Toward Reliable Decision-making in Information Systems

A

Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

Yiling Jia

May 2022

APPROVAL SHEET

This

Dissertation

is submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Author: Yiling Jia

This Dissertation has been read and approved by the examing committee:

Advisor: Hongning Wang

Advisor:

Committee Member: David Evans

Committee Member: Haifeng Xu

Committee Member: Cong Shen

Committee Member: Liangjie Hong

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

May 2022

Abstract

Nowadays information systems have been increasingly used in assisting our everyday decision-makings, from recommending movies to watch, products to purchase, to even providing treatment to patients. Such systems enable us to have every kind of data we possibly want at our fingertips. on the contrary, information system become more involved in making crucial decisions affecting human livelihoods, e.g., clinical decision-making criminal's incarceration. It brings up serious concerns on whether the systems could provide reliable service, and urges us to improve the reliability of the information systems.

The reliability of the decision-making can be evaluated in various aspects. First, accuracy undoubtedly is the foundation of a reliable information system. Users' trust on the system will be hurt if they are presented with instances that are subsequently found to be inferior. Besides, only returning the most relevance results to users is insufficient to help users to perceive the value of the provided information. To gain users' trust, the system also needs to be more transparent and make it more explicit why the users should pay attention to those returned results. In addition, served as the inter-media between information provider and consumer, modern information systems need to be trustful for both the consumer and the information provider. For the candidate instances, a reliable decision-making should not create discriminatory or unjust impacts when comparing across different demographics.

In this thesis, we focus on three aspects of the reliability of the decision-making in information system, *accuracy*, *fairness*, and *transparency*,. More specifically, we present three main tasks to enhance the modern information systems: 1) efficient online learning to rank; 2) fair online learning to rank; and 3) explainable recommendation. Our study provides a deep and thorough understanding of the importance of the reliability in information systems, and improves the reliability of the system by providing more effective, fair and transparent service. Rigorous theoretical analysis and extensive empirical evaluation validated the approaches' applicability in various contexts and applications.

Acknowledgements

I did not expect that I would study for a Ph.D. in US. This was a very important experience for me, and it changed the path of my life. However, it would never been possible for me to finish this work without the support and guidance that I have received from many people.

First of all, I would express my sincere thanks to my advisor, Prof. Hongning Wang. The most fortunate thing through my Ph.D. study is to have him as my advisor. His support is always there when needed and without fail he has always gone above and beyond. Prof. Wang has been an inspiration to me for his passion for research, insightful and critical thinking, rigorous research attitude. I am extremely lucky to be advised by him during my Ph.D. study.

I would like to thank my dissertation committee members Prof. David Evans, Prof. Haifeng Xu, Prof. Cong Shen, and Dr. Liangjie Hong for their invaluable suggestions on my Ph.D. research and dissertation. Prof. David Evans's comments on my research always inspire me. He encourages me to consider the impact of machine learning, or broadly say artificial intelligence, on the society. Discussing with Prof. Haifeng Xu and Prof. Cong Shen is always inspiring. Their critical thinking and rigorous attribute to research have been influencing me. To Dr. Liangjie Hong, for kindly sharing his suggestions from the practical perspective, which guides me to think more about the ethical concerns of my research.

I would like to thank all my collaborators during my Ph.D. study. Prof. Kamin Whitehouse is one of the most influential persons who showed me what a successful and rigorous researcher is like. Dr. Nipun Batra has been acting as my research mentor when I started my Ph.D. study. He made me understand the importance of self-disciplined life and rigorous experimental guidelines for the life of a graduate student. Prof. Quanquan Gu is always inspiring to me, and I learn a lot whenever talking with him. I would also thank my mentor Dr. Stephen Guo for his help and guidance during my internship in Walmart Labs, which was a very valuable experience for me. Many thanks to HCDM lab members for their kind help and support, who provide me a pleasant environment to study here.

I would like to thank my family, my parents, my grandmothers. Although they are far away in China, they always support me with their love and prayers and cheer me on. 最后，感谢我的父母，奶奶，外公和外婆。读博这六年跟你们在一起的时间加起来不到两个月，感谢你们的理解和支持，让我可以安心在国外读博。 Tutu, you were the light of my life when I was struggling in life. Hope you have a happy life in dog planet, running and playing everyday.

Finally, I would really thank my husband, Congchao. Because of you, I quit my job in China and came to US to pursue the Ph.D. degree. Without you, it would have been hard for me to reach this point. You always support me, encourage me and have confidence in me. No matter how frustrated I am, you are always able to make me laugh. I really feel lucky and full of joy to meet you, know you, and be with you for the past ten years.

May I continue to strive to live up to my dreams in the years to come.

To my family, and my dearest Tutu

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Overview

Information system serves as the predominant interface between users and massive amount of information. Enhanced by advanced machine learning algorithms, such systems are being adopted in a growing number of contexts and playing a central role in modern life, including health and wellness, entertainment, finance, and commerce [1]. With the information overload caused by the explosive growth of information, information systems bring myriad benefits. For example, recommendation systems help to filter, prioritize and highlight the relevant or valuable instance from a vast catalog; intelligent online recruiting platforms enable the matching between the applicants and the job requirements. On the contrary, information systems increasingly make crucial decisions affecting human livelihoods, for example, to decide bank loans, clinical decision-making, and the criminals' incarceration. A minor mistake in the information systems has the potential for significant negative impact. This makes a reliable decision-making in modern information system urgent and crucial.

The reliability of an information system can be evaluated in various aspects. First, beyond any doubt, accuracy is the foundation of a reliable information system. For example, users' trust on the recommendation system will be hurt if they are recommended with instances that are subsequently found to be inferior. A reliable information system should be able to provide right information to the users based on their information need to gain the trust. Besides, an accurate prediction alone is not sufficient to gain users' trust. Equipped with advanced machine learning techniques, existing modern information systems are more like a black-box: computerized oracles give advice, but cannot be questioned. Previous research shows that transparent information system, e.g., explaining how/why the results are presented to a user, help people make more accurate decisions [2], improve user acceptance [3], and increase trust in the information systems [4]. For example, in recommendation systems, in addition to "which one to buy", the users also seek for the reason of the recommendation, e.g., "why should I buy it". Last but not least, serving as the intermediary that provides the service to both consumers (users) and information providers (content providers), the information system is responsible to make reliable decisions for both sides. A reliable decision-making should not create discriminatory or unjust impacts when comparing across different demographics of the information providers. For example, in a job marketplaces, fail to consider the fairness of the decisions will lead to unequal treatment of a candidate based on some special properties such as gender and sex, which may limit the group's visibility and thus future engagement.

Many challenges are introduced in improving the reliability of the decision-making in information systems. First, in modern information systems, most of the models are learned with user implicit feedback, instead of the expert annotations. Such implicit feedback is not guaranteed to be reliable, e.g., click feedback is noisy and biased. Efficiently learn an effective information model with unreliable data is one of the key challenges. In the meantime, reliable information systems are also required to fullfill multiple objectives along with the accurate prediction, e.g., ensuring transparency and fairness. The system needs to trade off the multiple objectives when optimizing the serving performance to reach the desired outcome. To make things even worse, those objectives oftentimes can be conflicting, e.g., relevance vs. fairness. To tackle the aforementioned challenges, and improve the reliability of the decision-making in information systems, we focus on three main aspects,

accuracy, ***transparency*** and ***fairness***, and develop the algorithms in the following aspects: 1) efficient online learning to rank, 2) explainable recommendation, and 3)fair online learning to rank.

1.1 Challenges and Insights

1.1.1 Effective Online Learning to Rank with Pairwise Exploration

The study of learning to rank with implicit user feedback such as click data has received considerable attention in both academia and industry. By collecting data from user interactions, we can better capture the true utility of instance for each user and create large-scale training data for ranking optimization without extensive and expensive human annotations. However, directly learning with user implicit feedback suffers from the intrinsic noise and bias in the interactions [5–7]. Online learning to rank (OL2R), which interactively optimizes and updates a ranking model after interacting with users, has gain more and more attention in recent decade. OL2R combats the bias by interventions, e.g., displaying a modified ranked list to users to complete the knowledge of the model, which is known as the exploration in OL2R.

Although influential and theoretically sound, the current OL2R solutions are not compatible with the successful practices in offline learning to rank, which directly optimize rankers by minimizing loss defined by rank-based metrics, such as Average Relevance Position (ARP) [8] or Normalized Discounted Cumulative Gain (NDCG) [9]. As a result, the performance of existing OL2R solutions is still behind that of offline solutions, which directly restricts OL2R's real-world applicability.

The key barrier separating the practices in online and offline learning to rank is the need of exploration. Since users' feedback is implicit and known to be biased and noisy [5–7], more clicks on a top-ranked document do not necessarily indicate greater relevance. The requirement of effective exploration in the problem space for the online model update introduces several challenges for OL2R. First, due to the combinatorial natural of ranking, the action space is too large to be efficiently explored (e.g., all permutations of returned documents). Current OL2R solutions either explore in the model space, or explore by taking a pointwise approach to estimated the utility of each query-document pair. For model space exploration, though an interleaved test makes it possible to compare different rankers with respect to a hidden utility function in an unbiased fashion, it is hard to link this comparison to the optimization of any rank-based metrics. And with the required uniform sampling in the model space, this type of OL2R solutions suffers from high variance and high regret during online result serving and model update [10]. While for action space exploration, it has been proven that it is inferior to the pairwise or listwise approaches in offline learning to rank studies [11]. Second, in the past decade, advances in deep neural networks (DNN) have made sifnificant strides in improving offline learning to rank models, thanks to DNN's strong representation learning power. But quite remarkably, most existing work in online learning to rank (OL2R) still assume a linear scoring function [10, 12, 13]. However, turning a neural ranker online is non-trivial. While DNNs can be accurate on learning given user feedback, i.e., exploitation, developing practical methods to balance exploration and exploitation in complex online learning problems remains largely unsolved. In essence, quantifying a neural model's uncertainty on new data points remains challenging. Third, efficiency is crucial for online models to be applied in practice. Given the existing theoretical studies in bandit algorithms [14–16], the added computational cost for exploration, e.g., quantifying the uncertainty the existing model is prohibitively high. For example, the cost of the confidence interval construction depends the number of parameters of the model, which is often in the order of 100 thousands.

Online learning to rank with pairwise exploration. Motivated by the nature of ranking that the complete ranking of instances can be decomposed into a series of pairwise comparisons, we propose to explore in the pairwise instance ranking space. Specifically, the ranker predicts the relative rank order between a pair of instances. Exploration is only needed on the pairs where the ranker is currently still uncertain about their rank orders, e.g., apply random permutation in their relative rankings; and for others, where the ranker is already certain about their rank orders, the predicted rank orders will be followed. In particular, the certain and uncertain rank orders are determined based on the model's estimation uncertainty, e.g., the confidence interval of the estimation. Our theoretical analysis shows that for both linear and neural ranker, the exploration space shrinks exponentially fast as the ranker estimation converges. This leads to the gap-dependent sublinear upper bound on the cumulative regret defined on the number of mis-ordered pairs, e.g., up to the order of $O(\log^4(T))$. Our extensive empirical evaluations also demonstrate the strong advantage the proposed model

against a rich set of state-of-the-art OL2R solutions over a collection of OL2R benchmark datasets on standard retrieval metrics.

Scalable exploration with perturbed feedback. Despite being theoretically sound and empirically effective, confidence interval based uncertainty calculation has remarkable limitation when being applied in practice: its computational cost for performing the required exploration is prohibitively high (almost *cubic* to the number of the model's parameters). In this work, to eliminate the explicit confidence set construction, we develop a bootstrapping technique to measure the uncertainty of the ranker's output via a set of sample estimates. In particular, N rankers are trained in parallel, each of which is learned with the user feedback and independently generated pseudo noise from a zero-mean Gaussian distribution. The overall model's estimation uncertainty is then determined by the ensemble of the estimates from all N rankers. Besides regular neural network updates, no additional computation is needed, which greatly reduces the computational overhead, and make the pairwise OL2R model operational in practice. We rigorously prove that this model obtains the same regret, i.e., $O(\log^4(T))$, but the computational complexity is way much lower.

1.1.2 Fair Online Learning to Rank

Existing ranking solutions that target at user-focused utility optimization unfortunately ignores the impact of result ranking on the content providers, who might receive differential attention from the users depending on their results' ranked positions. Compared to the scenario where a pre-trained ranker is deployed in offline ranking, the differential treatment can be ***accumulated and amplified*** in a faster rate during the course of OL2R. This is driven by the need of *exploration* in OL2R, where the algorithm has to continuously place an intentionally selected subset of results on top.

Though various exploration strategies have been proposed in OL2R literature [10, 12, 17–26], none of them consider fairness when presenting results to users. With the fairness concern, a new conflict emerges in the already complicated explore-exploit dilemma in OL2R. For instance, to ensure fairness, some results cannot be displayed, which slows down or even prevents online model learning. The slow improving relevance estimation in turn can lead to poor user experience (i.e., higher regret); in the meanwhile, if the bias from earlier OL2R update cannot be quickly eliminated by new feedback, unfair result ranking will be accumulated and amplified (e.g., one group is always mistakenly preferred). This leads to a new paradox among three elements in OL2R: ***fairness***, ***exploration*** and ***exploitation***. Most of the existing fair ranking solutions unfortunately do not apply to OL2R as they require expert-labeled relevance data or logged feedback for all items to train a ranker beforehand. Their prerequisite on the availability of relevance labels restricts these solutions in the OL2R setting, where the ranker is trained via the interactions with the users from scratch.

Calibrating exploration and exploration for fair exposure in OL2R. Group fairness concerns whether each group of instances receives fair treatment, e.g., comparable exposure across different content providers in rankings. In particular, the placement of groups (e.g., content providers), instead of the detailed placement of instances, affects the unfairness across groups. In this work, we develop a fair OL2R framework which confines the exploration within a subset of random permutations, where fairness across groups is maintained while the feedback is unbiased. Briefly, in each round of result serving in OL2R, candidate templates of group placements will first be selected based on the currently accumulated unfairness. Then ranked list then will be generated with respect to the chosen group-level templates and the required exploration. In other words, fairness is treated as a hard constraint when striking the balance between exploration and exploitation. Extensive empirical analysis of the resulting algorithm is performed to demonstrate the effectiveness and advantages of our proposed framework comparing to existing fair OL2R solutions.

1.1.3 Multi-task Learning for Explainable Recommendation

As recommendation systems are becoming more and more involved in daily life, they also shape people's personal and social life. Therefore, the opaque nature of most deployed recommendation algorithms, such as latent factor models [27], eagerly calls for transparency [3]. The lack of transparency in personalization [4] leaves users in a dilemma: a user can only assess the quality of personalized results by taking the suggested items, e.g., read the recommended articles; however, in order for him/her to adopt the system's customized results, he/she needs to first build trust over the system. Arguably, the most important value of explanations in an information system is not to convince users to adopt customized results (i.e., promotion), but to allow them

to make more informed and accurate decisions about which results to utilize (i.e., satisfaction) [2]. If users are convinced to accept personalized results that are subsequently found to be inaccurate, their confidence in the system will rapidly deteriorate [3, 28].

Explaining recommendation systems' output is non-trivial. Given the same information need, e.g., searching for diabetes medications, different users might hold distinct decision criteria: e.g., price vs. side-effect. And therefore, they will need different explanations in assessing the utility of personalized results. Precisely understanding users' preferences is the prerequisite for generating informative explanations to support their decision making. The first challenge in generating user-oriented explanation is to understand what the users want to know when making their decisions. For example, if an algorithm could predict the opinionated content that the user would provide on the recommended item, this content can serve as an informative explanation to illustrate why the user should pay attention to those recommended items. To obtain this level of user understanding, we need to first identify aspects or features that mostly affect users' decision making and infer their importance to the users. Second, the fidelity of explanation and the quality of personalization have long been considered as irreconcilable [29]: one has to trade personalization quality for explainability. Various solutions have been proposed to approximate the underlying personalization mechanisms for explanation [29, 30]. But to what extent these approximated explanations comply with the personalization models is unknown, i.e., no guarantee in explanation fidelity.

We argue that a good recommendation system should consist of companion learning tasks focusing on different aspects of users' decisions over the recommended items such that the observed final decisions (e.g., clicks or ratings) can be mutually explained by the associated observations. This insight motivates us to view explainable recommendation as a multi-task learning problem [31–33], where two companion learning tasks are formed: one focuses on ***user preference modeling for personalization***, and another focuses on ***content modeling for explanation***. In particular, we propose two multi-task learning solutions to enhance and explain the recommendations by leveraging the information from the user opinionated review text.

Explainable recommendation with joint tensor decomposition. With the available user opinion review content, rich information can be extracted with feature-level sentiment analysis techniques [34]. With such information, we propose to model the user, item, feature as a three-way tensor to describe users' preferences on specific features of the items. Besides, another two three-way tensors encoded with opinionated phrases are modelled for explanations. Via a joint tensor decomposition, user, item, feature and opinionated phrases can be projected to a shared latent representation space. In this way, the learning of latent factors for users and instances are thus shaped by optimizing the objectives of personalization and explanation tasks. The explanations can thus be generated by filling the predicated features and corresponding opinionated descriptions in predefined templates. Extensive experimental comparisons between our proposed solution and existing explainable recommendation algorithms demonstrate the effectiveness of our solution in both item recommendation and explanation generation tasks.

Explainable recommendation with factorization tree. Prior studies show that rule-based explanations are more easy to perceive and justify by the end-users [35]. Motivated by this, we propose to integrate the rule-based decision making into the learning of latent factors. More specifically, we treat the latent factors as a function of the rules: based on different outcomes of the rules, the associated groups of users and items should be routed to the designated latent factors, which are then optimized for recommendation. As the latent factors are learned subject to the explanation rules, the fidelity of explanations is ensured; and because latent factors are optimized for improving personalization, the quality of personalization is also provided. Extensive experiment evaluations demonstrate improved quality in recommendation and explanation of our proposed solutions, compared to a set of state-of-the-art explainable recommendation algorithms.

1.2 Dissertation Structure

The rest of this thesis is organized as follows. In Chapter 2, we introduce the solutions to improve the accuracy of the online ranking system with unreliable feedback. We propose a pairwise exploration framework, named as PairRank, for online learning to rank. In Chapter 3, we consider the fairness in ranking, and present the fair OL2R framework, FairExp, ensure fair exposure across different content providers. In Chapter 4, we describe the solutions to improve the transparency of the decision-making in recommendation systems. We develop

two multi-task learning solutions, MTER and FacT, for explainable recommendation by leveraging the rich information in users' text review data. In Chapter 5, we conclude this dissertation and discuss future research directions.

Chapter 2

Effective Online Learning to Rank with Pairwise Exploration

Modern information systems mostly learn from user interactions. Compared to learning from annotated datasets, implicit feedback obtained through user interactions matches user preferences more closely. Furthermore, learning from user interactions makes supervised learning of ranking models possible when collecting explicit annotations from experts is economically infeasible or even impossible (e.g., private collection search). However, a big disadvantage of user interaction is not guaranteed to be reliable; they often contain different types of bias and noise [5–7]. Online learning to rank (OL2R) [10, 12, 13, 19, 21–23, 25, 36] aims at removing the bias from the learning process by directly interacting with users. OL2R combat bias by interventions, i.e., by displaying slightly modified rankings to the users. Previous research shows that OL2R is more robust to selection bias, position bias and interaction noise than other debiasing method for learning to rank [37]. In this chapter, we introduce an OL2R framework with pairwise exploration, named as PairRank. We present the details of performing pairwise OL2R with linear and neural rankers, with constructed confidence interval of the model's estimation. Then, we introduce the extension focusing on scalable exploration, e.g., eliminating the explicit confidence set construction, to make such pairwise OL2R framework operational in practice

2.1 Online Learning to Rank with Pairwise Exploration

The essence of an OL2R algorithm is to infer the utility of individual documents or a parameterized ranking function via trial and error. As the system is learning while serving the users, and the users' implicit feedback is known to be biased and noisy [5–7, 38], the key challenge in OL2R is to balance the need to present the best results estimated so far to satisfy users (i.e., to exploit) and the need to present currently underestimated results to best improve the ranker (i.e., to explore). Although influential and theoretically sound, the current OL2R solutions have serious deficiencies. First, they cannot directly optimize rankers with respect to any ***rank-based metrics***, such as Mean Average Precision (MAP) or Normalized Discounted Cumulative Gain (NDCG) [39]. Second, due to the combinatorial nature of ranking, the ***action space is too large*** to be efficiently explored. As a result, the performance of existing OL2R solutions is still behind that of offline solutions, which directly restricts OL2R's real-world applicability.

In this work, we aim to bridge the gap between the offline and online learning to rank by directly training a pairwise learning to rank model online. We target pairwise ranking models for three major reasons. First, a pairwise ranker reduces the exponentially sized action space to ***quadratic***, by deriving the full ranking order from the pairwise comparisons between documents. Second, existing studies in search log analysis demonstrate relative preferences derived from clicks are ***more accurate and reliable*** than absolute judgments [5, 7]. Third, pairwise learning to rank models have ***competitive empirical performance*** and have been widely used in practical systems [9, 11, 40]. In particular, we explore in the pairwise ranking space of all candidate documents via ***divide-and-conquer***: we partition documents under a given query into several blocks in each round of result serving, where the document ranking across different blocks of the partition is certain (e.g., all documents in

one block should be ranked higher than those in another block), but the ranking among documents within each block is still uncertain. The ranked list is thus generated by a topological sort across blocks of a partition and randomly shuffling within each block. We name our solution PairRank. We rigorously prove that with either a linear ranker or a neural ranker, the exploration space shrinks exponentially fast as the ranker estimation converges, such that the cumulative regret defined on the number of mis-ordered pairs has a sublinear upper bound. Our extensive empirical evaluations also demonstrate the strong advantage of our model against a rich set of state-of-the-art OL2R solutions over a collection of OL2R benchmark datasets on standard retrieval metrics.

2.1.1 Related Work

Online learning to rank

We broadly categorize existing OL2R solutions into three families.

The first type learns the best ranked list for each individual query separately by modeling users' click and examination behaviors with multi-armed bandit algorithms [21, 22, 41, 42]. Typically, solutions in this category depend on specific click models to decompose the estimation on each query-document pair; as a result, exploration is performed on the ranking of individual documents. For example, by assuming users examine documents from top to bottom until reaching the first relevant document, cascading bandit models perform exploration by ranking the documents based on the upper confidence bound of their estimated relevance [42–44]. Other types of click models have also been explored (such as the dependent click model) [20–24]. However, as the relevance is estimated for each query-document pair, such algorithms can hardly be generalized to unseen queries or documents. Moreover, pointwise relevance estimation is proved to be ineffective for rank estimation in established offline learning to rank studies [9, 11].

