to compute the gradient of expected ranking reward with respect to f and θ directly. Specifically in practice, while the cost of collecting $m(\pi)$ for all possible π is prohibitive, we can approximate $m(\pi)$ with the offline surrogate model $g(\pi)$ proposed in Section 2 so that we can estimate rewards on unobserved rankings without online experiments. Further, for the modeling of prior ranking distribution $Pr(\pi|f, \theta)$, we could apply and test different probabilistic models. For instance, we could adapt simple methods used in existing online learning-to-rank literature [59, 65, 83] (e.g., Plackett-luce distribution or multinomial logit), or try more advanced distributions such as CRS [77]. Also, instead of crafting $Pr(\pi|f, \theta)$ purely based on ranking scores $f(x|\theta)$, one can explore the possibility of incorporating rich model statistics such as scoring variance and output uncertainty in the construction of ranking models. Previous studies have shown that uncertainty in ranking scores can be effectively estimated through Bayesian network [35] and is helpful for data exploration in online learning to rank [26, 97]. We believe that such information could be useful for better modeling of prior ranking distribution and potentially benefit metric-agnostic ranking optimization.

3.3 Detached Modeling for Scoring and Ranking

To the best of our knowledge, most existing studies on learning to rank [56] does not explicitly differentiate the concept of *scoring* and *ranking*. Existing literature usually refers to a ranking model as a model that generate ranking scores for each item to rank (i.e., $f(x|\theta)$ in T2.1), and, in the actual ranking process, they directly sort items with the scores or sample rankings based on pre-defined probabilistic models over scores. For metric-agnostic ranking optimization, however, the integrated modeling of scoring and ranking has intrinsic disadvantages. For instance, user's sensitivity to result relevance often changes depending on their information needs and application scenarios. It is possible that items must be strictly ranked by their relevance in some sessions (e.g., to guarantee user's efficiency in information seeking) or could be loosely ranked by their relative novelty in all candidates (e.g., to encourage user's exploration on results). These diverse ranking needs cannot be satisfied with existing algorithms because they couple ranking with scoring and give almost no flexibility to the ranking process of items. When optimizing complex ranking metrics of which the internal structure could be unknown, such flexibility is important for building an effective ranking system.

To this end, we propose to detach the modeling of ranking with scoring in ranking problems. Specifically, we can split the construction of a ranking model into two parts: a *scoring module* that predicts item relevance based on their content and features; and a *ranking module* that generate result rankings based on the scores produced by the scoring module. Let $s = f(x|\theta)$ be the ranking score of an item x, then we define a ranking function $\mathcal{H}(\{s\}|\rho)$ (parameterized by ρ) that generates a ranking $\pi(I)$ as

$$\pi(I) \sim \mathcal{H}(\{f(x|\theta)|x \in I\}|\rho) \tag{12}$$

where $\mathcal{H}$ takes all ranking scores and context information as inputs to predict a ranking. This separated formulation of scoring and ranking module creates a generic framework for ranking optimization as existing ranking paradigms can be treated as special cases in it, e.g., the sort-by-score paradigm [56] can be viewed as a $\mathcal{H}$ implemented with a sorting algorithm; and the stochastic ranking paradigms with Plackett-luce [66] can be seen as a $\mathcal{H}$ built with a Markov decision process and softmax functions.

The proposed ranking framework allows us to create separated models for the ranking module and thus introduce significant flexibility in terms of model design. For instance, under a stochastic ranking framework, we can define $\mathcal{H}$ with a recurrent neural network and directly train the item sampling distribution based on the need of complex ranking metrics. Specifically, we plan to explore collaborative learning algorithms for the training of $f(x|\theta)$ and $\mathcal{H}(\{s\}|\rho)$. Given $\mathcal{H}(\{s\}|\rho)$, we can compute the prior probability of observing each $\pi(I)$ (i.e., $Pr(\pi|\mathcal{H}(\{f(x|\theta)|x \in I\}|\rho))$) and the expected metric reward with Eq. (11). Then, by iteratively fixing ρ

and θ, we can jointly optimize $f(x|\theta)$ and $\mathcal{H}(\{s\}|\rho)$ as

$$
\begin{aligned}
\theta_t &= \arg\max_{\theta} \sum_{\pi \in \mathcal{R}(I)} m(\pi) \cdot Pr(\pi|\mathcal{H}(\{f(x|\theta_{t-1})|x \in I\}|\rho_{t-1}) \\
\rho_t &= \arg\max_{\rho} \sum_{\pi \in \mathcal{R}(I)} m(\pi) \cdot Pr(\pi|\mathcal{H}(\{f(x|\theta_{t-1})|x \in I\}|\rho_{t-1})
\end{aligned}
\tag{13}
$$

where θ_t and ρ_t are parameters for f and $\mathcal{H}$ in time step t, respectively. Further, we could consider factors that are important for complex ranking metrics but independent with item relevance (e.g., user personality) in the construction of $\mathcal{H}$ to further improve the performance of the generated rankings. We believe that this detached modeling framework for scoring and ranking could have great potentials for metric-agnostic ranking optimization.