The second type of OL2R solutions leverage ranking features for relevance estimation and explores for the best ranker in the entire model space [12, 23, 25]. The most representative work is Dueling Bandit Gradient Descent (DBGD) [12, 45], which proposes an exploratory direction in each round of interaction and uses an interleaved test [38] to validate the exploration for model update. To ensure an unbiased gradient estimate, DBGD uniformly explores the entire model space, which unfortunately costs high variance and high regret during online ranking and model update. Subsequent methods improved upon DBGD by developing more efficient sampling strategies, such as multiple interleaving and projected gradient, to reduce variance [10, 17–19, 46]. However, as exploration is performed in the model space, click feedback is used to infer which ranker is preferred under a hypothetical utility function. It is difficult to reason how the update in DBGD is related to the optimization of any rank-based metric. Hence, though generalizable, this type of OL2R solutions' empirical performance is still worse than classical offline solutions.

The third type of OL2R solutions explores the unknown by result perturbation. In [36], the authors adopt ϵ-greedy to estimate a stochastic RankSVM [40, 47] model on the fly. Though RankSVM is effective for pairwise learning to rank, the totally random exploration by ϵ-greedy is independent from the learned ranker. It keeps distorting the ranked results, even when the ranker has identified some high-quality results. In [25], the authors perform exploration by sampling the next ranked document from a Plackett-Luce model and estimate gradients of this ranking model from the inferred pairwise result preferences. Although exploration is linked to the ranker's estimation, the convergence of this solution is still unknown.

Theoretical analysis for neural network

Recently, substantial progress has been made to understand the convergence of deep neural networks [48–55]. A series of recent studies showed that (stochastic) gradient descent can find global minimal of training loss under moderate assumptions [48, 55–58]. Besides, the neural tangent kernel (NTK) [59] technique was proposed to describe the change of a DNN during gradient descent based training. This motivates the theoretical study of DNNs with kernel methods. Research in [60–64] showed that by connecting DNN with kernel methods, (stochastic) gradient descent can learn a function that is competitive with the best function in the corresponding neural tangent kernel space. In particular, under the framework of NTK, some recent work show that the confidence interval of the learned parameters of a DNN can be constructed based on the random feature mapping defined by the neural network's gradient [15, 16]. This makes the quantification of a neural model's uncertainty possible, and enables our uncertainty-based exploration for neural OL2R.

2.1.2 Problem Formulation

In OL2R, a ranker interacts with users for T rounds. At each round $t = 1, 2, ..., T$, the ranker receives a query q_t and its associated candidate documents, which are represented as a set of d-dimensional query-document feature vectors $\mathcal{X}_t = \{\mathbf{x}_1^t, \mathbf{x}_2^t, \dots \mathbf{x}_{V_t}^t\}$ with $\mathbf{x}_i^t \in \mathbb{R}^d$ and $\|\mathbf{x}_i^t\| \leq P$. The ranker determines the ranking of the candidate documents $\pi_t = \big(\pi_t(1), \pi_t(2), \dots, \pi_t(V_t)\big) \in \Pi([V_t])$, based on its knowledge so far, where $\Pi([V_t])$ represents the set of all permutations of V_t documents and $\pi_t(i)$ is the rank position of document i under query q_t. Once the ranked list is returned to the user, the user examines the results and provides his/her click feedback $C_t = \{c_1^t, c_2^t, ..., c_{V_t}^t\}$, where $c_i^t = 1$ if the user clicks on document i at round t; otherwise $c_i^t = 0$. Based on this feedback, the ranker updates itself and proceeds to the next query.

C_t is known to be biased and noisy [5–7]. Existing studies find that users tend to click more on higher-ranked documents, as they stop examination early in the list. This is known as position bias. And users can only interact with the documents shown to them, known as the presentation bias. As a result, the ranker cannot simply treat non-clicked documents as irrelevant. Such implicit feedback imposes several key challenges for online learning to rank: how to deal with the implicit feedback, and in the meanwhile, how to effectively explore the unknowns for the model update. Following the practice in [5], we treat clicks as relative preference feedback and assume that clicked documents are preferred over the ***examined but unclicked*** ones. In addition, we consider every document that precedes a clicked document and the first subsequent unclicked document as examined. This approach has been widely adopted and proven to be effective in learning to rank [6, 10, 25]. Accordingly, we use o_t ($o_t \leq V_t$) to represent the index of the last examined position in π_t at round t.

Exploration is the key component that differentiates OL2R to offline L2R, where OL2R needs to serve while learning from its presented rankings. The most straightforward exploration is to provide a random list of candidate documents. However, such random exploration is less appreciated for OL2R as it hurts user experience, even though it may be beneficial for model training. Therefore, regret becomes an important metric for evaluating OL2R. Though various types of regret have been defined and analyzed in existing OL2R studies, few of them link to any rank-based metric, which is the key in ranker evaluation. For example, for OL2R solutions that explore in the document space, regret is typically defined on the number of clicks received on the presented ranking versus that known only in hindsight [21, 22, 42, 44]. For solutions that explore in the model space, such as DBGD, regret is defined as the number of rounds where the chosen ranker is preferred over the optimal ranker [10, 12]. It is difficult to reason how such measures indicate an OL2R solution's ranking performance against a desired retrieval metric, such as ARP and NDCG. To bridge this gap, we define regret by the number of mis-ordered pairs from the presented ranking to the ideal one, i.e., the Kendall tau rank distance,

$$\mathbf{R}_T = \mathbb{E}\big[\sum_{t=1}^{T} r_t\big] = \mathbb{E}\big[\sum_{t=1}^{T} K(\pi_t, \pi_t^*)\big]$$

where $K(\pi_t, \pi_t^*) = \Big|\big\{(i,j) : i{<}j, \big(\pi_t(i){<}\pi_t(j) \wedge \pi_t^*(i){>}\pi_t^*(j)\big) \vee \big(\pi_t(i){>}\pi_t(j) \wedge \pi_t^*(i){<}\pi_t^*(j)\big)\big\}\Big|$.

Remark 2.1.1. As shown in [65], most ranking metrics, such as Average Rank Position (ARP) and Normalized Discounted Cumulative Gain (NDCG), can be decomposed into pairwise comparisons; hence, this regret definition connects an OL2R algorithm's online performance with classical rank evaluations. We consider it more informative than "pointwise" regret defined in earlier work [22, 42].

2.1.3 Online Pairwise learning to Rank

The key in OL2R is to effectively explore the unknowns while providing high-quality results to the users, which is often referred to as the explore-exploit trade-off. In this work, we propose to directly train a pairwise model from its interactions with users and explore in the pairwise document ranking space via a ***divide-and-conquer*** strategy.

The high-level idea is illustrated in Figure 2.1. The insight is to handle the prohibitively large exploration space based on the nature of ranking: a complete ranking of instances can be decomposed into a series of pairwise comparisons, which reduces the exponentially sized action space to quadratic. We can then learn and explore in a parametric pairwise ranking space of all candidate instances via divide-and-conquer. Specifically, in each round of result serving, we partition all the estimated pairwise comparisons into two sets, the certain pairs and the uncertain pairs, according to the ranker's current prediction about those instances' relative rank

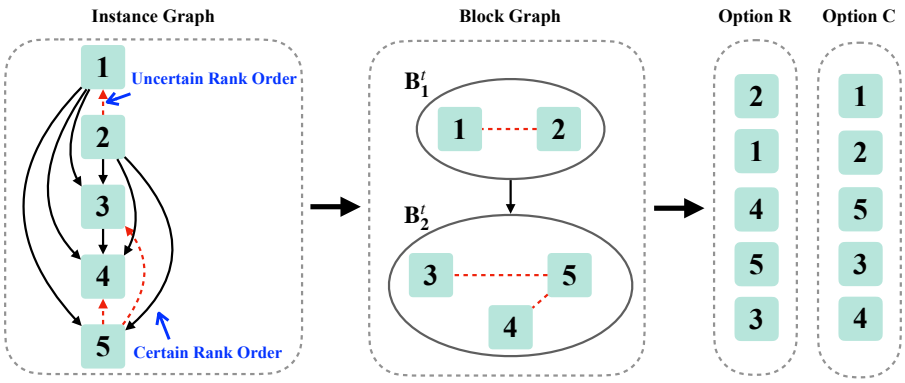


Figure 2.1: Pairwise explore and exploit by divide-and-conquer. Assume that the optimal ranking order among the 5 documents is $1 \succ 2 \succ 3 \succ 4 \succ 5$. At the current round t, the ranker is confident about its preference estimation between all the pairs expect $(1,2), (3,5), (4,5)$. In this example, the instantaneous regret of the first proposed ranked list is 3 and 2 in the second proposal.

order. Exploration is only needed on the pairs where the ranker is currently still uncertain about their rank orders, e.g., apply random permutation in their relative rankings; and for others, the ranker's predicted rank orders will be followed. In particular, we partition documents under a given query in each round of result serving into several blocks, where the document ranking across different blocks of the partition is certain (e.g., all documents in one block should be ranked higher than those in another block), but the ranking among documents within each block is still uncertain. The ranked list is thus generated by a topological sort across blocks of a partition and randomly shuffling within each block. We name our solution PairRank. For example, referring to the example shown in Figure 2.1, assume that the optimal ranking order among the 5 instances is $1 \succ 2 \succ 3 \succ 4 \succ 5$; and at the current round t, the ranker is confident about its pairwise prediction between all the pairs except $(1,2), (3,5), (4,5)$. Accordingly, two blocks of instances are constructed, e.g., the current model is certain that instances in block B_1^t are more relevance than instances in block B_2^t under the given query q_t. When generating the ranked list, we can randomly shuffle the order between instances 1 and 2 (i.e., to explore) while preserving the order between instance 1 and instance 3, 4, 5 (i.e., to exploit).

To perform such pairwise exploration, the key is to accurately quantifying the current ranker's uncertainty on pairwise rank prediction. The main reason behind the ranker's estimation uncertainty is the existence of noise in the click feedback. Passing through the learning process, such noise leads to the uncertainty in the ranker's parameter estimation, which then contributes to the uncertainty in its predicted relative rank order. Hence, accurate noise modeling and rigorous analysis of the parameter estimation convergence are the two key technical components of this thrust of PairRank. In the following sections, we present the details of PairRank with linear and neural rankers.

Learning with a Linear Ranker

We first adopt a single layer RankNet model with a sigmoid activation function [9] as our pairwise ranker. This choice is based on the promising empirical performance of RankNet and the feasibility of analyzing the resulting online solution's convergence. With $f(\cdot)$ as the relevance mapping function, in a single layer RankNet, the relevance can be obtained by the linear combination of the query-document feature vector $\mathbf{x}$ and the model $\boldsymbol{\theta}$, e.g., $f(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\theta}$. The probability that a document i is more relevant than document j under query q is computed as $\mathbb{P}(i \succ j|q) = \sigma(f(\mathbf{x}_i) - f(\mathbf{x}_j)) = \sigma(\mathbf{x}_i^\top \boldsymbol{\theta} - \mathbf{x}_j^\top \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ and $\|\boldsymbol{\theta}\| \leq Q$ is the model parameter and $\sigma(s) = 1/(1+\exp(-s))$. To simplify our notations, we use $\mathbf{x}_{ij}$ to denote $\mathbf{x}_i - \mathbf{x}_j$ in our subsequent discussions.

Linear ranking model. With the knowledge of $\boldsymbol{\theta}$, due to the monotonicity and transitivity of the sigmoid function, the ranking of documents in $\mathcal{X}$ can be determined by $\{\mathbf{x}_1^\top \boldsymbol{\theta}, \mathbf{x}_2^\top \boldsymbol{\theta}, \ldots, \mathbf{x}_{V_t}^\top \boldsymbol{\theta}\}$. Therefore, the key of learning a RankNet model is to estimate its parameter $\boldsymbol{\theta}$. As RankNet specifies a distribution on pairwise comparisons, the objective function for $\boldsymbol{\theta}$ estimation can be readily derived as the cross-entropy loss between

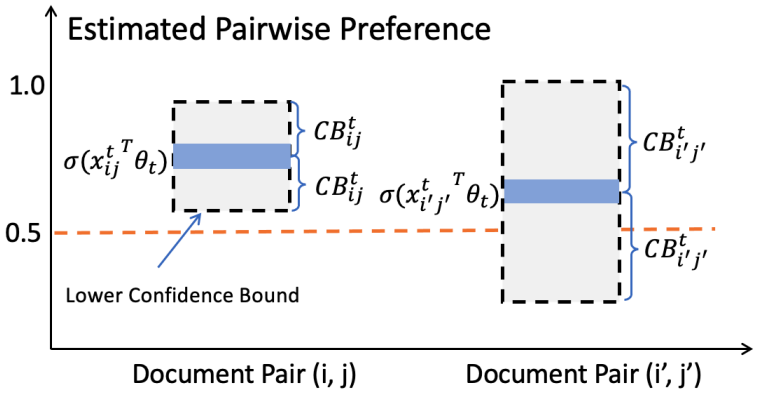


Figure 2.2: Illustration of certain and uncertain rank orders.

the predicted pairwise distribution on documents and those inferred from user feedback till round t:

$$\mathcal{L}_t = \sum_{s=1}^{t} \sum_{(i,j)\in\Omega_s} -y_{ij}^s \log\left(\sigma(\mathbf{x}_{ij}^{s\top}\boldsymbol{\theta})\right) - (1-y_{ij}^s)\log\left(1-\sigma(\mathbf{x}_{ij}^{s\top}\boldsymbol{\theta})\right) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 \tag{2.1.1}$$

where λ is the l_2 regularization coefficient, Ω_s denotes the set of document pairs that received different click feedback at round s, i.e., $\Omega_s = \{(i,j) : c_i^s \neq c_j^s, \forall \pi(i) < \pi(j) \leq o_s\}$, y_{ij}^s indicates whether the document i is preferred over document j in the click feedback, i.e., $y_{ij}^s = \frac{1}{2}(c_i^s - c_j^s) + \frac{1}{2}$ [9]. Due to the log-convexity of the loss function defined in Eq (2.1.1), the global optimal solution $\widehat{\boldsymbol{\theta}}_t$ at round t exists and can be efficiently obtained by gradient descent.

Online learning of RankNet boils down to the construction of $\{\Omega_s\}_{s=1}^T$ over time. However, the conventional practice of using all the inferred pairwise preferences from click feedback [6,40] imposes a higher risk in an online setting. In the presence of click noise (e.g., a user mistakenly clicks on an irrelevant document), pairing documents would cause a quadratically increasing number of noisy training instances, which impose strong negative impact on the quality of the learned ranker [66]. As the updated ranker is immediately executed, cascading of ranker deterioration is possible. To alleviate this deficiency, we propose to only use independent pairs inferred from the feedback, e.g., $\Omega_s^{ind} = \{(i,j) : c_i^s \neq c_j^s, \forall(\pi_s(i), \pi_s(j)) \in D_s\}$, where D represents the set of disjointed position pairs, for example, $D_s = \{(1,2),(3,4),...(o_t-1,o_t)\}$. In other words, we will only use a subset of non-overlapping pairwise comparisons for our online ranker update.

Uncertainty quantification. As discussed before, $\widehat{\boldsymbol{\theta}}_t$ is obtained based on the acquired feedback from what has been presented to the user, which is subject to various types of biases and noises [5–7]. Hence, $\widehat{\boldsymbol{\theta}}_t$ only reflects what the ranker knows so far; and it is vital to effectively explore the unknowns to complete its knowledge. In PairRank, we propose to explore in the pairwise document ranking space spanned by $\mathcal{X}_t$ under q_t, with respect to the current ranker's uncertainty about the pairwise comparisons.

The model estimation uncertainty is caused by the existence of click noise, i.e., $\|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\| \neq 0$, where $\boldsymbol{\theta}^*$ is the underlying ground-truth model parameter. And this model estimation uncertainty directly leads to the uncertainty in the ranker's pairwise preference estimation. To quantify the source of uncertainty, we follow conventional click models to assume that as long as the documents are examined, the clicks are independent of each other given the true relevance of documents, so as their noise [5,67,68]. As a result, the pairwise noise becomes the sum of noise from the two associated clicks. Because we only use the independent document pairs Ω^{ind}, the pairwise noise is thus independent of each other and the history of result serving, which directly leads to the following proposition.

Proposition 2.1.2. For any $t \in [T]$, $\forall (i,j) \in \Omega_t^{ind}$, the pairwise feedback follows $y_{ij}^t = \sigma({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}^*) + \xi_{ij}^t$, where ξ_{ij}^t satisfies that for all $\beta \in \mathbb{R}$, $\mathbb{E}[\exp(\beta \xi_{ij}^t) | \{\{\xi_{i'j'}^s\}_{(i',j') \in \Omega_s^{ind}}\}_{s=1}^{t-1}, \Omega_{1:t-1}^{ind}] \leq \exp(\beta^2 R^2)$, is a R-sub-Gaussian random variable.

According to the property of sub-Gaussian random variables, such assumption can be easily satisfied in practice as long as the pointwise click noise follows a sub-Gaussian distribution. For example, the pointwise noise can be modeled as a binary random variable related to the document's true relevance under the given query, which follows $\frac{1}{2}$-sub-Gaussian distribution. Therefore, based on the solution of Eq (2.1.1), the uncertainty of the estimated pairwise preference $\sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t)$ by RankNet at round t can be analytically bounded with a high probability, as shown in the following lemma.

Lemma 2.1.3. (Confidence Interval of Pairwise Preference Estimation for Linear Ranker). At round $t \in [T]$, for any pair of documents $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$ under query q_t, with probability at least $1 - \delta_1$, we have,

$$|\sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t) - \sigma({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}^*)| \leq \alpha_t \|\mathbf{x}_{ij}^t\|_{{\mathbf{A}_t^{lin}}^{-1}},$$

where $\alpha_t = (\frac{2k_\mu}{c_\mu})\Big(\sqrt{R^2 \log(\frac{\det(\mathbf{A}_t^{lin})}{\delta_1^2 \det(\lambda \mathbf{I})})} + \sqrt{\lambda} Q\Big)$, $\mathbf{A}_t^{lin} = \lambda \mathbf{I} + \sum_{s=1}^{t-1} \sum_{(i',j') \in \Omega_s^{ind}} \mathbf{x}_{i'j'}^s {\mathbf{x}_{i'j'}^s}^\top$, λ is the regularization coefficient, k_μ is the Lipschitz constant of the sigmoid link function σ, $c_\mu = \inf_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \dot{\sigma}(\mathbf{x}^\top \boldsymbol{\theta})$, with $\dot{\sigma}$ as the first derivative of σ, and R is the sub-Gaussian parameter for noise ξ.

This lemma provides a tight high probability bound of the pairwise preference estimation uncertainty under a single layer RankNet specified by $\widehat{\boldsymbol{\theta}}_t$, which enables us to perform efficient pairwise exploration for model update. To better illustrate our exploration strategy based on the pairwise estimation uncertainty, we introduce the following definition on document pairs.

Definition 2.1.4. (Certain Rank Order for Linear Ranker) At any round $t \in [T]$, the ranking order between document i and j, denoted as (i,j), is considered in a certain rank order if and only if $\sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t) - \alpha_t \|\mathbf{x}_{ij}^t\|_{{\mathbf{A}_t^{lin}}^{-1}} > \frac{1}{2}$.

Intuitively, based on Lemma 2.1.3, if (i,j) is in a certain rank order, with a high probability that the estimated preference (order) between document i and j is consistent with the ground-truth. For example, as shown in Figure 2.2, $\sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t)$ and $\sigma({\mathbf{x}_{i'j'}^t}^\top \widehat{\boldsymbol{\theta}}_t)$ represent the estimated pairwise preference on document pair (i,j) and (i',j') based on $\widehat{\boldsymbol{\theta}}_t$, while CB_{ij}^t and $CB_{i'j'}^t$ represent the corresponding confidence bound defined in Lemma 2.1.3, i.e., $CB_{ij}^t = \alpha_t \|\mathbf{x}_{ij}^t\|_{{\mathbf{A}_t^{lin}}^{-1}}$ and $CB_{i'j'}^t = \alpha_t \|\mathbf{x}_{i'j'}^t\|_{{\mathbf{A}_t^{lin}}^{-1}}$. According to Lemma 2.1.3, we know that the ground-truth pairwise preferences, $\sigma({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}^*)$ and $\sigma({\mathbf{x}_{i'j'}^t}^\top \boldsymbol{\theta}^*)$ lie within the corresponding confidence intervals with a probability at least $1 - \delta_1$, i.e., $\sigma({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}^*) \in [\sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t) - CB_{ij}^t, \sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t) + CB_{ij}^t]$. In Figure 2.2, for pair (i,j), the lower bound of the pairwise estimation, $\sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t) - CB_{i,j}^t$, is greater than $\frac{1}{2}$. This indicates that with a high probability $1 - \delta_1$, the estimated order between document i and j is consistent with the ground-truth; and thus there is no need to explore this pair. In contrast, with $\sigma({\mathbf{x}_{i'j'}^t}^\top \widehat{\boldsymbol{\theta}}_t) - CB_{i'j'}^t < \frac{1}{2}$, the estimated order $(i' \succ j')$ is still uncertain as the ground-truth model may present an opposite order e.g., $j' \succ i'$; hence, exploration on this pair is necessary. We use ω_c^t to represent the set of all certain rank orders at round t. Accordingly, the set of uncertain rank orders at round t is defined as: $\omega_u^t = \{(i,j) \in \Psi_t : (i,j) \notin \omega_c^t \wedge (j,i) \notin \omega_c^t\}$, and Ψ_t represent the set of all possible document pairs at round t, e.g., $\Psi_t = \{(i,j) \in [V_t]^2, i \neq j\}$ and $|\Psi_t| = V_t^2 - V_t$.

Explore the unknowns via divide-and-conquer. With the aforementioned pairwise estimation uncertainty and the corresponding sets of certain and uncertain rank orders, i.e., ω_c^t and ω_u^t, we can effectively explore the unknowns. Intuitively, we only need to randomize the ranking of documents among which the model is still uncertain about their ranking orders, i.e., the uncertain rank orders, and therefore obtain feedback to further update the model (and reduce uncertainty). This naturally leads to a divide-and-conquer exploration strategy in the space of pairwise document comparisons. Specifically, we partition $\mathcal{X}_t$ into different blocks based on ω_c^t and ω_u^t such that the ranking orders between any documents belonging to different blocks are certain, shown in the block graph in Figure 2.1. The ranked list can thus be generated by topological sort across blocks and

Algorithm 1 PairRank: online learning to rank with pairwise exploration

1: **Inputs:** λ, δ_1, δ_2
2: Initiate $\mathbf{M}_0 = \lambda\mathbf{I}, \widehat{\boldsymbol{\theta}}_1 = 0$
3: **for** $t = 1$ to T **do**
4: Receive query q_t and its corresponding candidate documents set $\mathcal{X}_t = \{\mathbf{x}_1^t, \mathbf{x}_2^t, ..., \mathbf{x}_{L_t}^t\}$.
5: $\omega_c^t = \{(i,j) \in \Psi_t : \sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_{t-1}) - \alpha_{t-1}\|\mathbf{x}_{ij}^t\|_{\mathbf{A}_{t-1}} > 1/2\}$
6: $\omega_u^t = \{(i,j) \in \Psi_t : (i,j) \notin \omega_c^t \wedge (j,i) \notin \omega_c^t\}$
7: Construct ordered block list $\mathcal{B}_t = \{\mathcal{B}_1^t, \mathcal{B}_2^t, ...\mathcal{B}_{d_t}^t\}$
8: Generate ranked list $\pi_t = \{\pi(\mathcal{B}_1^t), \pi(\mathcal{B}_2^t), ..., \pi(\mathcal{B}_{d_t}^t)\}$
9: Observe click feedback C_t, and corresponding o_t.
10: $\Omega_t^{ind} = \{(i,j) : c_i^t \neq c_j^t \wedge (\pi_t(i), \pi_t(j)) \in D\}$.
11: Set $\widehat{\theta}_t$ as the solution of minimizing Eq (2.1.1).
12: $\mathbf{A}_t^{lin} = \mathbf{A}_{t-1}^{lin} + \sum_{(i,j)\in\Omega_t^{ind}} \mathbf{x}_{ij}^t {\mathbf{x}_{ij}^t}^\top$.

random shuffling within each block. As the exploration is confined to the pairwise ranking space, it effectively reduces the exponentially sized action space to quadratic.

Algorithm 1 shows the detailed steps. At round t, we first construct ω_c^t and ω_u^t according to the current mode $\widehat{\boldsymbol{\theta}}_{t-1}$. Then, we create blocks of $\mathcal{X}_t$ according to the definition below.

Definition 2.1.5. (Block) At any round $t \in [T]$, any block $\mathcal{B}_b^t$ in block list $\mathcal{B}_t$ satisfies,

1. $\forall i \in \mathcal{B}_d^t, (i,j) \in \omega_c^t$ for any $j \in [L_t] \setminus \mathcal{B}_d^t$.
2. $\nexists k \in [V_t] \setminus \mathcal{B}_d^t$, for $i,j \in \mathcal{B}_d^t, (i,k) \in \omega_c^t \wedge (k,j) \in \omega_c^t$

Intuitively, each block is a subset of documents linked to each other by the uncertain rank order. It can be viewed as a connected component in an undirected graph with documents as vertices and uncertain rank order as edges under a given query. The connected components (blocks) can be found by linearly scanning through the vertices, based on breadth-first search or depth-first search if a vertex is not visited before. When the algorithm stops, each vertex (document) will be assigned to a connected component (block). Once the blocks are constructed, the order of the blocks can be obtained by topological sort (line 6). Let $\mathcal{B}_t = \{\mathcal{B}_1^t, \mathcal{B}_2^t, ...\mathcal{B}_{d_t}^t\}$ be the ordered block list for $\mathcal{X}_t$ at round t, the ranked list π_t is generated as $\pi_t = \{\pi(\mathcal{B}_1^t), \pi(\mathcal{B}_1^t), ..., \pi(\mathcal{B}_{d_t}^t)\}$, where $\pi(\cdot)$ randomly permutes its input set as output.

Improving exploration efficiency. To further improve exploration efficiency, we propose two options to generate the ranked list. As shown in Figure 2.1, the first ranked list is generated by randomly shuffling all documents within each block (referred to as random exploration), while in the second list, only the uncertain rank orders are shuffled, and the certain ones are preserved (referred to as conservative exploration). More specifically, for the second option, due to the monotonicity and transitivity of the sigmoid function, the document graph constructed with the candidate documents as the vertices and the certain rank order as the directed edges is a directed acyclic graph (DAG). We can perform a topological sort on the constructed document graph to efficiently generate the final ranked list. The certain rank orders are preserved by topological sort to exploit the ranker's high confidence predictions. On the other hand, the topological sort randomly chooses vertices with zero in-degree, among which there is no certain rank orders. This naturally achieves exploration among uncertain rank orders.

Learning with a Neural Ranker

In the past decade, advances in deep neural networks (DNN) have made significant strides in improving offline learning to rank models [9, 69], thanks to DNN's strong representation learning power. However, most of the existing OL2R models assume that the expected relevance of a document under the given query can be characterized by a linear function in the feature space, which unfortunately often fails in practice. With a linear assumption, the potentially complex non-linear relations between queries and documents are ignored. For

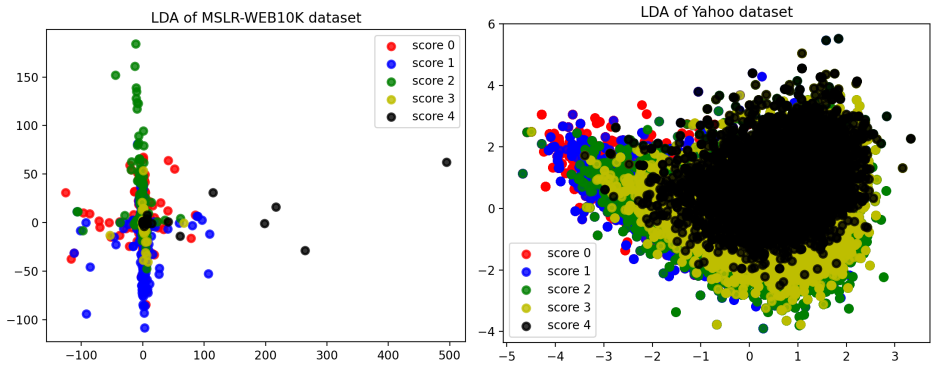


Figure 2.3: Linear discriminative analysis on two public benchmark datasets.

example, classical query-document features are usually constructed in parallel to the design and choices of ranking models. As a result, a lot of correlated and sometimes redundant features are introduced for historical reasons; and the ranker is expected to handle it. For instance, the classical keyword matching based features, such as TF-IDF, BM25 and language models, are known to be highly correlated [70]; and the number of in-links is also highly related to the PageRank feature.

To verify this issue, Figure 2.3 shows the linear discriminative analysis (LDA) [71] on two public learning to rank benchmark datasets.åThe technique of LDA is typically used for multi-class classification that automatically performs dimensionality reduction, providing a projection of the dataset that can best linearly separate the samples by their assigned class. We provide the entire labeled dataset for the algorithm to learn the separable representation. We set the reduced dimension to be two to visualize the results. In Figure 2.3, we can clearly observe that a linear model is insufficient to separate the classes in both datasets.

Besides, classical learning to rank tasks take the well-constructed query-document features as input, while in practice, such query-document features might not be always available. Thus, a neural ranker that has the ability to handle the raw input data, e.g., learning embeddings from query-document raw text data, has more advantage in real-world problems. To unleash the power of DNNs in OL2R, we propose to extend PairRank with deep ranking models.

Similar to PairRank with a linear ranker, we need to 1) estimate the pairwise preference with a neural ranker, and 2) quantify the estimation uncertainty of the neural ranker to perform the pairwise exploration. The first part can be achieved by directly replacing the linear function with a deep neural network to capture the non-linearity. The key challenge comes from the analytical quantification of the uncertainty of a neural model's prediction. This requirement seems impossible years ago; but thanks to the substantial progress made in recent theoretical deep learning, we now have useful tools to analyze the expressive power of DNNs. A series of recent studies showed that (stochastic) gradient descent can find global minima of training loss under moderate assumptions [48, 55–58]. As the generalization to a DNN model, studies in [60–64] showed that by using (stochastic) gradient descent, the learned parameters of a DNN are located in a particular regime, and the generalization error bound of the DNN can be characterized by the best function in the corresponding neural tangent kernel space [59]. In particular, under the framework of neural tangent kernel, two recent papers [15, 16] proposed that the confidence interval of the learned parameters of a DNN can be constructed based on the random feature mapping defined by the neural network's gradient on the input instances. The uncertainty of the pairwise rank estimation under a neural ranker can thus be quantified accordingly, which makes directly learning a neural ranking model online possible. Next, we will introduce the details of neural PairRank.

Neural ranking model. In order to learn the underlying optimal function h[1], we utilize a fully connected neural network $f(\mathbf{x};\boldsymbol{\theta})$ defined as $f(\mathbf{x};\boldsymbol{\theta}) = \sqrt{m}\mathbf{W}_L\phi(\mathbf{W}_{L-1}\phi(\dots\phi(\mathbf{W}_1\mathbf{x}))$, where depth $L \geq 2$, $\phi(\mathbf{x}) = \max\{\mathbf{x}, 0\}$, and $\mathbf{W}_1 \in \mathbb{R}^{m\times d}$, $\mathbf{W}_i \in \mathbb{R}^{m\times m}$, $2 \leq i \leq L-1$, $\mathbf{W}_L \in \mathbb{R}^{m\times 1}$, and $\boldsymbol{\theta} =$

[1] for a linear ranker, $h(\mathbf{x}) = \mathbf{x}^\top\boldsymbol{\theta}^*$ where $\boldsymbol{\theta}^*$ is the underlying optimal model.

$[\text{vec}(\mathbf{W}_1)^\top, \ldots, \text{vec}(\mathbf{W}_L)^\top]^\top \in \mathbb{R}^p$ with $p = m + md + m^2(L-2)$. Without loss of generality, we assume the width of each hidden layer is the same as m, concerning the feasibility of theoretical analysis. We also denote the gradient of the neural network function with $\mathbf{g}(\mathbf{x};\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} f(\mathbf{x};\boldsymbol{\theta}) \in \mathbb{R}^p$. We slightly change the objective function of the pairwise ranker for easy analysis:

$$\mathcal{L}_t(\boldsymbol{\theta}) = \sum_{s=1}^{t} \sum_{(i',j') \in \Omega_s^{ind}} -(1 - y_{i'j'}^s) \log(1 - \sigma(f_{i'j'}^s)) - y_{i'j'}^s \log(\sigma(f_{i'j'}^s)) + \frac{m\lambda}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2, \quad (2.1.2)$$

where m is the width of the neural network, λ is the l_2 regularization coefficient, $f_{i'j'}^s = f(\mathbf{x}_{i'};\boldsymbol{\theta}_{s-1}) - f(\mathbf{x}_{j'};\boldsymbol{\theta}_{s-1})$.

We can see that eq (2.1.2) is an l_2-regularized cross-entropy loss, where the regularization term centers at the randomly initialized network parameter $\boldsymbol{\theta}_0$. Follow the same criteria of inferring the pairwise preference from the implicit feedback in PairRank with linear ranker, we can obtain the training data observed so far. Then we adopt gradient descent to minimize eq (2.1.2) with step size η and total number of iterations J.

Uncertainty quantification. To quantify the estimation uncertainty, we follow the same assumptions on the click feedback and click noise in the previous sections (Proposition 2.1.2). By minimizing the loss function (2.1.2) with gradient descent, we obtain the following lemma.

Lemma 2.1.6. (Confidence Interval of Pairwise Preference Estimation under Non-linear Assumption). There exist positive constants C_1 and C_2 such that for any $\delta_1 \in (0,1)$, if the step size of gradient descent $\eta \leq C_1(TmL + m\lambda)^{-1}$ and $m \geq C_2 \max\{\lambda^{-1/2} L^{-3/2} (\log(TV_{\max}L^2/\delta_1))^{3/2}, T^7\lambda^{-7}L^{21}(\log m)^3\}$, then at round $t < T$, for any document pair $(i,j) \in \Psi_t$ under query q_t, with probability at least $1 - \delta_1$,

$$|\sigma(f_{ij}^t) - \sigma(h_{ij})| \leq \alpha_t \|\mathbf{g}_{ij}^t / \sqrt{m}\|_{\mathbf{A}_t^{neu-1}} + \epsilon(m), \quad (2.1.3)$$

where $V_{\max}$ represents the maximum number of documents under a query over time, $h_{ij} = h(\mathbf{x}_i) - h(\mathbf{x}_j)$, $\mathbf{g}_{ij}^s = \mathbf{g}(\mathbf{x}_i;\boldsymbol{\theta}_s) - \mathbf{g}(\mathbf{x}_j;\boldsymbol{\theta}_s)$, $\mathbf{A}_t^{neu} = \sum_{s=1}^{t-1} \sum_{(i',j') \in \Omega_s^{ind}} \frac{1}{m} \mathbf{g}_{i'j'}^s {\mathbf{g}_{i'j'}^s}^\top + \lambda \mathbf{I}$, $\bar{C}_1$, $\bar{C}_2$, $\bar{C}_3$ and $\bar{C}_4$ are positive constants,

$$\begin{aligned}
\epsilon(m) =& \bar{C}_1 \Big(T^{\frac{7}{6}} m^{-\frac{1}{6}} \lambda^{-\frac{7}{6}} L^4 \sqrt{\log(m)} (1 + \sqrt{T/\lambda}) + (1 - \eta m \lambda)^{\frac{J}{2}} \sqrt{TL/\lambda} \\
&+ T^{\frac{1}{6}} m^{-\frac{1}{6}} \lambda^{-\frac{1}{6}} L^{\frac{7}{2}} \sqrt{\log(m)} S + T^{\frac{2}{3}} m^{-\frac{1}{6}} \lambda^{-\frac{2}{3}} L^3 \sqrt{\log(m)} \Big), \\
\alpha_t =& \Big(1 + \bar{C}_2 T^{\frac{7}{6}} m^{-\frac{1}{6}} \sqrt{\log(m)} \lambda^{-\frac{7}{6}} L^4 \Big)^{\frac{1}{2}} \cdot \bar{\alpha}_t, \\
\bar{\alpha}_t =& \Big(\sqrt{\lambda} \bar{C}_3 + (\nu^2 \log(\frac{\det(\mathbf{A}_t^{neu})}{\delta_1^2 \det(\lambda \mathbf{I})}) + \bar{C}_4 T^{\frac{5}{3}} m^{-\frac{1}{6}} \lambda^{-\frac{1}{6}} L^4 \sqrt{\log(m)})^{\frac{1}{2}} \Big).
\end{aligned}$$

Accordingly, the certain rank order can be defined as follows,

Definition 2.1.7. (Certain Rank Order) At round t, the rank order between documents $(i,j) \in \Psi_t$ is in a certain rank order if and only if $\sigma(f_{ij}^t) - CB_{ij}^t > \frac{1}{2}$, where $CB_{ij}^t = \alpha_t \|\mathbf{g}_{ij}^t / \sqrt{m}\|_{\mathbf{A}_t^{-1}} - \epsilon(m)$ is the width of confidence bound about the estimated pairwise rank order.

With the constructed sets of certain and uncertain rank orders, the ranked list can be generated according to the ranking policy discussed in the previous section.

Extend to LambdaRank. LambdaRank directly optimizes the ranking metric of interest (e.g., NDCG) with a modified gradient based on RankNet [9]. For a given pair of documents, the confidence interval of LambdaRank's estimation can be calculated by gradients of the neural network in the same way as in RankNet (i.e., by Lemma 2.1.6). However, as the objective function of LambdaRank is unknown, it prevents us from theoretically analyzing the resulting online algorithm's regret. But similar empirical improvement

from LambdaRank against RankNet known in the offline settings [9] is also observed in our online versions of these two algorithms.

2.1.4 Regret Analysis

We theoretically analyze the regret of the propose pairwise exploration model, which is defined by the cumulative number of mis-ordered pairs in its proposed ranked list till round T. The key in analyzing this regret is to quantify how fast the model achieves certainty about its pairwise preference estimation in candidate documents. First, we define E_t as the success event at round t:

$$E_t = \big\{\forall (i,j) \in [V_t]^2, |\sigma(f_{ij}^t) - \sigma(h_{ij})| \leq CB_{ij}^t\big\}.$$

In the above definition, f_{ij}^t represents the estimated pairwise preference between document i and j, while h_{ij} represents the ground truth pairwise preference. This definition applies for both linear and neural rankers, with corresponding confidence interval defined in Lemma 2.1.3 and Lemma 2.1.6. Intuitively, E_t is the event that the estimated model is "close" to the optimal model at round t. According to Lemma 2.1.3 and Lemma 2.1.6, it is easy to reach the following conclusion,

Corollary 2.1.8. On the event E_t, it holds that $\sigma(h_{ij}) > 1/2$ if $(i,j) \in \omega_c^t$.

Our pairwise exploration suffers regret as a result of misplacing a pair of documents, i.e., swapping a pair already in the correct order. Based on Corollary 2.1.8, under event E_t, the certain rank order identified is consistent with the ground-truth. As our partition design always places documents under a certain rank order into distinct blocks, under event E_t the ranking order across blocks is consistent with the ground-truth. In other words, regret only occurs when ranking documents within each block.

To analyze the regret caused by random shuffling within each block, we need the following technical lemma derived from random matrix theory. We adapted it from Equation (5.23) of Theorem 5.39 from [72].

Lemma 2.1.9. Let $A \in \mathbb{R}^{n\times d}$ be a matrix whose rows $\mathbf{M}_i$ are independent sub-Gaussian isotropic random vectors in $\mathbb{R}^d$ with parameter σ, namely $\mathbb{E}[\exp(x^\top(\mathbf{M}_i - \mathbb{E}[\mathbf{M}_i])] \leq \exp(\sigma^2\|x\|^2/2)$ for any $x \in \mathbb{R}^d$. Then, there exist positive universal constants C_1 and C_2 such that, for every $t \geq 0$, the following holds with probability at least $1 - 2\exp(-C_2t^2)$, where $\epsilon = \sigma(C_1\sqrt{d/n} + t/\sqrt{n})$: $\|\mathbf{M}^\top\mathbf{M}/n - \mathbf{I}_d\| \leq \max\{\epsilon, \epsilon^2\}$.

The detailed proof can be found in [72]. We should note the condition in Lemma 2.1.9 is not hard to satisfy in OL2R: at every round, the ranker is serving a potentially distinct query; and even for the same query, different documents might be returned at different times. This gives the ranker a sufficient chance to collect informative observations for model estimation.

Regret Analysis for Linear Ranker

Based on Lemma 2.1.9, we have the following lemma, which provides a tight upper bound of the probability that the model's estimation of the pairwise preference is an uncertain rank order.

Lemma 2.1.10. At round $t \geq t'$, with $\delta_1 \in (0, \frac{1}{2})$, $\delta_2 \in (0, \frac{1}{2})$, $\beta \in (0, \frac{1}{2})$, and C_1, C_2 defined in Lemma 2.1.9, under event E_t, the following holds with probability at least $1 - \delta_2$: $\forall (i,j) \in [V_t]^2$, $\mathbb{P}\big((i,j) \in \omega_u^t\big) \leq \frac{8k_\mu^2\|\mathbf{x}_{ij}^t\|^2_{\mathbf{A}_t^{lin-1}}}{(1-2\beta)c_\mu^2\Delta_{\min}^2}\log\frac{1}{\delta_1}$, with $t' = \Big(\frac{c_1\sqrt{d}+c_2\sqrt{\log(\frac{1}{\delta_2})}+abd\sqrt{\frac{o_{\max}u^2}{d\lambda}}}{\lambda_{\min}(\mathbf{\Sigma})}\Big)^2 + \frac{2ab\log(1/\delta_1^2)+8a\lambda Q^2-\lambda}{\lambda_{\min}(\mathbf{\Sigma})}$, where $\Delta_{\min} = \min_{t\in T,(i,j)\in[V_t]^2}|\sigma(\mathbf{x}_{ij}^{t\top}\boldsymbol{\theta}^*) - \frac{1}{2}|$ representing the smallest gap of pairwise difference between any pair of documents associated to the same query over time (across all queries), $a = 4k_\mu^2u^2/(\beta^2c_\mu^2\Delta_{\min}^2)$, and $b = R^2 + 4\sqrt{\lambda}QR$,

Proof Sketch. According to the definition of certain rank order, a pairwise estimation $\sigma(\mathbf{x}_{ij}^{t\top}\widehat{\boldsymbol{\theta}})$ is certain if and only if $|\sigma(\mathbf{x}_{ij}^{t\top}\widehat{\boldsymbol{\theta}}) - 1/2| \geq \alpha_t\|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{lin-1}}$. By the reverse triangle inequality, the probability can be upper bounded by $\mathbb{P}\big(\big|\sigma(\mathbf{x}_{ij}^{t\top}\widehat{\boldsymbol{\theta}}) - \sigma(\mathbf{x}_{ij}^{t\top}\boldsymbol{\theta}^*)\big| - \big|\sigma(\mathbf{x}_{ij}^{t\top}\boldsymbol{\theta}^*) - 1/2\big|\big| \geq \alpha_t\|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{lin-1}}\big)$, which can be

further bounded by Theorem 1 in [14]. The key in this proof is to obtain a tighter bound on the uncertainty of PairRank's parameter estimation compared to the bound determined by δ_1 in Lemma 2.1.3, such that its confidence interval on a pair of documents' comparison at round t will exclude the possibility of flipping their ranking order, i.e., the lower confidence bound of this pairwise estimation is above 0.5. □

In each round of result serving, as the linear model $\widehat{\boldsymbol{\theta}}_t$ would not change until next round, the expected number of uncertain rank orders, denoted as $N_t = |\mathcal{E}_u^t|$, can be estimated by the summation of the uncertain probabilities over all possible pairwise comparisons under the current query q_t, e.g., $\mathbb{E}[N_t] = \frac{1}{2}(\sum_{(i,j)\in[V_t]^2} \mathbb{P}((i,j) \in \omega_u^t)$. Denote p_t as the probability that the user examines all documents in π_t at round t, and let $p^* = \min_{1\leq t\leq T} p_t$ be the minimal probability that all documents in a query are examined over time. We present the upper regret bound as follows.

Theorem 2.1.11. Assume pairwise query-document feature vector $\mathbf{x}_{ij}^t$ under query q_t, where $(i,j) \in [V_t]^2$ and $t \in [T]$, satisfies Proposition 1. With $\delta_1 \in (0, \frac{1}{2})$, $\delta_2 \in (0, \frac{1}{2})$, $\beta \in (0, \frac{1}{2})$, the T-step regret of the proposed model with a linear ranker is upper bounded by:

$$R_T \leq R' + (1-\delta_1)(1-\delta_2)p^{*-2}\left(2adV_{\max}\log(1+\frac{o_{\max}Tu^2}{2d\lambda}) + aw\right)^2$$

where $R' = t'V_{\max}^2 + (T-t')\big(\delta_2 V_{\max}^2 - (1-\delta_2)\delta_1 V_{\max}^2\big)$, with t' and a defined in Lemma 2.1.10, and $w = \sum_{s=t'}^{T}((V_{\max}^2 - 2V_{\max})u^2/(\lambda_{\min}(\Sigma_s))$, and $V_{\max}$ representing the maximum number of document associated to the same query over time. By choosing $\delta_1 = \delta_2 = 1/T$, we have the expected regret at most $R_T \leq O(d\log^4(T))$.