4 EFFICIENT RANKING EXPLORATION IN PARAMETER SPACE

Efficiency and convergence rate are important factors to analyze the quality of an optimization algorithm. It's particularly true for ranking optimization as the parameter space of ranking problems grows exponentially, if not faster, with respect to the number of items to rank. To address this problem, previous studies on ranking optimization often decompose ranking problems into a set of partial-order classification tasks under the PRP assumption. With fine-grained annotations on each partial-order pairs, this method can significantly reduce the parameter space of ranking problems from $O(n!)$ to $O(n^2)$ where n is the number of items. This makes it possible to optimize ranking models within polynomial time. In metric-agnostic ranking optimization, however, there isn't any trivial solution to reduce the parameter space of ranking problems. First, when we possess limited knowledge on the structure of ranking metrics and metric scores can only be obtained on list or session level, the cost of collecting training data would be high and fine-grained annotations on each item or item pairs are difficult or impossible to get. Second, due to their complicated nature, the reward hyper-surface of complex ranking metrics with respect to model parameters could have many local optima that hurt the efficiency and effectiveness of optimization algorithms.

In this section, we discuss the development of efficient learning algorithms for metric-agnostic ranking optimization. The goal is to reduce the parameter space or alleviate the need of data collection for complex ranking metric optimization and eventually make it possible to integrate the proposed metric-agnostic ranking optimization techniques into real IR systems.

4.1 Initialization with Multi-task Learning

Building large-scale models from scratches is considered to difficult in many applications such as computer vision and natural language process. This is also true for metric-agnostic ranking optimization considering the numerous number of possible rankings and giant parameter space. Previous studies [32] have found that pre-training models with large-scale unsupervised or weakly supervised data could significantly improve the efficiency and effectiveness of model optimization. A key observation is that, through pre-training, we could better initialize model parameters and narrow down the parameter space to explore in further optimization. Previous studies

on learning to rank [2, 3, 96] also show that good parameter initialization or good initial ranker can guarantee the convergence and performance of ranking models both theoretically and empirically.

Inspired by these observations, one potentially promising direction to pursue is the exploration of different parameter initialization and regularization techniques for efficient metric-agnostic ranking optimization. While the optimization of complex ranking metrics with unknown structures is still new to the IR community, there have already been many mature algorithms for how to construct training data and ranking models for simple metrics under PRP. Therefore, we could construct a multi-task learning framework that combines the traditional ranking optimization algorithms with the proposed methods for complex ranking metrics. Let $L_C(\mathcal{T}|\theta)$ be the loss function used by a metric-agnostic ranking optimization algorithm in Section 3 where $\mathcal{T}$ and θ are the training data and model parameters, respectively. Let $L_S(\mathcal{T}'|\theta)$ be the loss function of an ranking algorithm for simple metric optimization (e.g., pairwise cross entropy over human-annotated labels). Then, at time step t, we could update the model parameter θ as

$$
\theta_t = \theta_{t-1} + \alpha \Big(\phi(t, \theta_{t-1}) \frac{\partial L_C(\mathcal{T}|\theta)}{\partial \theta} + (1 - \phi(t, \theta_{t-1})) \frac{\partial L_S(\mathcal{T}'|\theta)}{\partial \theta} \Big) \tag{14}
$$