Proof Sketch. The regret is first decomposed into two parts: R' represents the regret when either E_t or Lemma 2.1.10 does not hold, in which the regret is out of our control, and we use the maximum number of pairs associated to a query over time, $L_{\max}$ to compute the regret. The second part corresponds to the cases when both events happen. Then, the instantaneous regret at round t can be bounded by

$$r_t = \mathbb{E}\big[K(\pi_t, \pi_t^*)\big] = \sum\nolimits_{i=1}^{d_t} \mathbb{E}\big[\frac{(N_i^t+1)N_i^t}{2}\big] \leq \mathbb{E}\big[\frac{N_t(N_t+1)}{2}\big] \tag{2.1.4}$$

where N_i^t denotes the number of uncertain rank orders in block $\mathcal{B}_i^t$ at round t, and N_t denotes the total number of uncertain rank orders. From the last inequality, it follows that in the worst case where the N_t uncertain rank orders are placed into the same block and thus generate at most $(N_t^2 + N_t)/2$ mis-ordered pairs with random shuffling. This is because based on the blocks, with N_t uncertain rank orders in one block, this block can at most have $N_t + 1$ documents. Then, the cumulative number of mis-ordered pairs can be bounded by the probability of observing uncertain rank orders in each round, which shrinks rapidly with more observations over time. □

Remark 2.1.12. By choosing $\delta = 1/T$, the theorem shows the expected regret increases at a rate of $\mathcal{O}(\log^4 T)$. In this analysis, we provide a gap-dependent regret upper bound of PairRank, where the gap $\Delta_{\min}$ characterizes the hardness of sorting the V_t candidate documents at round t. As the matrix M_t only contains information from observed document pairs, we adopt the probability of a ranked list is fully observed to tackle with the partial feedback [43, 73], which is a constant independent of T.

Remark 2.1.13. As the matrix $\mathbf{A}_t^{lin}$ only contains information from examined document pairs, our algorithm guarantees that the cumulative pairwise regret of the examined documents until round t ($\{1 : o_s\}_{s=1}^t$) to be sub-linear, while the regret in the leftover documents ($\{o_s + 1 : V_s\}_{s=1}^t$) is undetermined. We adopt a commonly used technique that leverages the probability that a ranked list is fully examined to bound the regret on those unexamined documents [43, 44, 73]. This probability is a constant independent of T. It is worth noting that our algorithm does not need the knowledge of p^* for model learning or result ranking; it is solely used for the regret analysis to handle the partial observations. From a practical perspective, the ranking quality of documents ranked below o_s for $s \in [T]$ does not affect users' online experience, as the users do not examine them. Hence, if we only count regret in the examined documents, R_T does not need to be scaled by p^*

Remark 2.1.14. Our regret is defined over the number of mis-ordered pairs, which is the *first* pairwise regret analysis for an OL2R algorithm, to the best of our knowledge. As we discussed before, existing OL2R algorithms optimize their own metrics, which can hardly link to any conventional rank metrics. As shown in [65], most classical ranking evaluation metrics, such as ARP and NDCG, are based on pairwise document comparisons. Our regret analysis of PairRank connects its theoretical property with such metrics, which has been later confirmed in our empirical evaluations.

Regret Analysis for Neural Ranker.

Following the same logic in the regret analysis for linear ranker, we have the following lemma bounding the probability that an estimated rank order being uncertain for neural ranker.

Lemma 2.1.15. With η, m satisfying the same conditions in Lemma 2.1.6, with $\delta_1 \in (0, 1/2)$ defined in Lemma 2.1.6, and $\delta_2 \in (0, 1/2)$, such that for $t \geq t' = \mathcal{O}(\log(1/\delta_2) + \log(1/\delta_1))$, under event E_t, the following holds with probability at least $1 - \delta_2$:

$$\forall (i,j) \in \Psi_t, \mathbb{P}((i,j) \notin \omega_t) \leq \frac{C_u \log(1/\delta_1)}{(\Delta_{\min} - 2\epsilon(m))^2} \|\mathbf{g}_{ij}^t/\sqrt{m}\|^2_{\mathbf{A}_t^{neu-1}},$$

where $C_u = 8\nu^2 k_\mu^2 / c_\mu^2$ with k_μ and c_μ as the Lipschitz constants for the sigmoid function, $\Delta_{\min} = \min_{t \in T, (i,j) \in \Psi_t} |\sigma(h_{ij}) - \frac{1}{2}|$ represents the smallest gap of pairwise difference between any pair of documents under the same query over time.

Remark 2.1.16. With m satisfying the condition in Lemma 2.1.6, and setting the corresponding η and $J = \widetilde{\mathcal{O}}(TL/\lambda)$, $\epsilon(m) = O(1)$ can be achieved. More specifically, there exists a positive constant c such that $\Delta_{\min} - 2\epsilon(m) = c\Delta_{\min}$.

Lemma 2.1.15 gives us a tight bound for an estimated pairwise order being uncertain. Intuitively, it targets to obtain a tighter bound on the uncertainty of the neural model's parameter estimation compared to the bound determined by δ_1 in Lemma 2.1.6.

With the same definition of p_t, we present the upper regret bound as follows.

Theorem 2.1.17. With δ_1 and δ_2 defined in Lemma 2.1.6, 2.1.15, η, m satisfying the same conditions in Lemma 2.1.6, there exist positive constants $\{C_i^r\}_{i=1}^2$ that with probability at least $1 - \delta_1$, the T-step regret is bounded by:

$$R_T \leq R' + (C_1^r \log(1/\delta_1)\widetilde{d}\log(1 + TV_{\max}/\lambda) + C_2^r)(1 - \delta_2)/(\Delta_{\min}^2 p^*)$$

where $R' = t'V_{\max}^2 + (T - t')\delta_2 V_{\max}^2$, with t' and $V_{\max}$ defined in Lemma 2.1.15. By choosing $\delta_1 = \delta_2 = 1/T$, the expected regret is at most $O(\widetilde{d}\log^4(T))$.

Remark 2.1.18. In this analysis, we provide a gap-dependent regret upper bound, where the gap $\Delta_{\min}$ characterizes the intrinsic difficulty of sorting the V_t candidate documents at round t. Intuitively, when $\Delta_{\min}$ is small, e.g., comparable to the network's resolution $\epsilon(m)$, many observations are needed to recognize the correct rank order between two documents.

2.1.5 Experiments

In this section, we empirically compare our proposed models with an extensive list of state-of-the-art OL2R algorithms on two large public learning to rank benchmark datasets.

Experiment Setup

Datasets. We empirically compare our proposed PairRank with an extensive list of state-of-the-art OL2R algorithms on two public learning to rank benchmark datasets: 1) Yahoo! Learning to Rank Challenge dataset [11], which consists of 292,921 queries and 709,877 documents represented by 700 ranking features;

and 2) MSLR-WEB10K [74], which contains 10,000 queries, each having 125 assessed documents on average, and is represented by 136 ranking features. Both datasets are labeled with a five-grade relevance scale: from not relevant (0) to perfectly relevant (4). We followed the train/test/validation split provided in the datasets. Both quantitative and qualitative evaluations are performed to examine our proposed solution, especially its advantages over existing OL2R solutions in online learning efficiency.

User interaction simulation. We simulate user behavior via the standard setup for OL2R evaluations [10, 25] to make our reported results directly comparable to those in literature. First, at each time t, a query is uniformly sampled from the training set for result serving. Then, the model determines the ranked list and returns it to the users. The interaction between the user and the list is then simulated with a dependent click model [68], which assumes that the user will sequentially examine the list and make a click decision on each document. At each position, the user decides whether to click on the document or not, modeled as a probability conditioned on the document's relevance label, e.g, $\mathbb{P}(click = 1|\text{relevance grade})$. After the click, the user might stop due to his/her satisfaction with the result or continue to examine more results. The stop probability after a click is modeled as $\mathbb{P}(stop = 1|click = 1, \text{relevance grade})$. If there is no click, the user will continue examining the next position. We employ three different click model configurations to represent three different types of users. Basically, we have the *perfect* users, who click on all relevant documents and do not stop browsing until the last returned document; the *navigational* users, who are very likely to click on the first highly relevant document and stop there; and the *informational* users, who tend to examine more documents, but sometimes click on irrelevant ones, and thus contribute a significant amount of noise in their click feedback.

Table 2.1: Configuration of simulated click models.

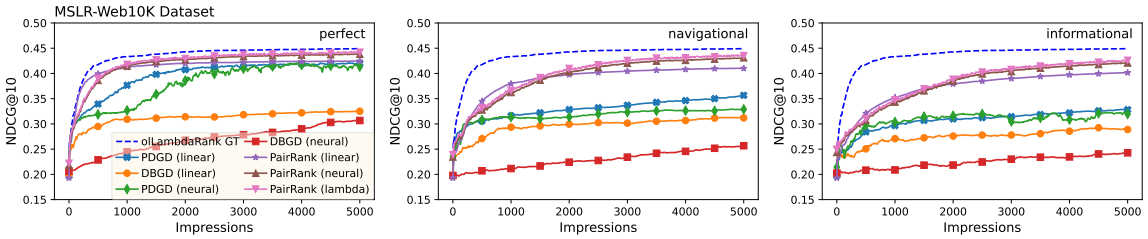
	Click Probability					Stop Probability				
R	0	1	2	3	4	0	1	2	3	4
per	0.0	0.2	0.4	0.8	1.0	0.0	0.0	0.0	0.0	0.0
nav	0.05	0.3	0.5	0.7	0.95	0.2	0.3	0.5	0.7	0.9
inf	0.4	0.6	0.7	0.8	0.9	0.1	0.2	0.3	0.4	0.5

Baselines. We list the OL2R solutions used for our empirical comparisons below. And we name our proposed model as PairRank (linear), PairRank (neural) to indicate different underlying ranking models. We also include PairRank (Lambda) in the experiment result discussions.

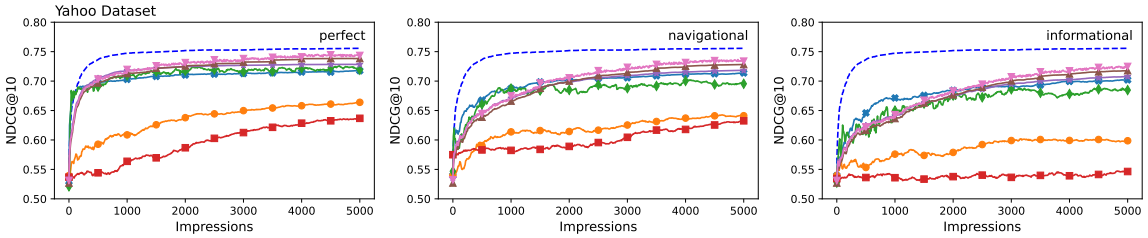
- **ϵ-Greedy** [36]: At each position, it randomly samples an unranked document with probability ϵ or selects the next best document based on the currently learned RankNet.
- **Linear-DBGD and Neural-DBGD** [12]: DBGD uniformly samples a direction from the entire model space for exploration and model update. We apply it to both linear and neural rankers.
- **Linear-PDGD and Neural-PDGD** [25]: PDGD samples the next ranked document from a Plackett-Luce model and estimates gradients from the inferred pairwise preferences. We also apply it to both linear and neural network rankers.
- **olLambdaRank GT**: At each round, we estimate a new LambdaRank model with ground-truth relevance labels of all the presented queries. This serves as the skyline in all our experiments.

Experiment Results

Offline evaluation. In offline evaluation, while the algorithm is learning, we evaluate the learned rankers on a separate testing set using its ground-truth relevance labels. We use Normalized Discounted Cumulative Gain at 10 (NDCG@10) to assess different algorithms' ranking quality. We can clearly observe that both linear and neural PairRank achieved significant improvement compared to all baselines. Under different click models, both linear and neural DBGD performed the worst. This is consistent with previous findings: DBGD depends on interleave tests to determine the update direction in the model space. But such model-level feedback cannot inform the optimization of any rank-based metric. Moreover, with a neural ranker, random exploration becomes very ineffective. PDGD consistently outperformed DBGD under different click models. However, its document sampling based exploration limits its learning efficiency, especially when users only examine a small



(a) Offline performance (NDCG@10) on the MSLR-WEB10K dataset.



(b) Offline performance (NDCG@10) on the Yahoo! dataset.

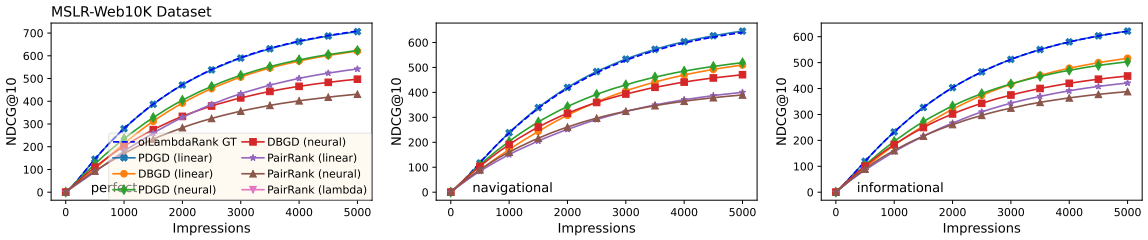
Figure 2.4: Offline ranking performance on two different datasets under three different click models.

portion of documents, e.g., the navigational users. It is worth noting that in the original paper [25], PDGD with a neural ranker outperformed linear ranker after much more interactions, e.g., 20000 iterations. Our proposed solutions with only 5000 iterations already achieved better performance than the best results reported for PDGD, which demonstrates the encouraging efficiency of our proposed OL2R solution. Compared to linear PairRank, the neural PairRank had a worse performance at the beginning. We attribute it to the limited training samples available at the initial rounds, i.e., the network parameters were not well estimated yet. But the neural model enables non-linear relation learning and quickly leads to better performance than the linear models when more observations arrive. Compared to neural PairRank, olLambdaRank directly optimizes the evaluation metrics, e.g., NDCG@10, with corresponding gradients. We can observe similar improvements from LambdaRank compared to RankNet as previously reported in offline settings. As a brief summary, our solution showed much faster convergence compared to all baselines. This directly leads to much better online performance improvement. We attribute this advantage to our uncertainty based exploration: it only explores when the ranker's estimation on a pair of documents is uncertain; for the certain pairs of documents, they are already in the correct ranking order for the ranker to directly return.

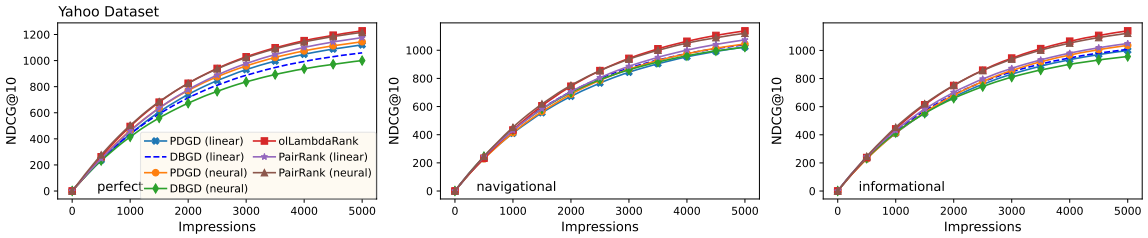
Online evaluation. In OL2R, in addition to the offline evaluation, the models' ranking performance during online optimization should also be considered, as it reflects user experience during model update. Sacrificing users experience for model training will compromise the goal of OL2R. We adopt the cumulative Normalized Discounted Cumulative Gain to assess models' online performance. For T rounds, the cumulative NDCG is calculated as

$$\text{Cumulative NDCG} = \sum_{t=1}^{T} \text{NDCG}(\pi_t) \cdot \gamma^{(t-1)},$$

which computes the expected reward a user receives with a probability γ that he/she stops searching after each query [25]. Following the previous work [10, 19, 25], we set $\gamma = 0.9995$. Figure 2.5 shows the online performance of the proposed online neural ranking model and all the other baselines. It is clear to observe that DBGD-based models have a much slower convergence and thus have worse online performance. Compared to the proposed solution, PDGD showed consistently worse performance, especially under the navigational and informational click models with a neural ranker. We attribute this difference to the exploration strategy used in PDGD: PDGD's sampling-based exploration can introduce unwanted distortion in the ranked results, especially at the early stage of online learning. We should note in cumulative NDCG ranking performance in the earlier stages plays a much more important role due to the strong shrinking effect of γ. Our proposed linear and neural PairRank demonstrated significant improvements over all baseline methods on both datasets under three different click models. Such improvement indicates the effectiveness our uncertainty based exploration, which only explores when the ranker's pairwise estimation is uncertain. Its advantage becomes more apparent



(a) Online performance (cNDCG@10) on the MSLR-WEB10K dataset.



(b) Online performance (cNDCG@10) on the Yahoo! dataset.

Figure 2.5: Online ranking performance on two different datasets under three different click models.

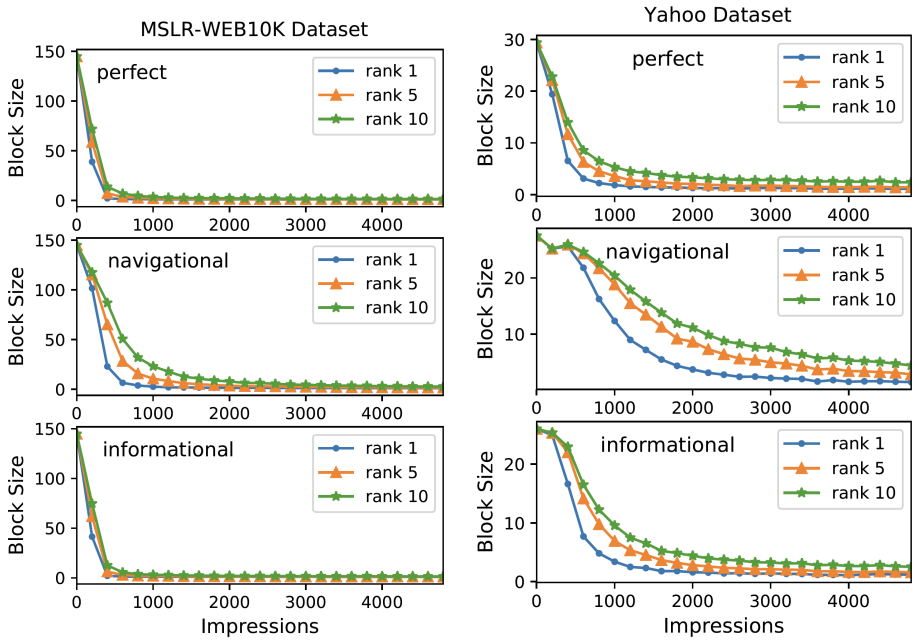
in this online ranking performance comparison, as an overly aggressive exploration in the early stage costs more in cumulative NDCG. We can also observe the improvement of olLambdaRank compared to neural PairRank in this online evaluation, although the difference is not very significant. The key reason is also the strong discount applied to the later stage of model learning: olLambdaRank's advantage in directly optimizing the rank metric becomes more apparent in the later stage, as suggested by the offline performance in Figure 2.4. At the beginning of model learning, both models are doing more explorations and therefore the online performance got more influenced by the number of document pairs with uncertain rank orders, rather than those with certain rank orders.

Zoom into PairRank.

To further verify the effectiveness of the exploration strategy devised in , we zoom into the trace of its block size across queries during the online model update. As PairRank uses random shuffling within each block for exploration, a smaller block size, especially at the top-ranked positions, is preferred to reduce regret. Figure 2.6 shows the size of document blocks at rank position 1, 5 and 10.

First, we can clearly observe that after hundred rounds of interactions, the sizes of blocks quickly converge to 1, especially at the top-ranked positions. This confirms our theoretical analysis about PairRank's block convergence. And by comparing the results across different click models, we can observe that the block size converges slower under the navigational click model. Similar trends can be observed in Figure 2.7b. In this figure, the number of blocks is calculated by averaging in every 200 iterations to reduce variance. We can observe that at the early stages, PairRank under the navigational click model has fewer blocks (hence more documents in one block), which indicates a higher uncertainty in model estimation. The key reason is that much fewer clicks can be observed in each round under the navigational click model, as the stop probability is much higher, i.e., stronger position bias. As a result, more interactions are needed to improve model estimation. For the same reason, in Figure 2.6, the block size at rank position 10 shrinks much slower than that at other positions also suggests position bias is the main factor that slows down the learning of PairRank.

In Figure 2.7a, we show the block size at each rank position in PairRank under the informational click model at different time points. One interesting finding is the blocks in the middle ranks, i.e., rank 300 - 400, tend to have more documents, while for the ranks at both the top and the bottom, the corresponding blocks tend to have fewer documents. We attribute it to the pairwise learning in PairRank, where the uncertainty is calculated on the pairwise preferences, and thus it is easy to identify the document pairs with greater differences.



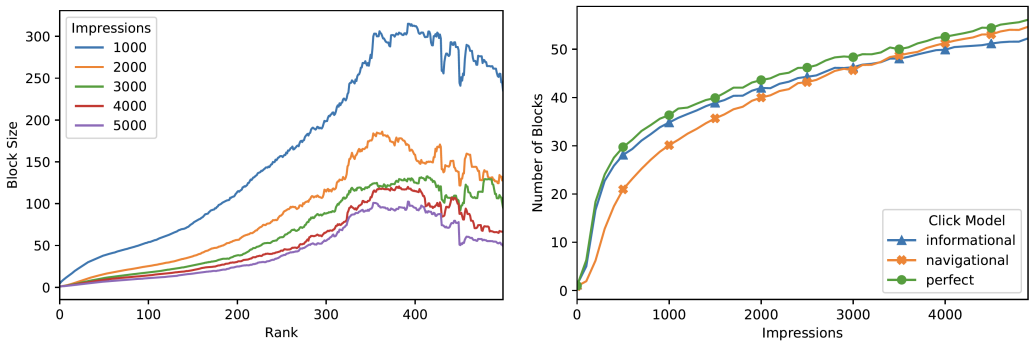
(a) MSLR-WEB10K Dataset (b) Yahoo Dataset

Figure 2.6: The size of blocks at top ranks.

2.2 Scalable Exploration with Perturbed Feedback.

Deep neural networks demonstrates significant advantages in improving ranking performance in retrieval tasks. Our PairRank with neural ranker provides encouraging performance improvement compared to PairRank (linear). Despite being theoretically sound and empirically effective, PairRank (neural)'s limitation is also remarkably serious: its computational cost for performing the required exploration is prohibitively high (almost *cubic* to the number of neural network's parameters). In particular, to quantify the uncertainty of its estimated pairwise preferences among candidate documents, it has to maintain a high-probability confidence set for the current ranker's parameter estimation over time. However, the construction of the confidence set depends on the dimensionality of the neural network's parameters: as required by the neural tangent kernel, an inverse of the covariance matrix computed based on the gradient of entire neural network is needed whenever the network is updated. For example, for a simple two layer feed-forward neural network with input dimension d and m neurons in each layer, the size of the covariance matrix is $(md + m^2 + m)^2$. The best known time complexity for computing the inverse of this covariance matrix is $O\big((md + m^2 + m)^{2.373}\big)$, by the optimized Coppersmith–Winograd algorithm [75]. This computational complexity quickly outpaces the limit of any modern computational machinery, given m or d are usually very large in practice (e.g., m is often in hundreds and d in tens of thousands) and this matrix inverse operation is needed in every round when the neural network is updated. Due to this limitation, PairRank (neural) has to employ the diagonal approximation of the covariance matrix in its actual implementations. But such a diagonal approximation loses its all theoretical guarantees, which unfortunately leads to a gap between the theoretical and empirical performance of PairRank (neural). And even how this gap would depend on the dimensionality of the network and affect PairRank (neural) performance is completely unknown. This inevitably limits the application of the neural OL2R solutions in practice.

In this work, we propose an efficient and scalable exploration strategy for PairRank by eliminating its explicit confidence set construction. The basic idea is to use bootstrapping technique to measure the uncertainty of a ranker's output via a set of sample estimates. In particular, we maintain N rankers in parallel. And in each round, after receiving the user's click feedback, each of the rankers is updated with the observed clicks and independently generated pseudo noise from a zero-mean Gaussian distribution. The overall model's estimation uncertainty on a pair of documents is then determined by an ensemble of the estimates from all N rankers. For example, for a document pair (i, j), if all N rankers predict $i \succ j$, then (i, j) is considered as in a certain rank



(a) MSLR-WEB10K with informational click model.

(b) Number of blocks identified on MSLR-WEB10K dataset

Figure 2.7: Qualitative analysis of blocks in PairRank (linear)

order, otherwise it is considered as in an uncertain rank order. Besides regular neural network updates, no additional computation is needed, which greatly reduces the computational overhead as required in PairRank. We name our new solution as Perturbed PairRank (or PePairRank in short). We rigorously prove that with a high probability PePairRank obtains the same regret as PairRank, but the computational complexity is way much lower. In addition, as no approximation is needed in PePairRank, its theoretical analysis directly suggests its empirical performance. Our extensive empirical evaluations demonstrate the strong advantage in both efficiency and effectiveness of PePairRank against PairRank and a rich set of state-of-the-art solutions over two OL2R benchmark datasets on standard retrieval metrics.

2.2.1 Related Work

Randomized exploration in online learning

Efficient exploration is critical for online algorithms, as the model learns by actively acquiring feedback from the environment [76]. Distinct from the deterministic exploration strategies, such as upper confidence bound [14,77], randomization-based exploration enjoys advantages in its light computational overhead and thus has received increasing attention in online learning community. The most straightforward randomization-based exploration strategy is ϵ-greedy [77], which takes the currently estimated best action with probability $1-\epsilon$, otherwise randomly take an action. It has been applied in OL2R in [36]. Almost no additional computation is needed in ϵ-greedy for exploration, but the exploration is also independent from the current model estimation and therefore can hardly be optimal in practice. More advanced randomization-based exploration strategies are built on the bootstrapping technique in statistics. Giro [78] explores by updating a model with a bootstrapped sample of its history with pseudo reward. In [79, 80], random noise is added to the observed feedback for the model training to achieve exploration in model's output. Such a strategy is proved to be effective in both linear and generalized linear model training. Most recently, Jia et al. [81] proved randomization can also be used for online neural network learning. The most closely related work to our study is [82, 83], where an ensemble of models are trained to approximate the confidence interval for the purpose of exploration in online model update.

2.2.2 Bootstrapping with Perturbed Feedback

Discussed in previous sections, PairRank (linear) and PairRank (neural) have a strong theoretical foundation. However, its scalability is severely limited due to the additional computation required for constructing the confidence interval. In particular, the covariance matrix $\mathbf{A}_t^{lin}$ in Lemma 2.1.3 is constructed with the feature vectors of the observed instance, and $\mathbf{A}_t^{neu}$ in Lemma 2.1.6 is constructed with the gradient of the scoring function with respect to the network parameters. In order to construct the confidence interval, the inverse of the covariance matrix $\mathbf{A}_t^{lin}$, $\mathbf{A}_t^{neu}$ has to be computed whenever the model is updated, which results in an unacceptably high computational cost (around $O(p^3)$ for neural ranker with p as the number of parameters in the neural network). As a consequence, it is practically impossible for PairRank to be exactly executed, especially for PairRank with neural ranker. In practice, approximation can be employed to make PairRank (neural) operational, e.g., using the diagonal approximation of $\mathbf{A}_t^{neu}$. However, there is no theoretical guarantee

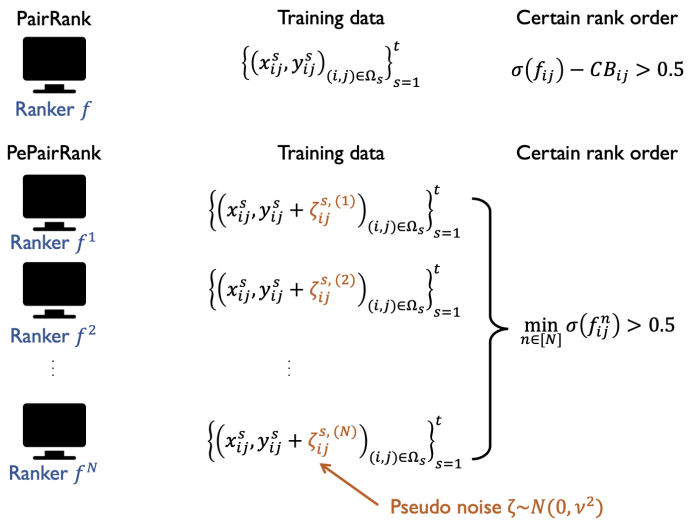


Figure 2.8: Comparison between PairRank and PePairRank

for such an approximation, which unfortunately breaks the theoretical promise of PairRank and directly leads to an unknown gap between its theoretical and empirical performance.

To bridge the gap, we develop an efficient and scalable strategy for recognizing the certain and uncertain rank orders without explicitly constructing the confidence set. And our basic idea is to leverage the bootstrapping technique to create randomness in a ranker's output. In particular, at each round, we perturb the entire user feedback history for N times with noise freshly and independently sampled from a zero-mean Gaussian distribution, and train the corresponding neural ranker as usual. Denote model $\boldsymbol{\theta}^{(n)}$ for $n \in [N]$ as the solution of minimizing the following objective function with gradient descent,

$$\begin{aligned}\boldsymbol{\theta}^{(n)} = \min \sum_{s=1}^{t} \sum_{(i,j)\in\Omega_s} & -\big(1 - (y_{ij}^s + \zeta_{ij}^{s,(n)})\big) \log\big(1 - \sigma(f_{ij})\big) \\ & - (y_{ij}^s + \zeta_{ij}^{s,(n)}) \log\big(\sigma(f_{ij})\big) + m\lambda/2 \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2, \end{aligned} \tag{2.2.1}$$

where $\{\zeta_{ij}^{s,(n)}\}_{s=1}^t \sim \mathcal{N}(0, \nu^2)$ are Gaussian random variables that are independently sampled in each round t, and ν is a hyper-parameter that controls the strength of perturbation (and thus the exploration) in PePairRank. The difference between PePairRank and PairRank is shown in Figure 2.8.

The detailed procedure of PePairRank is given in Algorithm 2. The algorithm starts by initializing the N neural rankers. At each round of interaction, given a query q_t, for each pair of candidate documents, N parallel predictions about their rank order will be generated by the set of neural rankers. If all the N estimations give the same prediction about the document pair's rank order, e.g., $i \succ j$ for document i and document j, then (i, j) is considered as a certain rank order (line 9 - line 13 in Algorithm 2). Otherwise, the relation between these two documents is still uncertain and further exploration is needed there when generating the ranked list. Once the sets of certain and uncertain rank orders are determined, we follow the same procedure of PairRank to generate the ranked list.

The key intuition for PePairRank is to utilize the variance introduced in the randomly perturbed click feedback to encourage exploration. With the injected perturbation, there are two kinds of deviations existing in the estimated pairwise preference in each of the N parallel neural rankers: 1) the deviation caused by the observation noise introduced by the click feedback; 2) the deviation caused by the added perturbations. By properly setting the variance parameter ν for the added perturbation, the corresponding deviation will help "correct" any deviation caused by the observation noise (e.g., insufficient observations might lead to inaccurate predictions in such models), and guarantee that the final estimation is optimistic with a constant probability

Algorithm 2 Perturbed PairRank (PePairRank)

Inputs: Number of rounds T, regularization coefficient λ, perturbation parameter ν, network width m, network depth L, and number of rankers N
Initialize N neural network models $\{\boldsymbol{\theta}_0^n\}_{n=1}^N$ with m and L
for $t = 1$ to T **do**
  $S_t^c = \emptyset, S_t^u = \emptyset$
  $q_t \leftarrow$ receive_query(t)
  $\mathcal{X}_t = \{\mathbf{x}_1^t, \mathbf{x}_2^t, \dots \mathbf{x}_{V_t}^t\} \leftarrow$ retrieve_candidate_documents(q_t)
  for each document pair $(i, j) \in [V_t]^2$ **do**
    $\{\sigma(f_{ij,t}^n)\}_{n=1}^N \leftarrow$ get_N_estimations($\mathbf{x}_i^t$, $\mathbf{x}_j^t$, $\{\boldsymbol{\theta}_t^n\}_{n=1}^N$)
    if $\min_{n\in[N]} \sigma(f_{ij,t}^n) > 1/2$ or $\max_{n\in[N]} \sigma(f_{ij,t}^n) < 1/2$ **then**
      $S_t^c \leftarrow S_t^c \cup (i, j)$
    else
      $S_t^u \leftarrow S_t^u \cup (i, j)$
  $\pi_t \leftarrow$ topological_sort(S_t^c, S_t^u)
  $C_t \leftarrow$ collect_click_feedback(π_t)
  $\Omega_t, \{y_{ij}\}_{(i,j)\in\Omega_t} \leftarrow$ construct_training_data(C_t)
  for $n = 1, ..., N$ **do**
    Generate $\{\{\zeta_{ij}\}_{(i,j)\in\Omega_s}\}_{s=1}^t \sim \mathcal{N}(0, \nu^2)$
    Set $\boldsymbol{\theta}_t^n$ by the output of gradient descent for solving Eq (2.2.1) with $\{\Omega_s\}_{s=1}^t$.

(i.e., the Optimism in the Face of Uncertainty principle [14]). On top of this perturbation, maintaining N models guarantees that the maximum of these estimations is optimistic with a high probability.

Compared to PairRank, which requires to maintain the inverse of the covariance matrix, PePairRank does not need any added computation for the purpose of exploration, besides the regular neural network updates. As a result, PePairRank greatly alleviates the computation burden in PairRank. More specifically, PairRank generally takes the following steps: (1) predict the rank order in each document pair, (2) generate the ranked list with divide-and-conquer with respect to the constructed certain and uncertain rank orders, and (3) update the model according to the newly received click feedback. With p representing the total number of parameters in a neural ranker, PairRank has the time complexity $\mathcal{O}(V_t p + V_t^2)$ for the first step. As the N neural models in PePairRank are independent from each other, the time complexity of PePairRank in the first step is also $\mathcal{O}(V_t p + V_t^2)$ by executing the N ranker's predictions in parallel. For the third step, again by training the N neural rankers in parallel using gradient descent, both PairRank and PePairRank have the time complexity of $\mathcal{O}(Jp\sum_{s=1}^t |\Omega_s|)$ where J is the number of epochs for training the neural network. The key difference lies in the second step. PairRank requires the computation of the inverse covariance matrix, which has the time complexity at least $\mathcal{O}(p^{2.2373})$. Besides, constructing the confidence interval for all the document pairs has the time complexity of $\mathcal{O}(V_t^2 p^2)$. While for PePairRank, finding the minimum of the N predictions for all the document pairs costs $\mathcal{O}(NV_t^2)$. Once the certain and uncertain rank orders are determined, both algorithms require $\mathcal{O}(V_t + E_t)$ for the topological sort, where E_t represents the number of certain rank orders and $E_t \leq V_t$. Therefore, for the second step, PairRank has the total time complexity as $\mathcal{O}(p^{2.2373} + V_t^2 p^2 + V_t + E_t) = \mathcal{O}(p^{2.2373} + V_t^2 p^2)$, while PePairRank has the time complexity as $\mathcal{O}(NV_t^2 + V_t + E_t) = \mathcal{O}(NV_t^2)$. As p is oftentimes in the order of tens of thousands (if not less), PePairRank greatly reduces the time required for performing exploration in neural PairRank. And also empirically, the number of parallel rankers N in PePairRank does not need to be large. For example, in our experiments, we found $N = 2$ already led to promising performance of PePairRank comparing to PairRank.

2.2.3 Regret Analysis

In this section, we provide the regret analysis of the proposed exploration strategy. For better readibility, we present the analysis of a linear ranker. According to our previous analysis for PairRank (neural), under the neural tangent technique and the convergence analysis of the gradient descent in neural network optimization,

the linear analysis can be readily applied to the neural ranker. And we discuss the difference between the analysis between the linear ranker and neural ranker at the end of this section.

Follow the standard assumption in previous sections, we assume that on the examined documents where $\pi_t(i) \leq o_t$, the obtained feedback C_t is independent from each other given the true relevance of documents, so is their noise [5, 67, 68]. Therefore, the noise in the inferred preference pair becomes the sum of noise from the clicks in the two associated documents. And we also only use the independent pairs to construct Ω_t as suggested in PairRank. Thus, the pairwise noise satisfies Proposition 2.1.2.

With $f(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{x}^\top \boldsymbol{\theta}$ as the scoring function for a linear ranker, the loss function can be re-written as,

$$\mathcal{L}_t^{(n)}(\boldsymbol{\theta}) = \sum_{s=-d+1}^{t} \sum_{(i,j)\in\Omega_s^{ind}} -\big(y_{ij}^s + \zeta_{ij}^{s,(n)}\big) \log(\sigma({\mathbf{x}_{ij}^s}^\top \boldsymbol{\theta})) - \big(1 - (y_{ij}^s + \zeta_{ij}^{s,(n)})\big) \log\big(1 - \sigma({\mathbf{x}_{ij}^s}^\top \boldsymbol{\theta})\big),$$

where d is the dimension of the feature vectors. With $|\Omega_s| = 1$, $\mathbf{x}_{ij}^s = \sqrt{\lambda}\mathbf{e_i}$, $y_{ij}^s = 0$ for $s \in [-d+1, 0]$, this loss function can be interpreted as adding l_2 regularization to the cross-entropy loss.

By optimizing the objective, we can have the following lemmas to bound the deviations caused by different sources of noise.

Lemma 2.2.1. (Deviation from observation noise). At round $t < T$, for any pair of document $(\boldsymbol{x}_i^t, \boldsymbol{x}_j^t)$ under query q_t, with probability at least $1 - \delta$, we have,

$$|\sigma({\mathbf{x}_{ij}^t}^\top \bar{\boldsymbol{\theta}}_t) - \sigma({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}_t^*)| \leq \alpha_t \|\mathbf{x}_{ij}^t\|_{\boldsymbol{A}_t^{-1}},$$

where $\alpha_t = (2k_\mu/c_\mu)\big(\sqrt{R^2 \log(\det(\boldsymbol{A}_t)/(\delta^2 \det(\lambda \mathbf{I})))} + d\big)$, $\boldsymbol{A}_t = \lambda \mathbf{I} + \sum_{s=1}^{t-1} \sum_{(i'j')\in\Omega_s} \boldsymbol{x}_{i'j'} \boldsymbol{x}_{i'j'}^\top$, k_μ is the Lipschitz constant of the sigmoid link function σ, $c_\mu = \inf_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \dot{\sigma}(\boldsymbol{x}^\top \boldsymbol{\theta})$, with $\dot{\sigma}$ as the first derivative of σ.

Accordingly, we define E_t as the success event at round t:

$$E_t = \big\{\forall (i,j) \in [V_t]^2, |\sigma({\mathbf{x}_{ij}^t}^\top \bar{\boldsymbol{\theta}}_t) - \sigma({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}^*)| \leq \alpha_t \|\mathbf{x}_{ij}^t\|_{\boldsymbol{A}_t^{-1}}\big\}.$$

Intuitively, E_t is the event that the auxiliary solution $\bar{\boldsymbol{\theta}}_t$ is "close" to the optimal model $\boldsymbol{\theta}^*$ at round t.

As discussed before, we define the regret as the number of mis-ordered pairs. Therefore, the key step in regret analysis is to quantify the probability that an estimated preference is uncertain. According to Algorithm 2, the certain rank order in the perturbed pairwise ranker is defined as follows,

Definition 2.2.2. (Certain Rank Order) At round t, the rank order between documents $(i,j) \in [V_t]^2$ belongs to the set of certain rank orders ω_t^c if and only if $\min_{n\in[N]} \sigma\left(f_{ij}^{t,(n)}\right) > \frac{1}{2}$ or $\max_{n\in[N]} \sigma\left(f_{ij}^{t,(n)}\right) < \frac{1}{2}$; otherwise, $(i,j) \in \omega_t^u$.

According to the definition, and the deviations caused by the observation noise and the pseudo noise, we have the following lemma quantifying the probability of an estimation being uncertain.

Lemma 2.2.3. There exist positive constants c, C_1 and C_2, that with $t' = \left(\frac{C_1\sqrt{d} + C_2\sqrt{\log(1/\delta)} + (P^2 R k_\mu)/(\sqrt{\lambda} c_\mu \Delta_{\min})}{\lambda_{\min}(\Sigma)}\right)^2 + \frac{2k_\mu P}{c_\mu \Delta_{\min}}\left(\sqrt{R^2 \log(1/\delta)} + \sqrt{\lambda} Q\right)$, $\delta \in (0,1)$, for round $t \geq t'$, with probability at least $1 - \delta$, event E_t holds with α_t defined in Lemma 2.2.1, with $N \geq \log \delta / \log(1 - \exp(-\beta^2)/(4\sqrt{\pi}\beta))$, where $\beta = \frac{k_\mu^2 \alpha_t^2}{c_\mu^2 \nu^2}$, for a document pair (i,j) that $i \succ j$ for the given query, the probability that the estimated pairwise preference is uncertain is upper bounded as $\mathbb{P}\big((i,j) \in \omega_t^u\big) \leq \frac{2N\nu^2 k_\mu^2 \|\mathbf{x}_{ij}^t\|^2_{\mathbf{A}_t^{-1}}}{c_\mu^2 c^2 \Delta_{\min}^2}$, where $\Delta_{\min} = \min_{t\in T, (i,j)\in[V_t]^2} |\sigma({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}^*) - \frac{1}{2}|$

representing the smallest gap of pairwise difference between any pair of documents associated to the same query over time (across all queries).

The detailed proof is provided in the appendix. This lemma provides the upper bound of the probability that an estimated pairwise preference is uncertain. The key idea is to analyze the concentration and anti-concentration property of the deviation caused by the pseudo noise. In particular, the deviation caused by the pseudo noise γ, and the ensemble of N rankers should be sufficiently large so that for document pairs (i, j), the maximum estimated pairwise preference, $\max_{n\in[N]} \sigma(\mathbf{x}_{ij}^{t\top} \widehat{\boldsymbol{\theta}}_t^{(n)})$ is optimism to trigger exploration. On the other hand, with more observations, the probability of being uncertain will be shrinking with the concentration property of the pseudo noise.

Following the assumption in [26, 43, 84], denote p_t as the probability that the user examines all documents in τ_t at round t, and let $p^* = \min_{1\le t\le T} p_t$ be the minimal probability that all documents in a query are examined over time. The regret of the proposed model can be upper bounded as follows.

Theorem 2.2.4. Assume pairwise query-document feature vector $\mathbf{x}_{ij}^t$ under query q_t, where $(i, j) \in [V_t]^2$ and $t \in [T]$, satisfies Proposition 1. With $\delta \in (0, 1)$, with probability at least $1 - \delta$, the T-step regret of the proposed model is upper bounded by:

$$R_T \le R' + \frac{1}{p^*} 2dV_{\max} C \log\left(1 + \frac{o_{\max} T P^2}{2d\lambda}\right)$$

where $R' = t' V_{\max}$, $V_{\max}$ represents the maximum number of document associated with the same query over time, and t' is defined in Lemma 2.2.3, and $w = \sum_{s=t'}^{T} \left((V_{\max}^2 - 2V_{\max})P^2/\lambda_{\min}(\boldsymbol{A}_s)\right)$, and By choosing $\delta_1 = \delta_2 = 1/T$, we have the expected regret at most $R_T \le O(d \log^4(T))$.

We provide the detailed proof in the appendix. According to the pairwise exploration strategy, the regret only comes from the document pairs that are uncertain, e.g., random shuffling will be conducted to perform the exploration. With the quantified uncertain probability in Lemma 2.2.3, the pairwise regret can be upper bounded accordingly.

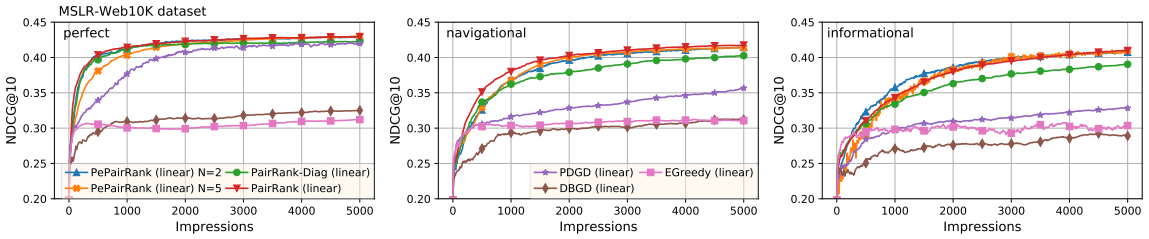
In neural rankers, the neural network approximation error should be considered in addition to the deviations caused by the noise. According to the analysis in [81], the variance of the added noise should be set according to the deviations caused by both the observation noise and the approximation error. Based on the theoretical analysis in [84], by properly setting the width of the neural network and the step size of gradient descent, the model with a neural ranker will still have a sublinear regret.

2.2.4 Experiment Results

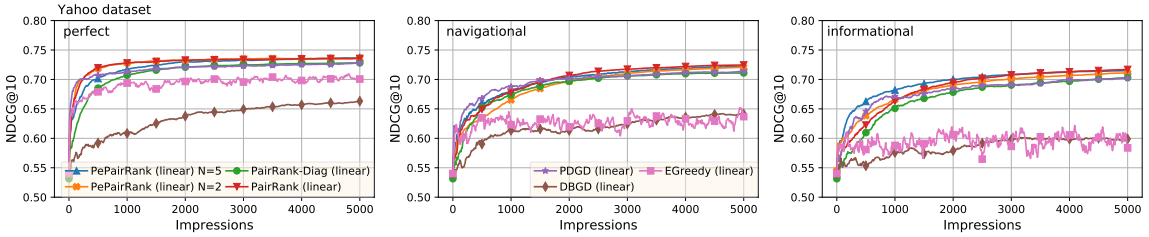
Online and offline performance.

We first compare our proposed model with the baselines using a linear ranker. The results are reported in Figure 2.9. For PePairRank (linear), we reported the best performance with $N = 2$ and $N = 5$. We can clearly observe PePairRank (linear) maintained PairRank's strong advantage over other OL2R solutions, including ϵ-Greedy, DBGD, and PDGD, in both online and offline evaluations across three click models. It is also obvious that a straightforward perturbation of model's output, i.e., ϵ-Greedy, basically led to the worst OL2R performance, although it is often the default choice for exploration in online learning.