where α is the learning rate and $\phi(t, \theta_{t-1})$ is a function that takes the historic model parameters or gradients to predict the current weights for different ranking losses. The learning framework described in Eq. (14) can be viewed as a multi-task learning algorithm that balance parameter optimization for two different targets. The rational behind this is that, while the simple loss function $L_S(\mathcal{T}'|\theta)$ may not be perfect for complex ranking metrics, it should be able to produce a reasonably good starting point. By combining it with $L_C(\mathcal{T}|\theta)$, we allow the model to quick initialize parameters based on simple losses built on a separate or the same training data $\mathcal{T}'$ (with fine-grained labels) and thus increase the overall convergence rate of the optimization algorithms. Here, the weighting function $\phi(t, \theta_{t-1})$ is introduced to balance the parameter initialization process with the final objective in metric-agnostic ranking optimization. It should give relatively high weights to $L_S(\mathcal{T}'|\theta)$ in the beginning and gradually move to $L_C(\mathcal{T}|\theta)$ when we have more confidence on the model parameters. This could be achieved in multiple way such as implementing $\phi(t, \theta_{t-1})$ with a simulated annealing algorithm [84] or a parameter-free convex learning algorithm (e.g. the coin betting algorithm using Krichevsky-Trofimov Estimator [67]). Also, $L_S(\mathcal{T}'|\theta)$ is not necessarily a single loss function and could be a combination of multiple loss functions built with different types of weak supervision signals [31, 33].

4.2 Exploration with Uncertainty Modeling

Generally, large-scale training data are important, if not essential, for the generalizability and reliability of learning-to-rank models. Unfortunately, in metric-agnostic ranking optimization, the scores of complex ranking metrics have to be collected online and it's often prohibitive to get data that are large enough to cover multiple rankings in each session. Therefore, how to collect training data wisely is particularly important for the efficiency and convergence of ranking optimization algorithms.

In information theory [13], information can be viewed as a resolution of uncertainty. This indicates that an informative training example should be the data samples that can significantly reduce
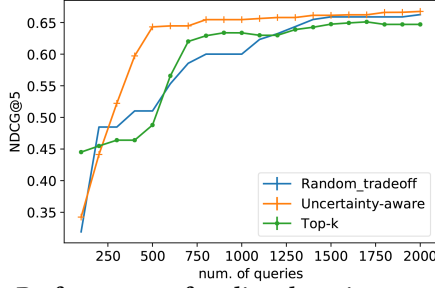
Figure 2: Performance of online learning to rank algorithms.

Figure 3: An illustration of state reorganization.

uncertainty in model outputs [78]. Model uncertainty (or variance) has been extensively studied in explainable AI [10] and accountable AI [21]. Specifically in ranking optimization, however, it's potential hasn't been fully explored in existing literature. Recently, studies [26, 97] on online learning to rank has shown that model uncertainty estimation could be helpful for identifying informative data for efficient training and model optimization. To better illustrate this, we followed the experiment setup on MSLR-30K[1] used by Yang et al. [97] and compared the ranking functions learned with (1) Top-k: simply sort and show candidate results based on their relevance scores predicted by the current ranking model; (2) Random: randomly sort and present candidate results; (3) Uncertainty-aware: sort and show results based on their relevance scores plus the uncertainty (estimated with random dropout [35]) of current model outputs. As shown in Figure 2, by selecting and presenting items based on their ranking score uncertainty (i.e., the orange line), we can collect training data that significantly increase the convergence speed and ranking performance of a ranking algorithm. This motivates us to further investigate the potential of uncertainty modeling for efficient metric-agnostic ranking optimization.

Based on these observations, we propose to conduct uncertainty estimation for models in metric-agnostic ranking optimization and use the estimated uncertainty information to improve the efficiency of training data collection. Our idea is to collect ground truth data on cases where the current ranking model or surrogate metric model has the highest uncertainty so that algorithms can converge quickly with less training data. Formally, let the uncertainty/variance of a surrogate metric model $g(\pi)$ over ranking π be δ_π, and the uncertainty/variance of the probability to observe π given the ranking model (i.e., $Pr(\pi|f, \theta)$ in Eq. (11) or $\mathcal{H}(\{f(x|\theta)|x \in I\}|\rho)$ in Eq. (12)) be ϵ_{pi}. For simplicity, we assume that $g(\pi)$ and $Pr(\pi)$ are independent variables following normal distribution $\mathcal{N}(g(\pi), \delta_\pi^2)$ and $\mathcal{N}(Pr(\pi), \epsilon_\pi^2)$. Then the uncertainty of expected ranking reward in Eq. (11) or (13), i.e., $\Delta(I)$, can be computed as

$$\Delta(I)^2 = \sum_{\pi \in \mathcal{R}(I)} Var(g(\pi) \cdot Pr(\pi))^2$$
$$= \sum_{\pi \in \mathcal{R}(I)} (g(\pi)^2 \epsilon_\pi^2 + Pr(\pi)^2 \delta_\pi^2 + \epsilon_\pi^2 \delta_\pi^2) \quad (15)$$

where $\mathcal{R}(I)$ is all possible rankings in session item set I. Based on the theory of active learning [27], annotating data with highest model uncertainty could sharply decrease the number of training examples the learner needs in order to achieve a good performance.
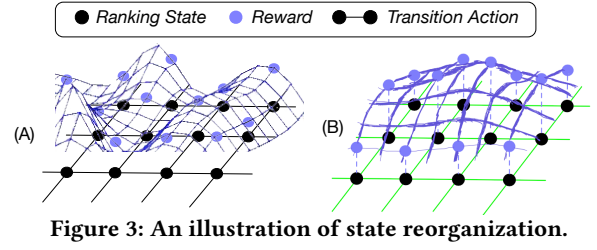
[1]https://www.microsoft.com/en-us/research/project/mslr/