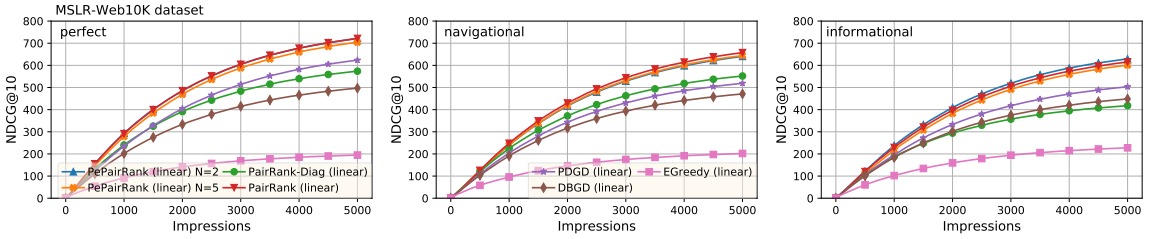
More importantly, PairRank with a diagonal approximated covariance matrix showed serious degradation in its ranking performance, especially its online performance. For example, on MSLR-Web10K dataset, PairRank with diagonal approximation was even worse than PDGD in the online evaluation under both perfect and informational click models. This means an approximated covariance matrix cannot accurately measure the ranker's estimation uncertainty. Furthermore, this inaccuracy's impact is not deterministic: under perfect feedback, the seriously degenerated online performance together with mild decrease in offline performance suggest the model over explored; but under informational feedback, both online and offline performance dropped, which suggests insufficient estimation. This demonstrates the complication of using approximations in OL2R solutions, which loses all theoretical guarantees in the original analysis. As a result, it also strongly



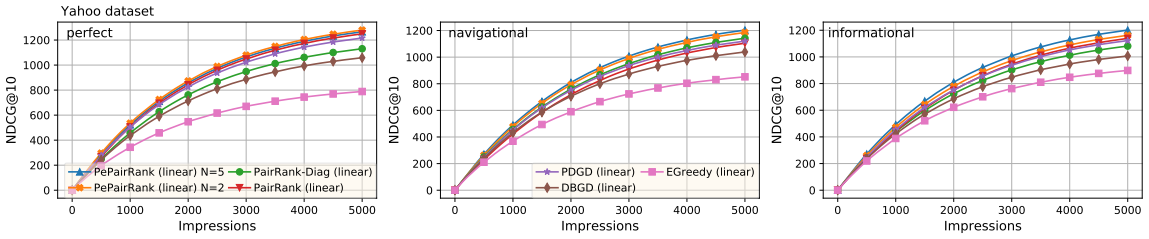
(a) Offline performance (NDCG@10) of linear rankers on the MSLR-WEB10K dataset.



(b) Offline performance (NDCG@10) of linear rankers on the Yahoo! dataset.



(c) Online performance (NDCG@10) of linear rankers on the MSLR-WEB10K dataset.



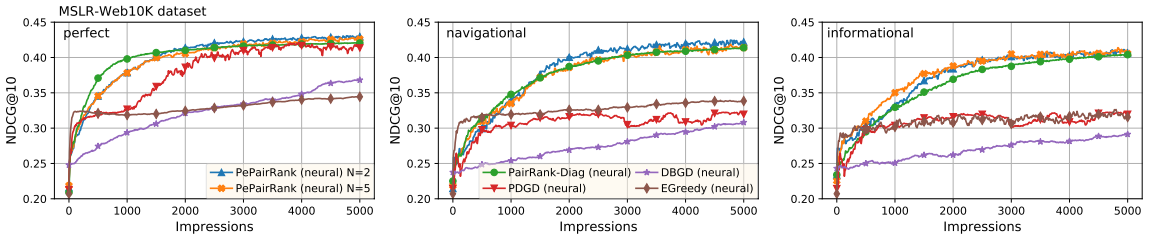
(d) Online performance (NDCG@10) of linear rankers on the Yahoo! dataset.

Figure 2.9: Offline and online ranking performance of linear rankers on two different datasets under three different click models.

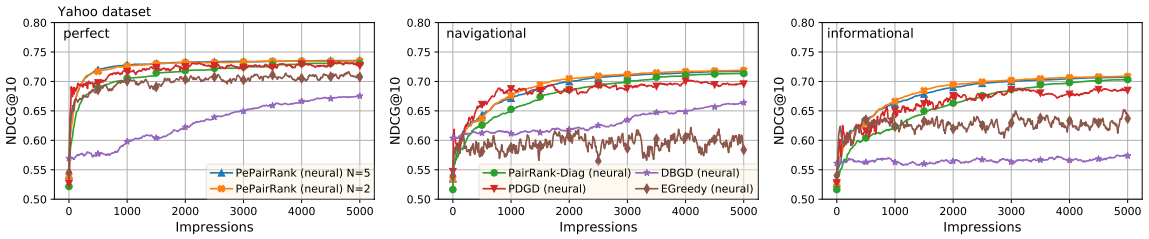
suggests olRankNet might not be optimal in practice, given the diagonal approximation employed to make its computation feasible. This will be demonstrated in our experiments next.

It is worth noting that PePairRank (linear) with $N = 2$ already exhibits faster convergence than PairRank, and simply increasing N does not necessarily further improve the model's performance. This result is very promising: as the only computational overhead in our perturbation-based exploration strategy is to estimate $N - 1$ additional rankers, the actual added cost in practice is minimum when $N = 2$. Similar observation is also obtained when applied to neural rankers.

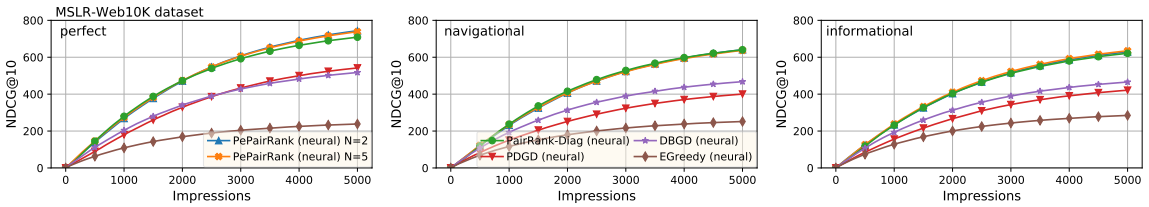
In Figure 2.10, we report the results obtained on the neural rankers. First of all, PairRank (neural) and PePairRank (neural) still showed significant improvement over other OL2R solutions, including ϵ-Greedy, DBGD, MGD and PDGD. This means the pairwise exploration implemented in PairRank (neural) is still effective for neural OL2R. The most important finding in this experiment is that PePairRank (neural) outperformed PairRank (neural), where $m = 100$ for the neural network structure. As we have repeatedly mentioned, though enjoying nice theoretical advantages, in practice it is impossible to use the required full covariance matrix to compute the confidence interval in PairRank (neural), the diagonal approximation creates an unknown gap



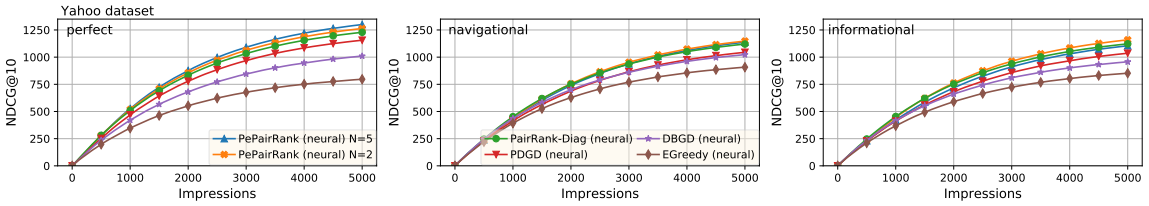
(a) Offline performance (NDCG@10) of neural rankers on the MSLR-WEB10K dataset.



(b) Offline performance (NDCG@10) of neural rankers on the Yahoo! dataset.



(c) Online performance (NDCG@10) of neural rankers on the MSLR-WEB10K dataset.

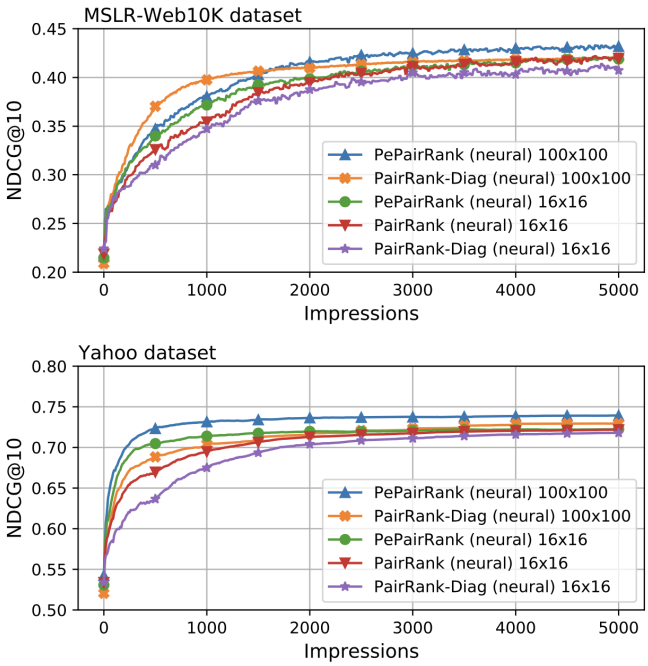


(d) Online performance (NDCG@10) of neural rankers on the Yahoo! dataset.

Figure 2.10: Offline and Online ranking performance of neural rankers on two different datasets under three different click models.

from its theoretical guarantee to practical performance. In Figure 2.11, we compare the offline performance of PePairRank (neural) with $m = 100$, PairRank (neural) with $m = 100$, and neural models with simpler neural structures under the perfect click model. The results demonstrate that for the models with 16×16 neural network, the diagonal approximation hurts the performance compared to using the full covariance matrix. As proved in our theoretical analysis, PePairRank (neural) enjoys the same theoretical regret guarantee as PairRank (neural); but because it does not need to endure the approximation, all its nice theoretical properties are preserved in its actual implementation. Compared to the results obtained in linear models, we have good reasons to believe PairRank (neural) with full covariance matrix could perform even better, but with a much larger (if not infeasible) computational overhead.

Again the impact from the number of parallel rankers one needs to maintain for perturbation-based exploration in PePairRank (neural) is still not sensitive. As shown in both Figure 2.10, $N = 2$ gave us the most promising empirical performance, with the minimum added computational overhead. This is a strongly desired property for applying PePairRank (neural) in practice.



Model	NDCG@10
P^{2}NeurRank 100x100	0.4318
olRankNet Diag 100x100	0.4201
P^{2}NeurRank 16x16	0.4189
olRankNet Full 16x16	0.4195
olRankNet Diag 16x16	0.4073

Model	NDCG@10
P^{2}NeurRank 100x100	0.7392
olRankNet Diag 100x100	0.7294
P^{2}NeurRank 16x16	0.7223
olRankNet Full 16x16	0.7217
olRankNet Diag 16x16	0.7179

Figure 2.11: Comparison among neural rankers

Zoom into PePairRank

In this experiment, we provide detailed analysis on PePairRank (neural). PePairRank (neural) has only two hyper-parameters, in addition to those inherited from PairRank (neural), i.e., the number of parallel rankers N and the scaled of pesudo noise ν^2. In Figure 2.12 and 2.13, we report the online and offline performance of PePairRank (neural) with varying value of N for a fixed variance scale ν^2 of the added noise. We can clearly observe that with a larger noise scale, e.g., $\nu = 0.1$, setting $N = 2$ gives the best performance, comparing to $N = 5$ and $N = 10$. When the added noise scale is small, e.g., $\nu = 0.01$, setting $N = 10$ demonstrates better performance than those with fewer number of models. This indicates that the variance of the added noise ν and the number of parallel rankers N together control the exploration in PePairRank (neural). A larger variance scale, e.g., $\nu = 0.1$, together with too many models, e.g., $N = 10$, lead to more aggressive exploration and less effective model training. A smaller variance, e.g., $\nu = 0.01$, together with fewer models might not lead to sufficient exploration for model update, which also leads to worse performance. Therefore, in practice, the value of ν and N should be carefully handled to perform effective exploration. And considering the added computational overhead, using fewer parallel rankers with larger scale of added noise should be a preferred solution.

Efficiency comparison.

In this section, we compare the running time of our proposed PePairRank (neural) and the PairRank (neural) with full covariance matrix and PePairRank (neural) with $N = 1$, which ind models. We performed the experiments on a NVIDIA GeForce RTX 2080Ti graphical card. As discussed before, with complex neural networks, e.g., $m = 100$, it is impossible to perform the inverse of the full covariance matrix due to both high space and time complexity. Therefore, to compare running time, we perform the experiments on the MSLR-Web10K dataset using a simpler neural network with $m = 16$, where the perfect click model is adopted to generate the clicks. The result is reported in Figure 2.14. We compare the PairRank (neural) with full covariance matrix and PePairRank (neural) with $N = 1$, which ind with a full covariance matrix, PairRank (neural) with full covariance matrix and PePairRank (neural) with $N = 1$, which ind with diagonal covariance matrix, PePairRank (neural) with N=2 and PePairRank (neural) with N = 1. For PePairRank (neural), no extra computation is required for exploration. Therefore, the running time of PePairRank (neural) with N = 1 can be viewed as the time used for the model training. We can clearly notice the big gap between the running time of PairRank (neural) with full covariance matrix and PePairRank (neural) with $N = 1$, which indicates the

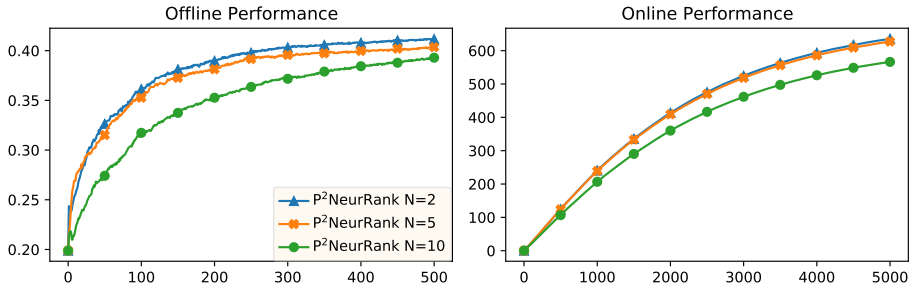


Figure 2.12: PairRank with variance $\nu^2 = 0.1$

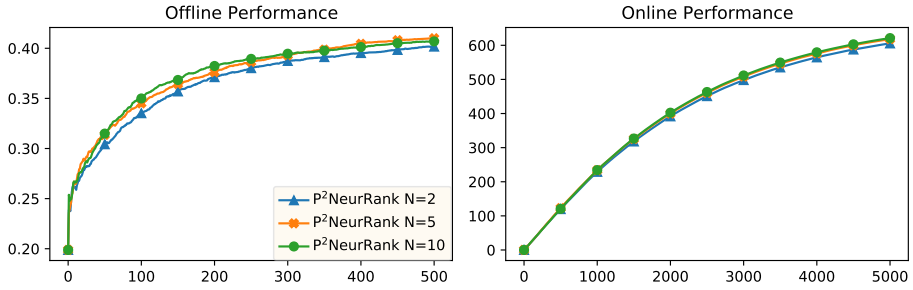


Figure 2.13: PairRank with variance $\nu^2 = 0.01$

computational overhead caused by constructing the confidence interval with the full covariance matrix. Using diagonal approximation greatly reduces the total running time. However, according to our previous discussion, there is no theoretical performance guarantee for such an approximation, and our empirical results show that the diagonal approximation often leads to decreased performance in both offline and online evaluations. On the other hand, PePairRank (neural) with N=2 takes slightly more time than PairRank (neural) with diagonal approximation, while the empirical performance is significantly better (shown in Figure 2.9 and Figure 2.11). Besides, in practice, the N models can be trained in parallel, which will further reduce the running time. This demonstrate the feasibility and advantage of our proposed OL2R model in real applications.

2.3 Conclusion

In this chapter, we focus on improving the relaibility of decision-making in OL2R with unreliable user feedback. Motivated by the success of offline models, we propose to estimate a pairwise learning to rank model on the fly, named as PairRank. Based on the model's pairwise order estimation confidence, exploration is performed only on the pairs where the ranker is still uncertain, i.e., *divide-and-conquer*. It reduces the exponentially large search space to quadratic. We prove a sub-linear upper regret bound defined on the number of mis-ordered pairs, which directly links PairRank's convergence with classical ranking evaluations. Our empirical experiments support our regret analysis and demonstrate significant improvement of PairRank over several state-of-the-art OL2R baselines. Then, we developed a provable efficient exploration strategy for neural PairRank based on bootstrapping, named as PePairRank. Confidence interval based PairRank solutions are too expensive to be exactly implemented in practice. PePairRank is proved to induce a sublinear upper regret bound counted over the number of mis-ordered pairs during online result serving, and its added computational overhead is feasible. Our extensive empirical evaluations demonstrate that our perturbation-based exploration unleashes the power of neural rankers in OL2R, with minimally added computational overhead (e.g., oftentimes only one additional ranker is needed to introduce the required exploration). And our perturbation-based exploration is general and can also be used in linear models when the input feature dimension is very large.

Our effort sheds light on deploying powerful offline learning to rank solutions online and directly optimizing rank-based metrics, e.g., RankNet and LambdaRank. Our current theoretical analysis depends on gradient descent over the entire training set for model update in each round, which is still expensive and should be further optimized. We would like to investigate the possibility of more efficient model update, e.g., online

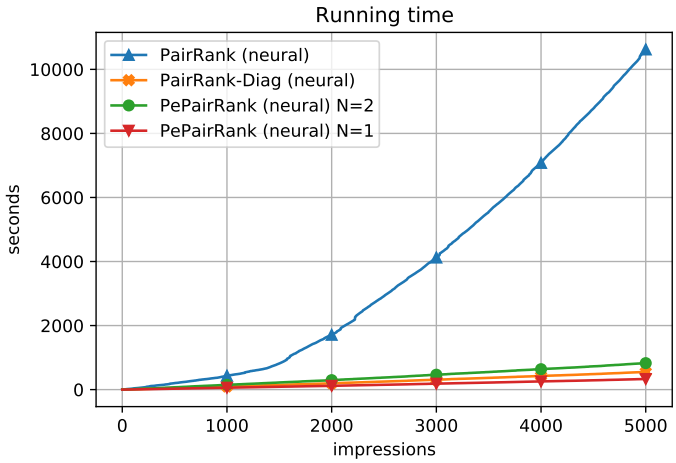


Figure 2.14: Efficiency comparison

stochastic gradient descent or continual learning, and the corresponding effect on model convergence and regret analysis. In addition, how to generalize our neural ranker architecture to more flexible choices, e.g., recurrent neural networks and transformers, is another important direction of our future work. Most OL2R solutions focus on population-level ranker estimation; thanks to the improved learning efficiency by PairRank, it is possible for us to study individual-level ranking problems, e.g., personalized OL2R.

Chapter 3

Fair Online Learning to Rank

Existing OL2R solutions target at user-focused utility optimization, which unfortunately ignores the impact of result ranking on the content providers, who might receive differential attention from the users depending on their results' ranked positions. More specifically, it is widely known that the position of an instance in the ranked list has a crucial influence on its exposure and chance to be consumed by the users, e.g., position bias [5, 6, 85]. Even a minor difference in relevance can translate into huge discrepancy in exposure across groups and thus societal or economic impact [86, 87]. For example, job postings ranked on top positions of LinkedIn's search result pages are more likely to be examined and considered by most applicants; as a result, those employers can gain an edge in the business competitions [88]. In this chapter, we introduce our research on fairness aspect of OL2R, aiming to develop OL2R solutions that are fair, and thus reliable to the content providers. We present the work focusing on calibrating the exploration and exploitation trade-off in OL2R for fairness constraints.

3.1 Fair Exploration-Exploitation for OL2R

Fairness does not only pertain to OL2R; instead, any ranking solutions (online or offline) should concern the fairness. But OL2R brings in unique new challenges for fair ranking: compared to the scenario where a pre-trained ranker is deployed in offline ranking, the differential treatment can be *accumulated and amplified* in a faster rate during the course of OL2R. This is driven by the need of *exploration* in OL2R, where the algorithm has to continuously place an intentionally selected subset of results on top. Though various exploration strategies have been proposed in OL2R literature, including those exploring in model space [10, 12, 17–19] and ranking space [20–26], none of them consider fairness when presenting results to users. When the exploration is deterministic, e.g., confidence interval based methods [22, 23, 26], the situation can become even worse, especially when the confidence estimation is correlated with protected attributes of results. This introduces a new conflict in the already complicated explore-exploit dilemma in OL2R. For instance, to ensure fairness, some results cannot be displayed, which slows down or even prevents online model learning. The slow improving relevance estimation in turn can lead to poor user experience (i.e., higher regret); in the meanwhile, if the bias from earlier OL2R update cannot be quickly eliminated by new feedback, unfair result ranking will be accumulated and amplified (e.g., one group is always mistakenly preferred). This leads to a new paradox among three elements in OL2R: *fairness*, *exploration* and *exploitation*.

In this chapter, we introduce a general new framework to achieve fairness defined on group exposure during OL2R, named as **FairExp**. The key idea is to calibrate the trade-off between exploration and exploitation under fairness constraints. In particular, when the model is exploring a set of results for relevance feedback, the exploration is confined within a subset of random permutations, where fairness across groups is maintained while the feedback is still unbiased. When the model is exploiting, fairness is directly enforced. This is achieved by constructing a set of group-level placement templates for the top-k positions. Briefly, in each round of result serving, candidate templates will first be selected based on the currently accumulated unfairness. Based on the expected exposure of each rank position under the current query, all templates need to maintain

unfairness below a threshold. As fairness on group exposure only concerns the placement of each group (i.e., their ranking order), the construction of such templates is independent from the ranker's relevance estimation. Then, the ranked list will be generated with respect to the chosen template and the required exploration by the OL2R algorithm. When multiple templates are qualified, the one with the projected minimum OL2R regret will be chosen (i.e., exploitation). In other words, fairness is treated as a hard constraint when striking the balance between exploration and exploitation.

Our proposed framework is *general* and can be applied to OL2R solutions with ranking space exploration, as long as their exploration is deterministic (i.e., where to explore can be explicitly computed). And a typical family of such solutions are the confidence interval based methods, such as TopRank [22], RecurRank [23] and PairRank [26]. Extensive empirical analysis of the resulting algorithm is performed on two public learning to rank datasets to demonstrate the effectiveness and advantages of our proposed framework comparing to existing fair OL2R solutions.