Therefore, to conduct efficient optimization, we can actively select ranking session I based on $\Delta(I)$ and select specific ranking $\pi \in \mathcal{R}(I)$ based on $Var(g(\pi) \cdot Pr(\pi))$ to collect ground truth metric score $m(\pi)$ in online experiments. Finally, for the computation of δ_π and ϵ_π, we could directly utilize variational inference [41] and Monte Carlo dropout [35, 53, 63] to infer them from the current surrogate metric models and ranking models.

4.3 Hypersurface and State Reorganization

Existing ML theories and techniques are well-developed for convex optimization problems. Particularly, gradient descent based optimizers [14, 72] have been shown to be effective in finding global optima for differentiable convex loss or reward functions. When the loss or reward distribution is non-convex, however, there is no guarantee that those methods could produce good results and gradient descent could easily be stuck by local minimums. Non-convex reward hypersurface is often an important problem for complex ranking metrics and it could jeopardize the efficiency and effectiveness of optimization algorithms.

A key reason behind the non-convex reward hypersurface in metric-agnostic ranking optimization is the fact that, for complex ranking metrics, the swap of two adjacent items does not guarantee a consistent gradient direction for the ranking reward. Previously, under the PRP assumption, the change of relative positions for two items would always lead to a better or worse reward in terms of ranking metrics [71]. Thus, ranking losses built for these metrics are mostly convex and algorithms that directly optimize the partial orders of items could converge quickly while achieving reasonably good performance [56]. In contrast, complex ranking metrics have more complicated structures and could be affected by factors which may not be quantified or known in advance. For example, user engagement in news recommendation is significantly affected by item novelty. The swap of two items could lead to both increase and decrease of user engagement depending on the context. Without proper solutions, these phenomena naturally create non-convex reward hypersurface that are difficult to optimize.

Therefore, to solve the problem of non-convex reward hypersurface, we want to explore the possibility of refining the reward hypersurface of complex ranking metrics for better optimization efficiency and effectiveness. We may achieve this through reward analysis and ranking state reorganization. Similar to the ranking state formulation in Section 3.1, suppose that we have already created a differentiable reward function with techniques proposed in Section 3, which give us a reward hypersurface over ranking states such as Figure 3(A). When we connect ranking states naively through item flips (as discussed in Section 3.1), the reward hypersurface of complex ranking metrics could be non-convex with respect to the changes of ranking states. To solve this problem, one needs

to conduct detailed analysis on the ranking rewards and group ranking states with similar properties together. For example, we could conduct data clustering with both the reward scores and the local context features of each ranking, assuming that data within the same cluster would have more consistent characteristics with respect to parameter perturbations on ranking models. Based on the result of reward analysis, we could develop new ranking paradigms that create a new set of transaction actions that connect states with similar properties intra and inter clusters to build a reward hypersurface with better convexity.

The development of such ranking paradigms is directly related to the design of ranking module in Section 3.3. For example, instead of connecting ranking states with random item flips, we could build a ranking module that selects both positions and items with probabilistic models. We could optimize the sampling distributions together with the surrogate reward model so that ranking states with similar predicted rewards would be neighbors under the new ranking paradigm. Intuitively, this means reorganizing ranking states through a new set of transaction actions so that we could create a reward hypersurface with less local optima, as shown in Figure 3(B). Though this is a difficult task with non-trivial solutions, we believe that any progress on this direction could significantly contribute to the studies of complex ranking metric optimization and potentially change the future of ranking optimization research.

5 RELATED WORK

Learning to Rank and Ranking Optimization. Learning to rank (LTR) refers to the techniques of adapting ML algorithms to build ranking models for IR problems [56]. The basic idea of LTR is to build feature representations for each item and design a scoring function that takes the features to predict the ranking score of each item [56]. Traditional LTR models usually assume the independence of item relevance [71] and score each item purely based on its individual feature vector [56]. Recently, more advanced LTR techniques acknowledge the importance of contextual information and create multi-variate scoring functions that jointly score multiple items together with all their feature vectors [2, 5, 46, 68, 69]. To optimize the parameters in ranking models, classic LTR methods design different loss functions based on hand-crafted ranking metrics such as NDCG [92]. Depending how many items are concerned in each loss function, LTR loss functions can be broadly categorized into pointwise [37], pairwise [18, 48], and listwise[17, 20] methods. Despite their differences, existing loss functions require explicit knowledge on the ranking metrics and fine-grained labels on each item [56], which is often unavailable in metric-agnostic ranking optimization.

Ranking Evaluation and Performance Metrics. Ranking evaluation is critical for IR as it essentially determines the direction of system optimization [25, 87]. Classic ranking evaluation studies mostly focus on evaluating IR systems based on their ability to retrieve relevant documents. For instance, a variety of ranking metrics are designed based on the relevance of each item [75] and use different heuristics to model the importance of different ranking positions [22, 44, 51]. In late 1990s, ranking evaluation becomes more complicated when the objective of IR moves from retrieving topical relevant documents to documents that satisfies the individual needs of search engine or recommendation users [86]. For example, document's relevance to users depending on multiple factors such as authority, timeliness, novelty, etc., many of which cannot be captured by the topical relevance between queries and documents [60]. Thus, modern ranking evaluation on real IR systems often rely on user-centric and session-level metrics such as user engagements, user satisfactions, etc. [28, 57, 64]. These metrics usually suffer from noisy user behaviors and cannot be explicitly formulated based on individual relevance labels. How to optimize these metrics directly is mostly unknown in existing literature.

Unbiased and Online Learning to Rank. Studies on unbiased and online LTR can be viewed as some initial attempts to optimize ranking with noisy metrics. In general, user interactions with IR systems often implicitly reflect their opinions and satisfaction on result rankings [39]. Therefore, modern IR systems often use user behavior data (e.g., clicks) as relevance signals to improve ranking systems [47]. As user behaviors often contain significant noise and collecting user interactions is expensive and time-consuming, studies on unbiased LTR and online LTR focus on how to effectively remove noise and bias from user data and efficiently train LTR models with less user interactions [4]. Examples include the applications of counterfactual learning [1, 3, 49, 91], variance reductions techniques in bandit learning [42, 76, 88, 89], etc.. Nonetheless, existing studies on unbiased and online LTR mostly focus on how to interpret item-level user behavior signals and collect relevance feedback on each result [6]. From this perspective, they are similar to existing LTR methods using fine-grained relevance annotations.

Optimization Algorithms and ML Paradigms. In last decades, we have observed significant improvements in the design of optimization algorithms in ML [12, 43, 80]. Depending on the format of training data, existing ML methods can be broadly categorized as supervised, semi-supervised, unsupervised algorithms. Also, depending on the format of supervision signals, a variety of learning algorithms have been proposed, included reinforcement learning [50, 82], few-shot learning [36, 70, 81], multi-task learning [34, 73], adversarial learning [58, 90, 100], etc.. Despite the different formulations used by these algorithms, most of them optimize model parameters following the classic gradient descent framework by iteratively updating models with their current gradients with respect to the learning targets [11].

6 CONCLUSION

In this paper, we discuss the challenges and opportunities of metric-agnostic ranking optimization, and propose three research tasks, i.e., structure-agnostic metric modeling and simulation, differentiable ranking process with coarse-grained reward, and efficient ranking exploration in parameter space. Given that ranking has served as a critical component of most IR systems and IR applications have become the most important tools for people to access information online, the development of effective techniques to support better ranking models and optimization algorithms will be significant. Besides, studies on metric-agnostic ranking optimization could have great value from a broader perspective of AI technology as advanced ranking optimization techniques could lead to a higher level of machine intelligence from passive learning to active learning in real-world environments. Through this paper, we hope to encourage more researchers to look into the problem and develop new ranking paradigms and optimization algorithms in future.

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*. 4–14.

[2] Qingyao Ai, Keping Bi, Jiafeng Guo, and W Bruce Croft. 2018. Learning a deep listwise context model for ranking refinement. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 135–144.

[3] 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.

[4] Qingyao Ai, Jiaxin Mao, Yiqun Liu, and W Bruce Croft. 2018. Unbiased learning to rank: Theory and practice. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2305–2306.