3.1.1 Related Work

In the context of decision-making, fairness is the absence of any prejudice or favoritism toward an individual or group based on their inherent or acquired characteristics [89], such as gender and race. As fairness is an elusive concept, an abundance of definitions and models of fairness have been proposed. According to the level of fairness, fairness formulations can be distinguished into [90] *individual fairness*, i.e., similar individuals should be treated similarly, and *group fairness*, i.e., individuals are partitioned into groups based on their protected attributes and different groups should be treated similarly. In this proposed work, we focus on group fairness.

Most work in fairness so far focused on classification algorithms used in decision making. The definition of fairness varies under different concerns of the protected individuals or groups so as the corresponding methods. Parity-based metrics typically consider the predicted positive rates across different groups. For example, *Demographic or statistical parity* [91–94] defines fairness as an equal probability of being classified with positive label, while *disparate impact* [93] considers the ratio between unprivileged and privileged groups. There are also some fairness definitions focusing on the confusion matrix which take additional aspects such as true positive rate, true negative rate into consideration, such as *equal opportunity* [95, 96], which consider a algorithm to be fair if it true positive rate is the same across different groups, and *equalized odds* [97], which considers both true positive rate and false positive rate to be the same across different groups. Besides, *counterfactual fairness* [98–100] was proposed to ensure that the prediction for an individual coincides with the decision if the sensitive variable would have been different. To tackle the fairness problem in classification, with the defined fairness metrics, many methods are proposed to mitigating the unfairness in the decision-making. Causal methods assumes that the training data reflect some form of underlying discrimination, and try to uncover causal relationships in the data and find dependencies between sensitive and non-sensitive variables [98, 100–105]. Sampling methods focus on identifying groups of data that are disadvantaged by the model, and creating samples for the training of robust algorithms [106–111]. There is also some work focusing on learning a fair representation of the data with a mapping or projection function in which fairness in ensured and the fidelity of the prediction is preserved [91, 93, 112–114]. Regularization and constraint optimization methods [92, 93, 115–119] are popularly used to either penalize the model for discriminatory practices, or include notions of fairness in the loss function during model training.

Unlike the classification problem, ranking problem is a structured output prediction problem with an exponentially large output space. Due to the special property of ranking, the fairness definition from classification can not directly be translated to ranking problems. Unfairness in ranking typically either comes from the external factors such as the bias in the data or during the relevance estimation [120–122], or the internal design of the ranking system [123–127], e.g., the ranking generated by the system may be unfair to different groups.

Fairness in ranking is concerned with an insufficient presence or a consistently differential treatment over different groups in the ranked lists [128]. Various definitions of fairness in ranking have been proposed, together with their fair ranking solutions [129]. Most metrics are motivated by those defined in fair classification problems [89]. Specifically, solutions focusing on group parity enforces a proportional allocation of exposure between groups. For example, [123] proposed to reduce the difference in occurrences of different groups on a subset of ranking positions. And a series of work proposed to set a limit on the number of items

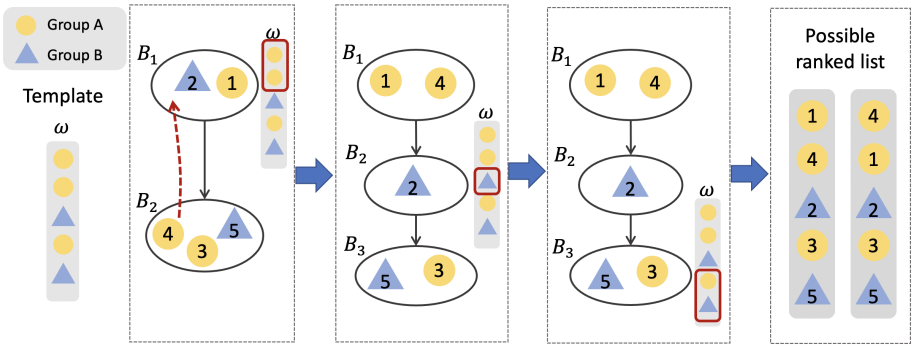


Figure 3.1: Illustration of FairSwap: calibrate pairwise exploration in PairRank under fairness constraint. It satisfies the required fair placement of groups by introducing minimum number of swaps between blocks generated in PairRank.

from each group in the top-k positions [88, 124, 127]. On the other hand, merit-based fairness of exposure allocates exposure to groups based on their merit instead of the group size. Both in-processing [130–132] and post-processing approaches [86, 87] are proposed to achieve this type of fairness in ranking. However, all the methods assume either the expert-labeled relevance or logged implicit feedback are available for model training or result ranking beforehand. This unfortunately prevents the application of such fair ranking solutions in the online setting.

The most relevant work to ours is the FairCo algorithm proposed in [133], which applies proportional control to mitigate unfairness in dynamic learning to rank. Its online ranker update is achieved by the inverse propensity scoring method on user clicks [134]. This however imposes strong assumption on the user examination behavior (i.e., position-based examination). And when enforcing fairness, uncertainty in the ranker's relevance estimation is not considered. As a result, inaccurate fairness measure leads to unfair control in this solution. Additionally, the provided theoretical analysis is limited to a fixed set of items, which cannot be applied to the unseen queries and items. Some recent work studied fairness under inference uncertainty but in somehow different settings. [135] studied the impact from the uncertainty in sensitive attributes on fair ranking. And [136] focused on the uncertainty in the estimated merit and individual fairness in the ranking problem. In this work, we study group fairness defined by the exposure on the item side in OL2R, where we assume the grouping of items is given and new queries and items can emerge at any time.

3.1.2 Calibrating Explore-Exploit Trade-off Trade-off for Fair OL2R

Fairness in OL2R

It is asserted by the probability ranking principle that user-side utility reaches its optimal when the documents are ranked by the expected values of their estimated relevance (merit) to the user [137]. However, it unfortunately ignores the differential treatment that content providers would receive from the user side in such rankings. The key resource that a ranking system allocates among the documents is exposure [87], which influences the probability that the documents to be examined by the users and consequentially the social and/or economic benefit the content provider will receive. Without loss of generality, in this work, we assume all ranking candidates among all queries over time belong to two groups (G_A and G_B). We define the instantaneous unfairness resulted in a particular ranked list as the difference between the exposure received by group G_A and group G_B from the presented ranking,

$$UF(G_A, G_B, \beta) = Exposure(G_A) - \beta Exposure(G_B),$$

where β is an unfairness coefficient that controls the relative exposure that group G_A and G_B should receive. β is an hyper-parameter chosen by the system designer to weigh the discrepancy in exposure between the two groups based on the need of specific applications. The definition of exposure is also application-specific: it can be quantified as the examination probability [133] or the clicks/dwelling time received by the documents. In this work, we adopt the classical choice of examination probability to simplify our discussion, but our solution can be generalized to other definitions with specific instrument to measure/infer the corresponding exposure.

Algorithm 3 FairExp-PairRank

Input: unfairness coefficient β, unfairness threshold ϵ, L2 coefficient λ, uncertainty coefficient δ_1, length of the returned ranked list k.
Initialize $\mathbf{M}_0 = \lambda\mathbf{I}, \boldsymbol{\theta}_1 = 0, UF_0 = 0$.
for $t = 1, \ldots, T$ **do**
  $q_t \leftarrow$receive_query(t)
  $\mathcal{X}_t = \{\mathbf{x}_1^t, \cdots, \mathbf{x}_{L_t}^t\} \leftarrow$ retrieve_documents(q_t)
  $\bar{\Omega}_t^k \leftarrow$ retrieve_group_placement(k, ϵ, β, $UF_{t-1}^{realized}$)
  $S_t^c, S_t^u \leftarrow$ construct_order_sets($\mathcal{X}_t, \boldsymbol{\theta}_{t-1}, \mathbf{M}_{t-1}, \delta_1$)
  for $\omega \in \bar{\Omega}_t^k$ **do**
    $\pi^\omega, a^\omega \leftarrow$ FairSwap(S_t^c, S_t^u, ω)
  $\pi_t \leftarrow \pi^\omega = \text{argmin}_{\omega \in \bar{\Omega}_t^k} n^\omega$
  $C_t \leftarrow$ collect_click_feedback(π_t)
  $UF_t \leftarrow UF_{t-1}$ + realized_unfairness(π_t, C_t)
  $D_t \leftarrow$ construct_training_set(π_t, C_t)
  Obtain $\boldsymbol{\theta}_t$ by minimizing Eq (2.1.1)
  $\mathbf{A}_t = \mathbf{A}_{t-1} + \sum_{(i,j) \in D_t} \mathbf{x}_{ij}\mathbf{x}_{ij}^\top$

To capture the difference of the cumulative treatment received during the course of OL2R up to time t, we consider a ranking system fair, if for any $t \in [T]$, we have

$$|UF_t| = \left| \sum\nolimits_{s=1}^{t} \Big(Exposure_s(G_A) - \beta Exposure_s(G_B) \Big) \right| \leq \epsilon, \tag{3.1.1}$$

where $\epsilon > 0$ is a threshold chosen by the system designer to satisfy task-specific requirement of fairness.

Our definition of unfairness is general to cover most of the existing exposure-based ranking fairness metrics. First, for demographic parity constraint proposed in [87] which enforces that the average exposure of groups to be equal, β can be set as the expected ratio of the group sizes between G_A and G_B; while for fairness concerning the disparate treatment [87, 133], of which the exposure received by each group should be proportional to their relevance or merit, β can be set as the ratio of expected utility between groups. Besides, our unfairness definition is not limited to only two groups. Different pairs of groups can have different β to control the relative exposure as long as the βs satisfy the transitivity constraint.

Fair Placement of Groups

Group fairness concerns whether each group of documents receives fair treatment, e.g., the exposure received by documents belonging to group G_A should be comparable to that received by documents belonging to group G_B. Therefore, when considering the top-k ranking positions, the placement of groups, instead of the detailed placement of documents, affects the unfairness across groups. Such a group-level ranking space for fairness control is much smaller than the exponentially sized ranking space for individual documents. Specifically, with a fixed value k, which is chosen by the system designer for each specific ranking task, the set of all possible combinations of group rankings Ω^k is fixed and can be constructed beforehand. For example, with two groups at top-k positions, there are in total 2^k different group-level placements, assuming each group has at least k candidate documents under each query.

For a particular group-level placement template for the top-k positions, its resulting unfairness can be computed based on the expected exposure at each position, before the user examines the realized ranking. For example, under the position-based examination model [138], the examination probability of each document i only depends on its position in the ranked list, e.g., $P(e(i) = 1|i, \pi) = P(\pi(i))$. Several techniques have been proposed to estimate such position-based examination probabilities in practice [134, 139, 140]. Other more sophisticated examination models can also be employed for the purpose to account for more factors than just the rank positions, such as document content [141] and query intent [142]. Then, given the group-level placement, the expected exposure each group will receive can be projected accordingly beforehand; and such values can be indexed for efficient online access later. For example, at round t, for the group placement

template $\omega = \{A, A, B, A, B\} \in \Omega^5$, we can compute the expected exposure for group G_A and G_B as,

$$\widetilde{Exposure}_t^{\omega}(G_A) = P(1) + P(2) + P(4),$$
$$\widetilde{Exposure}_t^{\omega}(G_B) = P(3) + P(5).$$

where $\widetilde{Exposure}_t^{\omega}(G)$ denotes the expected exposure that group G will receive under the group placement template ω at round t.

According to the definition in Eq (3.1.1), with the unfairness coefficient β, and currently realized unfairness UF_{t-1}, if ω is chosen, the expected (projected) cumulative unfairness at round t, i.e., $\widetilde{UF}_t$, can be estimated as,

$$\widetilde{UF}_t^{\omega}(G_A, G_B, \beta) = UF_{t-1}(G_A, G_B, \beta) + \widetilde{Exposure}_t^{\omega}(G_A) - \beta \times \widetilde{Exposure}_t^{\omega}(G_B). \tag{3.1.2}$$

$|\widetilde{UF}_t^{\omega}(G_A, G_B, \beta)|$ can then be used to guide the selection of final group placement template for round t. But we should note that $\widetilde{UF}_t^{\omega}(G_A, G_B, \beta)$ indicates the estimated cumulative unfairness for template ω. After user examines the results, the realized (true) unfairness $UF_t(G_A, G_B, \beta)$ is computed based on the actual exposure in the displayed ranking π_t, which will then be used to guide fairness control in the next round. Under specific types of exposure, e.g., position-based examination, the expected exposure can be directly used to update $UF_t(G_A, G_B, \beta)$.

Calibrating Explore-Exploit Trade-off

Based on the notion of projected unfairness of a particular group placement template, $UF_t^{\omega}(G_A, G_B, \beta)$, we design our FairExp framework to ensure fairness during OL2R.

To make our discussion more concrete, we demonstrate how to cast PairRank into a fair OL2R solution with our proposed FairExp framework in Algorithm 3, named as FairExp-PairRank. Specifically, at round t, we first select all templates from Ω^k that satisfy Eq (3.1.1) under the current cumulative unfairness UF_{t-1} and the projected instantaneous unfairness induced by the template (line 6 in Algorithm 3). More formally, denote this set of templates as $\bar{\Omega}_t^k$, such that $\bar{\Omega}_t^k = \{\omega \in \Omega^k : |UF_{t-1} + \widetilde{Exposure}_t^{\omega}(G_A) - \beta \times \widetilde{Exposure}_t^{\omega}(G_B)| \leq \epsilon, \}$. For each group-level placement template ω in $\bar{\Omega}_t^k$, we use it to calibrate the exploration and exploitation at round t: the exploration will be confined within a subset of random permutations satisfying ω; while for the exploitation, the fairness will be directly enforced: when the algorithm's chosen ranking is against the template's requirement, follow the template's.

To find the satisfying fair ranking at each round t in PairRank, the most straightforward way is to first generate all possible ranked lists that satisfy the templates in $\bar{\Omega}_t^k$, and then return the one with minimum expected regret, which can be readily computed based on the violation of certain rank order set S_t^c in PairRank [26]. However, the search space becomes exponentially large. Hence though valid, such an exhaustive search is prohibitively expensive for online result serving.

In FairExp-PairRank, we propose to modify the block structure in PairRank to efficiently find the ranking with minimum expected regret under each qualifying template $\omega \in \bar{\Omega}_t^k$, and then find the best ranking across all templates (line 8 - line 11 in Algorithm 3). We name this modification method as FairSwap and describe its procedure in Algorithm 4. More specifically, with the certain and uncertain rank orders (S_t^c and S_t^u), PairRank divides the documents into a sorted list of n blocks $B = \{b_1, ..., b_n\}$ (line 2 in Algorithm 4). Then, for a chosen template, we can segment it according to the size of each block in the sorted list, and then calibrate the blocks segment by segment. For example, in line 5 of Algorithm 4, a segment ω' of the template ω is obtained based on current block b_i. Then FairSwap will check whether the group combination of the block is satisfied with the template segment ω'. If b_i is satisfied ($cnt = 0$ in Algorithm 4), no modification is needed for the current block. The ranking of documents can be generated with respect to the group template. As any permutation of documents within a block is allowed in PairRank, no regret is added. But for the segment where we do not have enough documents for group g in the corresponding block ($cnt \neq 0$ in Algorithm 4), we have to fetch satisfying documents from the lower ranked block(s) (line 9 to line 13 in Algorithm 4). FairSwap

Algorithm 4 FairSwap

1: **Input:** certain order set S_t^c, uncertain order set S_t^u, group placement template ω.
2: $B = \{b_1, b_2, ...b_n\} \leftarrow$ rank_blocks(S_t^c, S_t^u)
3: $s = 0$
4: **for** b_i in B **do**
5: $\quad e \leftarrow s + size(b_i), \omega' \leftarrow \omega[s:e], s \leftarrow e$
6: $\quad g, cnt \leftarrow$ get_missing_group(ω', b_i)
7: $\quad$ **if** $cnt \neq 0$ **then**
8: $\quad\quad \Theta = \emptyset, j \leftarrow i + 1$
9: $\quad\quad$ **while** $|\Theta| < cnt$ and $j \leq n$ **do**
10: $\quad\quad\quad b_j^g \leftarrow$ fetch_documents($b_j, g, cnt - |\Theta|$)
11: $\quad\quad\quad \Theta \leftarrow \Theta \cup b_j^g$
12: $\quad\quad\quad j \leftarrow j + 1$
13: $\quad\quad b_i^{new} \leftarrow$ fetch_documents($b_i, \bar{g}, cnt$)
14: $\quad\quad b_i \leftarrow b_i - b^{new}, b_i \leftarrow b_i \cup \Theta$
15: $\quad\quad$ Insert b^{new} after b_i and update block list B
16: $\pi^\omega \leftarrow$ generate_ranking(B, ω)
17: $a^\omega \leftarrow$ count_violations(π^ω, S_t^c)
18: return π^ω, a^ω

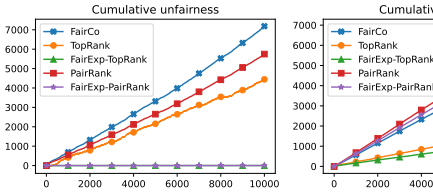


Figure 3.2: MSLR-Web10K with group attribute=#inlinks, $\beta = 1.0$

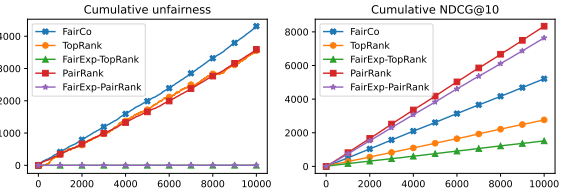


Figure 3.3: Yahoo with group attribute=Feature 9, $\beta = 1.0$

sequentially scans the following blocks after b_i, and fetches documents belonging to group g in the block (line 10 - line 11 in Algorithm 4). Once the sufficient number of documents belonging to group g are obtained, FairSwap will add them to the current block b_i, and remove cnt documents belonging to $\bar{g}$ to a new block b^{new}, which will be inserted after b_i to guarantee that the added regret is minimized. After the structure of blocks is calibrated, the ranked list π^ω and the corresponding number of violations a^ω will be generated.

We use an intuitive example in Figure 3.1 to further illustrate the procedure of FairSwap. In this example, PairRank already partitions the five documents into two blocks: $B_1 = \{\mathbf{x}_1, \mathbf{x}_2\}$ and $B_2 = \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}$ according to PairRank, and requires the first block to be ranked above the second. The chosen template ω requires the ranking to follow $\{A, A, B, A, B\}$. Then, FairSwap gets the first segment $\omega' = \{A, A\}$ based on the size of the first block. ω' requires two documents from G_A, but there is only one such document in the first block. Hence FairSwap needs to fetch one from the next block. FairSwap will arbitrarily move one of them into the first block, e.g., $\mathbf{x}_4$ in the example, to avoid any presentation bias. But because the first segment only requires two documents and they must belong to G_A, $\mathbf{x}_2$ has to be moved downwards. Therefore, FairSwap creates a new block only containing $\mathbf{x}_2$ and inserts it before the next block. Then FairSwap moves to the next segments, until new violations emerge between a segment and its corresponding block. The same procedure will be applied to calibrate the block structure. In this result, because $\mathbf{x}_4$ is moved up ahead of $\mathbf{x}_2$ to satisfy ω, added regret is introduced. And this added regret can be precisely computed using the certain rank order set S_t^c in PairRank. For instance, in this example, the added regret will be 2, if the finally output ranking is $\{\mathbf{x}_4 \succ \mathbf{x}_1 \succ \mathbf{x}_2 \succ \mathbf{x}_3 \succ \mathbf{x}_5\}$.

3.1.3 Experiment

In this section, we empirically compared our proposed fair OL2R solution with existing state-of-the-art OL2R and fair OL2R algorithms on two public learning to rank benchmark datasets.

Table 3.1: Group information for two datasets.

Dataset	Group attribute	Group A		Group B	
		Size	Utility	Size	Utility
MSLR WEB10K	Inlink number	64.85%	0.6665	35.15%	0.6776
	PageRank	50.57%	0.7302	49.43%	0.6093
Yahoo!	Feature 9	49.22%	1.2371	50.78%	0.5893
	Feature 471	85.75%	1.2384	14.25%	1.2239

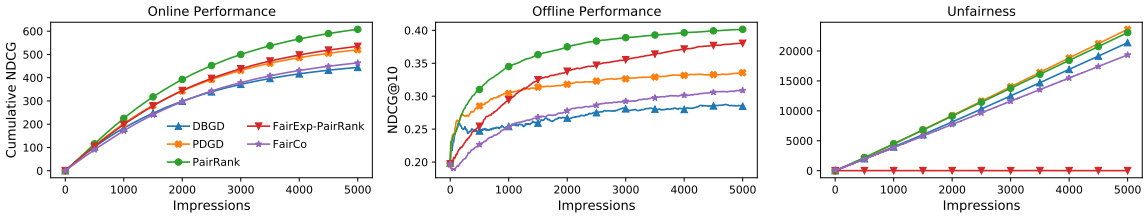


Figure 3.4: MSLR-WEB10K dataset with group attribute=number of inlinks, $\beta = 1.0$

Experiment Setup

Datasets. We experimented on two publicly available learning to rank benchmarks, Yahoo! Learning to Rank Challenge dataset [11], which consists of 292,921 queries and 709,877 documents represented by 700 ranking features, and MSLR-WEB10K [74], which contains 10,000 queries, each having 125 documents on average represented by 136 ranking features. Both datasets are labeled on a five-grade relevance scale: from not relevant (0) to perfectly relevant (4). These two datasets are the most popularly used in literature for evaluating OL2R algorithms. We followed the train/test/validation split provided in the datasets to perform the cross-validation to make our results comparable to previously reported results. These two are the most frequently used datasets for OL2R evaluations.

Since no group information about the documents is provided in these two datasets, we manually divided the datasets into groups. For MSLR-WEB10K dataset, where the detailed information about each ranking feature is provided by the data publisher, we chose the document-specific feature inlink number (feature id 128) and PageRank (feature id 130) as the group attribute, as they measure the popularity of documents. Inlink number roughly separates the dataset into two groups with similar average utility, while PageRank separates the datatset into two groups with similar size. But the Yahoo dataset did not provide any specific feature information. To follow a similar split as obtained in the MSLR-WEB10K dataset (i.e., average utility vs., size across document groups), we checked all the features and chose feature 9 and feature 471 as the group attribute. The detailed group constructions in these two datasets are provided in Table 3.1. In addition to studying the trade-off among exploration, exploitation and fairness in OL2R, our choice of group attributes will also demonstrate how the intrinsic property of the problem/dataset affects these three elements in fair OL2R.

User interaction simulation. We follow standard settings in OL2R literature to simulate user clicks [25, 26], which is the most popularly used procedure for OL2R evaluation. At each round, a query is uniformly sampled from the training set for result serving. A ranked list will be determined by the model and returned to the user. Use clicks are simulated with a dependent click model (DCM) [68], which assumes that the user will sequentially scan the list and make click decisions on the examined documents. In DCM, the probabilities of clicking on a given document and stopping the subsequent examination are both conditioned on the document's true relevance label. As both datasets are labeled on a five-grade relevance scale, we set the click probability for relevance 0-4 as: $\{0.4, 0.6, 0.7, 0.8, 0.9\}$, and the stop probability as: $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. Such a click model simulates the users who tend to examine more documents, but sometimes click on irrelevant documents. Hence, there exists significant amount of noise in the click feedback. To reflect presentation bias, only top $k = 10$ ranked results are returned to the users.