[5] Qingyao Ai, Xuanhui Wang, Sebastian Bruch, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2019. Learning Groupwise Multivariate Scoring Functions Using Deep Neural Networks. In *Proceedings of the 2019 ACM international conference on the theory of information retrieval*. ACM.

[6] 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.

[7] Azzah Al-Maskari, Mark Sanderson, and Paul Clough. 2007. The Relationship between IR Effectiveness Measures and User Satisfaction. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Amsterdam, The Netherlands) *(SIGIR '07)*. Association for Computing Machinery, New York, NY, USA, 773–774. https://doi.org/10.1145/1277741.1277902

[8] Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles LA Clarke. 2021. Shallow pooling for sparse labels. *arXiv preprint arXiv:2109.00062* (2021).

[9] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *The 41st international acm sigir conference on research & development in information retrieval*. 405–414.

[10] Szymon Bobek and Grzegorz J Nalepa. 2021. Introducing Uncertainty into Explainable AI Methods. In *International Conference on Computational Science*. Springer, 444–457.

[11] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer, 177–186.

[12] Léon Bottou, Frank E Curtis, and Jorge Nocedal. 2018. Optimization methods for large-scale machine learning. *Siam Review* 60, 2 (2018), 223–311.

[13] Leon Brillouin. 2013. *Science and information theory*. Courier Corporation.

[14] Sébastien Bubeck. 2017. Convex Optimization: Algorithms and Complexity. *Foundations and Trends in Machine Learning* 8 (2017).

[15] Chris Buckley, Darrin Dimmick, Ian Soboroff, and Ellen Voorhees. 2007. Bias and the limits of pooling for large collections. *Information retrieval* 10, 6 (2007), 491–508.

[16] Jacob Buckman, Carles Gelada, and Marc G Bellemare. 2020. The importance of pessimism in fixed-dataset policy optimization. *arXiv preprint arXiv:2009.06799* (2020).

[17] Christopher Burges, Robert Ragno, and Quoc Le. 2006. Learning to rank with nonsmooth cost functions. *Advances in neural information processing systems* 19 (2006), 193–200.

[18] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*. 89–96.

[19] Yinqiong Cai, Jiafeng Guo, Yixing Fan, Qingyao Ai, Ruqing Zhang, and Xueqi Cheng. 2022. Hard Negatives or False Negatives: Correcting Pooling Bias in Training Neural Ranking Models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 118–127.

[20] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*. 129–136.

[21] Paula Carvalho, Danielle Caled, Mário J Silva, Bruno Martins, João Paulo Carvalho, Joaquim Carreira, João Pedro Fonseca, Teresa Gomes, and Pedro Camacho. 2021. Assessing News Credibility: Misinformation Content Indicators. (2021).

[22] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 621–630.

[23] Ye Chen, Yiqun Liu, Min Zhang, and Shaoping Ma. 2017. User satisfaction prediction with mouse movement information in heterogeneous search environment. *IEEE Transactions on Knowledge and Data Engineering* 29, 11 (2017), 2470–2483.

[24] Aleksandr Chuklin, Pavel Serdyukov, and Maarten De Rijke. 2013. Click model-based information retrieval metrics. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 493–502.

[25] Cyril Cleverdon. 1967. The Cranfield tests on index language devices. In *Aslib proceedings*. MCB UP Ltd.

[26] Daniel Cohen, Bhaskar Mitra, Oleg Lesota, Navid Rekabsaz, and Carsten Eickhoff. 2021. Not all relevance scores are equal: Efficient uncertainty and calibration modeling for deep retrieval models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 654–664.

[27] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active learning with statistical models. *Journal of artificial intelligence research* 4 (1996), 129–145.

[28] Michael Cole, Jingjing Liu, Nicholas Belkin, Ralf Bierig, Jacek Gwizdka, C Liu, Jin Zhang, and X Zhang. 2009. Usefulness as the criterion for evaluation of interactive information retrieval. *Proc. HCIR* (2009), 1–4.

[29] 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*. 87–94.

[30] Robert Dadashi, Shideh Rezaeifar, Nino Vieillard, Léonard Hussenot, Olivier Pietquin, and Matthieu Geist. 2021. Offline Reinforcement Learning with Pseudometric Learning. *arXiv preprint arXiv:2103.01948* (2021).

[31] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 65–74.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*.

[33] Nadav Eiron and Kevin S McCurley. 2003. Analysis of anchor text for web search. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. 459–460.

[34] Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 109–117.

[35] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. PMLR, 1050–1059.

[36] Victor Garcia and Joan Bruna. 2017. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043* (2017).

[37] Fredric C Gey. 1994. Inferring probability of relevance using the method of logistic regression. In *SIGIR'94*. Springer, 222–231.

[38] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[39] Ahmed Hassan, Rosie Jones, and Kristina Lisa Klinkner. 2010. Beyond DCG: user behavior as a predictor of a successful search. In *Proceedings of the third ACM international conference on Web search and data mining*. 221–230.

[40] Claudia Hauff, Leif Azzopardi, and Djoerd Hiemstra. 2009. The combination and evaluation of query performance prediction methods. In *European Conference on Information Retrieval*. Springer, 301–312.

[41] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. *Journal of Machine Learning Research* 14, 5 (2013).

[42] Rolf Jagerman, Harrie Oosterhuis, and Maarten de Rijke. 2019. To model or to intervene: A comparison of counterfactual and online learning to rank from user interactions. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 15–24.

[43] Prateek Jain and Purushottam Kar. 2017. Non-convex optimization for machine learning. *arXiv preprint arXiv:1712.07897* (2017).

[44] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[45] Jiepu Jiang and James Allan. 2016. Correlation Between System and User Metrics in a Session. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval* (Carrboro, North Carolina, USA) *(CHIIR '16)*. Association for Computing Machinery, New York, NY, USA, 285–288. https://doi.org/10.1145/2854946.2855005

[46] Ray Jiang, Sven Gowal, Timothy A Mann, and Danilo J Rezende. 2018. Beyond greedy ranking: Slate optimization via list-CVAE. *arXiv preprint arXiv:1803.01682* (2018).

[47] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 133–142.

[48] Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 217–226.

[49] 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.

[50] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research* 4 (1996), 237–285.

[51] Diane Kelly, Xin Fu, and Chirag Shah. 2007. Effects of rank and precision of search results on users' evaluations of system performance. *University of North Carolina* (2007).

[52] Hitesh Khandelwal, Viet Ha-Thuc, Avishek Dutta, Yining Lu, Nan Du, Zhihao Li, and Qi Hu. 2021. Jointly Optimize Capacity, Latency and Engagement in Large-scale Recommendation Systems. In *Fifteenth ACM Conference on Recommender Systems*. 559–561.

[53] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. 2020. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International Conference on Machine Learning*. PMLR, 5436–5446.

[54] Dmitry Lagun, Chih-Hung Hsieh, Dale Webster, and Vidhya Navalpakkam. 2014. Towards better measurement of attention and satisfaction in mobile search. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 113–122.

[55] Mengyang Liu, Yiqun Liu, Jiaxin Mao, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. " Satisfaction with Failure" or" Unsatisfied Success" Investigating the Relationship between Search Success and User Satisfaction. In *Proceedings of the 2018 World Wide Web Conference*. 1533–1542.

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

[57] Yiqun Liu, Ye Chen, Jinhui Tang, Jiashen Sun, Min Zhang, Shaoping Ma, and Xuan Zhu. 2015. Different users, different opinions: Predicting search satisfaction with mouse movement information. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 493–502.

[58] Daniel Lowd and Christopher Meek. 2005. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 641–647.

[59] Jiaqi Ma, Xinyang Yi, Weijing Tang, Zhe Zhao, Lichan Hong, Ed Chi, and Qiaozhu Mei. 2021. Learning-to-Rank with Partitioned Preference: Fast Estimation for the Plackett-Luce Model. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 928–936.

[60] Jiaxin Mao, Yiqun Liu, Ke Zhou, Jian-Yun Nie, Jingtao Song, Min Zhang, Shaoping Ma, Jiashen Sun, and Hengliang Luo. 2016. When does relevance mean usefulness and user satisfaction in web search?. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 463–472.

[61] Donald Metzler and W Bruce Croft. 2005. A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. 472–479.

[62] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling fairness and bias in dynamic learning-to-rank. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 429–438.

[63] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Icml*.

[64] Daan Odijk, Ryen W White, Ahmed Hassan Awadallah, and Susan T Dumais. 2015. Struggling and success in web search. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1551–1560.

[65] Harrie Oosterhuis. 2021. Computationally Efficient Optimization of Plackett-Luce Ranking Models for Relevance and Fairness. *arXiv preprint arXiv:2105.00855* (2021).

[66] 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*. 1293–1302.

[67] Francesco Orabona and Dávid Pál. 2016. Parameter-Free Convex Learning through Coin Betting. In *Workshop on Automatic Machine Learning*. PMLR, 75–82.

[68] Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. 2020. Setrank: Learning a permutation-invariant ranking model for information retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 499–508.

[69] Rama Kumar Pasumarthi, Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2020. Permutation equivariant document interaction network for neural learning to rank. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 145–148.

[70] Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning. (2016).

[71] Stephen E Robertson and K Sparck Jones. 1976. Relevance weighting of search terms. *Journal of the American Society for Information science* 27, 3 (1976), 129–146.

[72] Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747* (2016).

[73] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

[74] Havard Rue and Leonhard Held. 2005. *Gaussian Markov random fields: theory and applications*. CRC press.

[75] Tefko Saracevic. 1996. Relevance reconsidered. In *Proceedings of the second conference on conceptions of library and information science (CoLIS 2)*. ACM New York, 201–218.

[76] Anne Schuth, Harrie Oosterhuis, Shimon Whiteson, and Maarten de Rijke. 2016. Multileave gradient descent for fast online learning to rank. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. 457–466.

[77] Arjun Seshadri, Stephen Ragain, and Johan Ugander. 2020. Learning Rich Rankings. *Advances in Neural Information Processing Systems* (2020).

[78] Burr Settles. 2009. Active learning literature survey. (2009).

[79] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2219–2228.

[80] Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. 2012. *Optimization for machine learning*. Mit Press.

[81] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1199–1208.

[82] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[83] Maksim Tkachenko and Hady W Lauw. 2016. Plackett-luce regression mixture model for heterogeneous rankings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 237–246.

[84] Peter JM Van Laarhoven and Emile HL Aarts. 1987. Simulated annealing. In *Simulated annealing: Theory and applications*. Springer, 7–15.

[85] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[86] Suzan Verberne, Maarten van der Heijden, Max Hinne, Maya Sappelli, Saskia Koldijk, Eduard Hoenkamp, and Wessel Kraaij. 2013. Reliability and validity of query intent assessments. *Journal of the American Society for Information Science and Technology* 64, 11 (2013), 2224–2237.

[87] Ellen M Voorhees. 2001. The philosophy of information retrieval evaluation. In *Workshop of the cross-language evaluation forum for european languages*. Springer, 355–370.

[88] Huazheng Wang, Sonwoo Kim, Eric McCord-Snook, Qingyun Wu, and Hongning Wang. 2019. Variance reduction in gradient exploration for online learning to rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 835–844.

[89] Huazheng Wang, Ramsey Langley, Sonwoo Kim, Eric McCord-Snook, and Hongning Wang. 2018. Efficient exploration of gradient space for online learning to rank. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 145–154.

[90] Kunfeng Wang, Chao Gou, Yanjie Duan, Yilun Lin, Xinhu Zheng, and Fei-Yue Wang. 2017. Generative adversarial networks: introduction and outlook. *IEEE/CAA Journal of Automatica Sinica* 4, 4 (2017), 588–598.

[91] 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*. 115–124.

[92] Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2018. The lambdaloss framework for ranking metric optimization. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1313–1322.

[93] Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. 2021. Uncertainty Weighted Actor-Critic for Offline Reinforcement Learning. *arXiv preprint arXiv:2105.08140* (2021).

[94] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*. 1192–1199.

[95] Teng Xiao and Donglin Wang. 2021. A general offline reinforcement learning framework for interactive recommendation. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*.

[96] Tao Yang, Shikai Fang, Shibo Li, Yulan Wang, and Qingyao Ai. 2020. Analysis of multivariate scoring functions for automatic unbiased learning to rank. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2277–2280.

[97] Tao Yang, Chen Luo, Hanqing Lu, Parth Gupta, Bing Yin, and Qingyao Ai. 2022. Can clicks be both labels and features? Unbiased Behavior Feature Collection and Uncertainty-aware Learning to Rank. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 6–17.

[98] Emine Yilmaz, Manisha Verma, Nick Craswell, Filip Radlinski, and Peter Bailey. 2014. Relevance and effort: An analysis of document utility. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 91–100.

[99] 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*. 1011–1018.

[100] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM*

Conference on AI, Ethics, and Society. 335–340.

[101] Ying Zhao, Falk Scholer, and Yohannes Tsegay. 2008. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *European conference on information retrieval.* Springer, 52–64.

[102] Yifei Zhao, Yu-Hang Zhou, Mingdong Ou, Huan Xu, and Nan Li. 2020. Maximizing Cumulative User Engagement in Sequential Recommendation: An Online

Optimization Perspective. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2784–2792.

[103] Yun Zhou and W Bruce Croft. 2007. Query performance prediction in web search environments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval.* 543–550.