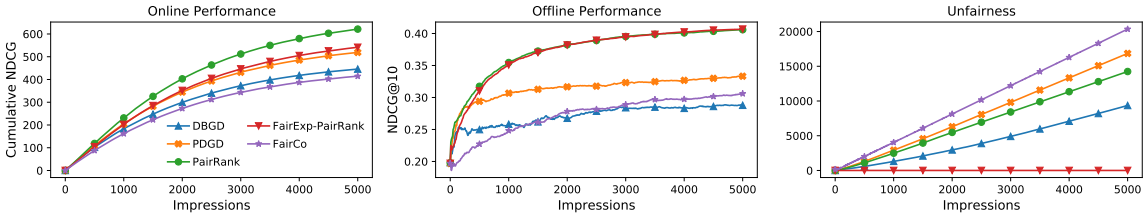


Figure 3.5: MSLR-WEB10K dataset with group attribute=PageRank score, $\beta = 1.22$

Baselines and settings. We compare the proposed fair framework FairExp on top of TopRank and PairRank, with several state-of-the-art OL2R solutions, and the fair OL2R model, FairCo.

- **DBGD** [12]: DBGD uniformly samples a direction from the entire model space for exploration and model update.
- **PDGD** [25]: PDGD samples the next ranked document from a Plackett-Luce model and estimates gradients from the inferred pairwise preferences.
- **PairRank** [26]: PairRank directly learns a pairwise model with logistic regression, and explores based on the model's uncertainty with divide-and-conquer.
- **TopRank** [22]: TopRank maintains a partial order over the ranking candidates and randomly permutes the pairs where the model's estimation is still uncertain. In essence, TopRank is very similar to PairRank, but it performs the rank estimation under each query independently.
- **FairCo** [133]: FairCo achieves the group fairness with a proportional control based on the exposure of each group. This is the only fair OL2R solution known in literature so far.

For the comparison between FairExp-PairRank and the baselines, we perform a cross-validation on both datasets. For each fold in cross validation, the models are trained with the simulated clicks on the training dataset, and the hyper-parameters are selected based on their offline performance (relevance learning) on the validation set. We used grid search for the best set of hyper-parameters. For PDGD and DBGD, learning rate is selected among $\{10^{-i}\}_{i=1}^{3}$. For PairRank, we did a grid search for its regularization parameter λ over $\{10^{-i}\}_{i=1}^{3}$ and exploration parameter α over $\{10^{-i}\}_{i=1}^{3}$. FairCo depends on the pre-defined examination probability of each position to perform inverse propensity scoring. We followed [8] to use randomization to estimate the position-based examination probabilities. The fairness coefficient in FairCo is searched over $\{10^{-i}\}_{i=1}^{3}$. For FairExp-PairRank, we did the same grid search for the hyper-parameters associated with PairRank. To make the fairness control comparable between FairExp-PairRank and FairCo, we set β to the ratio of the true utility of each group. For example, for MSLR-WEB10K dataset with inlink as the group attribute, we set β to 1 according to the statistics reported in Table 3.1. For FairCo, we set the merit of each group based on the ground-truth relevance labels. To calculate exposure, FairExp-PairRank, FairExp-TopRank and FairCo use the independently estimated examination probabilities as the exposure received by each document. With such a setting, all these three fair OL2R solutions focus on the merit-based exposure for group fairness.

Evaluation. We evaluate the fair OL2R models under three aspects, user satisfaction (i.e., exploitation), relevance learning (i.e., exploration), and unfairness treatments across groups over time. For user satisfaction and relevance learning, we adopt the same metric and setting, e.g., NDCG@10 and Cumulative NDCG@10, as we did in OL2R. Besides, we calculate unfairness according to Eq (3.1.1) with a chosen β value in each dataset.

Experiment Results

FairExp-TopRank. TopRank learns the ranking of documents for each query independently, such that its sample complexity increases linearly with the number of unique queries and documents in a corpus. This puts it in a disadvantageous position when compared with the parametric ranking models, such as PairRank, because TopRank cannot generalize its observations across queries. Hence, in practice when the ranking

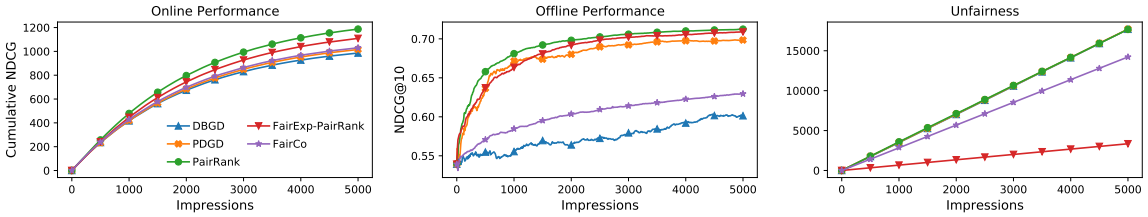


Figure 3.6: Yahoo dataset with group attribute=feature 9, $\beta = 1.0$

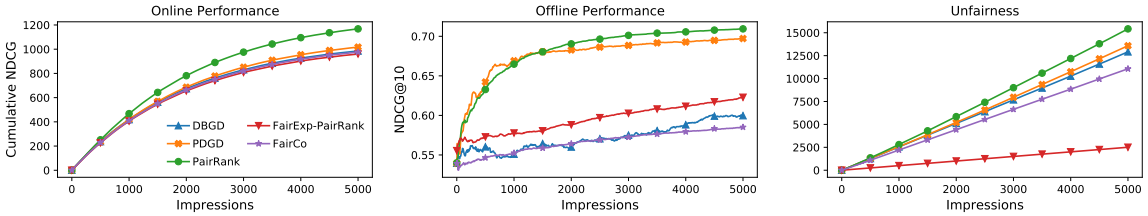


Figure 3.7: Yahoo dataset with group attribute=feature 471, $\beta = 0.477$

features are available, TopRank is not a good choice for OL2R. But in this experiment, we created a dedicated dataset only to evaluate the effectiveness of our proposed FairExp framework for turning TopRank into a fair OL2R solution. In particular, we randomly select 10 queries from both MSLR-Web10K and Yahoo datasets, and performed evaluation only for TopRank, PairRank and their fair versions, and FairCo for 10,000 rounds, so that each query will be selected and modeled for sufficient times in TopRank and FairExp-TopRank.

We reported the cumulative unfairness and cumulative NDCG@10 for the five OL2R solutions in Figure 3.2 and 3.3. We can clearly observe that FairExp is able to maintain almost perfect fairness for TopRank and PairRank, compared to FairCo. To our surprise, FairCo even accumulated more unfairness than TopRank and PairRank, which are not constrained by fairness controls at all. This is caused by the "overreaction" in FairCo for fairness control. FairCo considers the currently accumulated disparity between the exposure received in each group and directly promotes all documents in the disadvantaged group by the disparity. Such a promotion fails to consider what's actually needed in the current round, and is very likely to over compensate. As a result, FairCo has to compensate different groups back and forth. But in our FairExp framework, the projected unfairness after the current round is considered to minimize the impact on subsequent rounds. On the other hand, in terms of ranking performance, TopRank performed the worst, comparing to FairCo and PairRank. And FairExp further downgrades its ranking performance due to the fairness control. Similar effect is observed in PairRank vs., FairExp-PairRank, but it is expected. In our following experiments, we will exclude TopRank and its fair version in our discussion, as their ranking performance is too much worse than other parametric OL2R solutions.

Exploration, exploitation, and fairness in FairExp-PairRank. We report the online performance, offline performance and unfairness on the two datasets with four different group attributes in Figure 3.4 to 3.7. First, we can clearly observe that PairRank consistently showed the best online performance across all the settings. And due to the fairness control, FairExp-PairRank encounters reduced online performance. This is the expected trade-off between utility and fairness: to allocate fair exposure for different groups, documents with low-quality have to be swapped to the top positions. Fortunately, due to our FairSwap strategy that guarantees minimum added regret and the already good performance of PairRank, the cost of such trade-off is under control, such that FairExp-PairRank even outperformed most of other OL2R solutions that are not subject to fairness constraints. On the other hand, FairCo showed worse online performance than FairExp-PairRank and other OL2R solutions. We attribute this to FairCo's "overreacting" fairness control. In FairCo, to reduce the current unfairness, the ranking scores of documents belonging to the underrepresented group will all be promoted by the same value; but the differences of their relevance quality are ignored. As a result, documents with low-quality will be equally promoted for fairness, which directly results in its bad online performance.

Offline performance indicates the models' convergence for relevance learning. We can observe that the

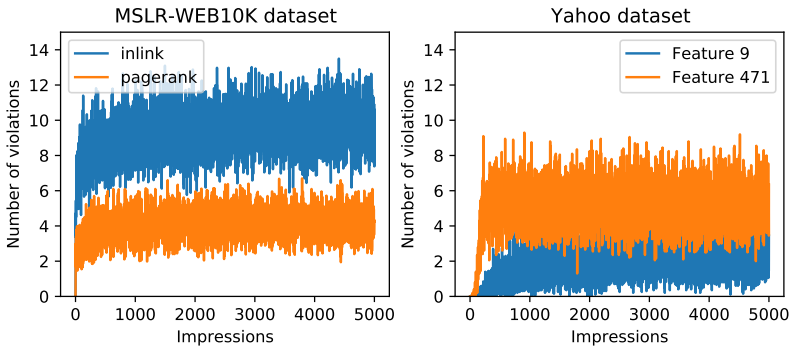


Figure 3.8: Number of violations on certain rank orders.

convergence rate of FairExp-PairRank is slower than that in PairRank. This is consistent with our previous discussion that due to the fairness control some documents originally selected for exploration cannot be displayed, which directly slows down the improvement of relevance estimation. Besides, we can observe that such control shows different impact across different group settings. For example, there is almost no impact on the offline performance of FairExp-PairRank on MSLR-WEB10K dataset when choosing the PageRank feature to define groups, while on Yahoo dataset with feature 471 as the group attribute, the drop is significant. We checked the detailed output in FairExp-PairRank and found that in the former case, the fairness constraint can be largely satisfied with the original rankings in PairRank. Therefore, little calibration is added. However, in the latter case the underexplored documents always got demoted by fairness control (because the group size differs significantly), it seriously slowed down relevance learning.

On the other hand, thanks to our template-guided fairness control, FairExp-PairRank showed significant advantages in its cumulative unfairness compared to all baselines across all settings, as fairness is handled as a hard constraint in it. It is worth noting that in the Yahoo dataset, even with FairExp-PairRank, the cumulative unfairness is increasing. This is because there are many queries with highly imbalanced candidate documents from the two groups such that no ranking can be generated with respect to the fair templates in $\bar{\Omega}$. In FairExp, when no qualified template can be satisfied, we will first choose a ranking with the minimum unfairness, and then compare their added regret. And this observation also reflects the intrinsic incompatibility of an environment in fair ranking problems: the distribution of queries and/or their associated documents is out of the control of algorithms. Compared to FairExp-PairRank that treats unfairness as a hard constraint, FairCo shows much higher unfairness during the course of OL2R. And this is still resulted from its "overreacting" behavior in the proportional control: it aims to fix all the previous unfair treatments in the current query, which might over compensate the current underrepresented group and lead to oscillation in the subsequent controls.

Zoom into FairExp-PairRank To further verify the effect of fairness control on the trade-off between exploration and exploitation in PairRank, we zoom into the trace of the number of violations on the certain rank orders during online interactions in Figure 3.8. We can observe that the fairness control inevitably leads to constant violations on the identified certain rank orders (i.e., added regret) and the impact varies across different group configurations. This also shows another intrinsic incompatibility in fair result ranking: when the optimal utility-driven ranking is not fair, the system has to trade utility for fairness. And this trade-off is beyond the system's control, as it is affected by the distribution of queries that are issued by the users and the associated documents that are produced by the content providers.

3.2 Conclusion

In this chapter, we focus on improving the reliability of decision-making in OL2R by ensuring fair treatment across content providers. Existing OL2R solutions focus on optimizing user-oriented utilities of the ranked list, but ignore the potential unfair treatment that result ranking can introduce on the content providers. We present a general framework FairExp for fair OL2R. During the interaction with the users, the trade-off between exploration and exploitation in OL2R is calibrated with respect to the fairness constraint. By taking advantage

of the structure of deterministic exploration in PairRank, fairness control, relevance learning and online quality can be best optimized simultaneously. We proved that our strategy introduces the minimum distortion in OL2R's regret to obtain fairness. And our empirical evaluations demonstrate the strong advantage of FairExp in balancing exploitation, exploration and fairness in OL2R.

Chapter 4

Multi-task Learning for Explainable Recommendation

Modern personalized information systems are black boxes to their users: computerized oracles give advice, but cannot be questioned. The lack of transparency in personalization [4] leaves users in a dilemma: a user can only assess the quality of personalized results by taking the suggested instances, e.g., read the recommended articles; however, in order for him/her to adopt the system's customized results, he/she needs to first build trust over the system. If users are persuaded to accept recommended results that are subsequently found to be inferior, their confidence and trust in the system will rapidly deteriorate [3, 28]. Hence, the fidelity of explanations becomes a prerequisite for explainable recommendations, and also, reliable information system. In this chapter, we introduce two solutions for explainable recommendation. Given the rich information in the users' review text, we adopt feature-level sentiment analysis [34, 143] on reviews to construct a domain-specific sentiment lexicon. Based on the lexicon, we develop two mutli-task learning models [31–33], where two companion learning tasks are formed: one focuses on *user preference modeling for personalization*, and another focuses on *content modeling for explanation*. The two tasks operate in the same latent space by sharing the representation of users, instances and aspects. Therefore, the learning of latent factors for users and instances are thus shaped by both the loss from personalization and explanation tasks.

4.1 Explainable Recommendation with Joint Tensor Factorization

We believe the tension between recommendation quality and explanation fidelity is not necessarily inevitable; and a good recommendation algorithm should consist of companion learning tasks focusing on different aspects of users' decisions over the recommended instances, such that the observed final decisions (e.g., clicks or ratings) can be mutually explained by the associated observations. Due to the effectiveness in numerous practical deployments [27, 144], we focus on latent factor models to develop explainable personalization solutions. The essence of a latent factor model based personalization solution is to represent users and instances in a latent space such that the distance between them captures the users' preference over the instances [27, 144]. To introduce explainability, when constructing the latent space, we also need to consider the generation of explanations. For example, the identified decision aspects should also be embedded in this space, such that their distances to a particular user-instance pair reflect their importance for the user to evaluate the customized instance. Intuitively, in this latent space, users should be surrounded by their preferred instances; and each user-instance pair should be surrounded by the aspects that best characterize the matching between them (shown in Figure 4.1). Both personalization and explanation can thus be effectively addressed by searching in this latent space.

In this work, we focus on explaining factorization-based recommendation algorithms [145, 146] by taking a holistic view of item recommendation and sentiment analysis. We present a joint tensor factorization solution to integrate two complementary tasks of *user preference modeling for recommendation* and *opinionated content modeling for explanation*, i.e., a multi-task learning approach [147–149]. The task of item recommendation is

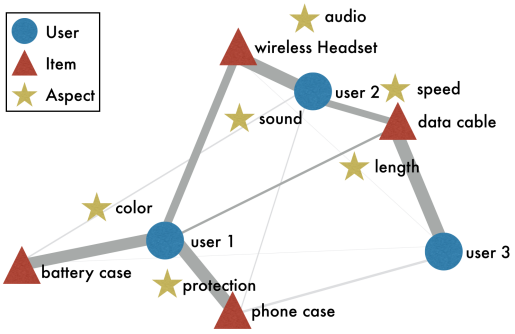


Figure 4.1: Multi-task personalization and explanation learning scheme.

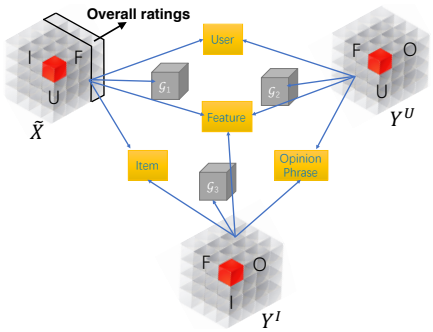


Figure 4.2: Joint tensor factorization.

modeled by a three-way tensor over user, item and feature, to describe users' preferences on individual items' features, constructed by feature-level sentiment analysis in opinionated review content. The companion task of opinionated content analysis is modeled by another two three-way tensors, one over user, feature, opinionated phrase, and another over item, feature, opinionated phrase, both of which are constructed from user-generated review content. Via a joint factorization over these three tensors, we map users, items, features and opinionated phrases to a shared latent representation space. Extensive experimental comparisons between our proposed solution and existing explainable recommendation algorithms demonstrate the effectiveness of our solution in both item recommendation and explanation generation tasks in two different application scenarios, i.e., product recommendation based on Amazon reviews and restaurant recommendation based on Yelp reviews. In particular, we perform serious user studies to investigate the utility of our explainable recommendation in practice. Positive user feedback further validates the value of our proposed solution.

4.1.1 Related Work

Explainable personalization is still in its nascent stage, although from a broader perspective it is closely related to general explainable machine learning [150–154]. Studies related to explanations can also be found in cognitive science [155], psychology [156] and philosophy [157]. Past work has considered explaining latent factor models [143], explainable deep models [158], social explainable recommendations [159], visual explanations [160], sequential explanations [161] and dynamic explanations [162]. Broadly speaking, we categorize the existing explanation methods into neighbor-based and feature-based categories. Both of them however suffer from the trade-off between recommendation quality and explanation fidelity to different extents.

The neighbor-based explanation methods root in content-based collaborative filtering [163, 164]. As the recommendations are made directly by measuring similarities between users and/or items, explaining the recommended results becomes straightforward. For example, Herlocker et al. proposed 21 types of explanation interfaces for a collaborative filtering system [165] and found a histogram showing the ratings from similar users was the most persuasive. Sharma and Cosley [28] conducted user studies to investigate the effect of social explanations, e.g., "X, Y and 2 other friends like this." But the unsatisfactory recommendation quality limits the utility of provided explanations at the first place. This type of explanation has also been used in latent factor based collaborative filtering algorithms, where the similarity is measured in the learnt latent space [29]. Zhang et al. [143, 166] combined techniques for phrase-level sentiment analysis with matrix factorization. Abdollahi and Nasraoui [29] introduced explainability as a constraint in factorization: the learned latent factors for a user should be close to those learned for the items positively rated by him/her. There are also solutions considering the latent factor models from a probabilistic view, which provides the flexibility of modeling associated opinionated text data for explanation. Wang and Blei [167] combine probabilistic matrix factorization with topic modeling for article recommendation. Explanations are provided by matching topics in items against the target user. A follow-up work [168] introduces aspect-level topic modeling to capture users' finer-grained sentiment on different aspects of an item, so that aspect-level explanations become possible. Ren et al. [169] introduce social relations into topic modeling based recommendation via a concept named viewpoint, which enables social explanation. However, as the latent space is not constructed for explanation, there is no guarantee that such type of explanations will comply with the recommendations.

Feature-based explanation methods introduce information beyond classical dyadic interaction between users and items, such as user ratings and clicks. Earlier work in this category uses meta-data of items for explanation. For instance, Tintarev et al. [170] use genre, director and cast to explain movie recommendations. Bilgic and Mooney [2] extracted keywords from recommended books as explanations. But such explanations are heavily item-focused, and therefore independent of the recommendation algorithms. Their fidelity is often questionable. Later works in this category integrate feature representation learning with recommendation model learning, with the hope that the provided explanations can best correlate with the recommendations. For example, in [143], phrase-level sentiment analysis is first used to extract users' feature-level descriptions of the items, and joint matrix or tensor factorization is then performed to map users, items and features onto the same latent space. The explanations are created by looking for the most related features to the user and recommended items in the learnt latent space, which is essentially neighbor-based explanation. But as the feature representation is learnt jointly with user and item representations, this type of explanations is believed to be more relevant and informative. Recently, neural attentive models are also developed to directly rank user reviews for explanation [171].

4.1.2 Joint Tensor Factorization for Explainable Recommendation

In this section, we elaborate our multi-task learning solution for explainable recommendation. We exploit the opinionated review text data that users provide in addition to their overall assessments of the recommended items to enhance and explain the recommendation. Two companion learning tasks, i.e., *user preference modeling for recommendation* and *opinionated content modeling for explanation*, are integrated via a joint tensor factorization.

In the following discussions, we denote m, n, p, q as the number of users, items, features and opinionated phrases in a dataset, and a, b, c, d as the corresponding dimensions of latent factors for them in the learnt model. As a result, after factorization, users, items, features and opinion phrases can be represented by four non-negative matrices $U \in \mathbb{R}_+^{m\times a}$, $I \in \mathbb{R}_+^{n\times b}$, $F \in \mathbb{R}_+^{p\times c}$ and $O \in \mathbb{R}_+^{q\times d}$ in the latent factor space, respectively. Note that these four types of entities are associated with different degrees of complexity in practice, and therefore we do not restrict them to the same dimension of latent factors. To capture users' detailed feature-level opinions, we assume the existence of a domain-specific sentiment lexicon $\mathcal{L}$. Each entry of $\mathcal{L}$ takes the form of (feature, opinion, sentiment polarity), abbreviated as (f, o, s), to represent the sentiment polarity s inferred from an opinionated text phrase o describing feature f. Specifically, we label the sentiment polarity s as positive (+1) or negative (-1), but the developed solution can be seamlessly extended to multi-grade or continues rating cases. Based on this notation, our sentiment analysis is to map each user's review into a set of (f, o, s) entries. We use R_i^U and R_j^I to denote the set of reviews associated with user i and item j, respectively.

User Preference Modeling for Item Recommendation

This task is to predict the relevance of a recommendation candidate to a user, such that relevant candidates can be ranked higher. Traditional solutions perform the estimation by mapping users and items to a shared latent space via low-rank factorization over an input user-item affinity matrix [172, 173]. However, because this input matrix is usually constructed by users' overall assessment of items, such as clicks or ratings, the learnt factors cannot differentiate nor explain the detailed reason that a user likes/dislikes an item. To address this limitation, we focus on feature-level preference modeling for item recommendation. We aim to not only predict a user's overall assessment of an item, but also his/her preference on each feature of this item to enhance the recommendation.

Since different users would focus on different features of the same item, and even for the same feature of an item, different users might express distinct opinions on them, we use a three-way tensor $X \in \mathbb{R}_+^{m\times n\times p}$ to summarize such a high-dimensional relation. The key is to define the element X_{ijk} in this tensor, which measures to what extent user i appreciates item j's feature k reflected in his/her opinionated review set R_i^U. In this work, we adopt the method developed in [174] to construct a domain-specific sentiment lexicon $\mathcal{L}$ for analyzing users' detailed feature-level opinions. As the construction of a sentiment lexicon is not a contribution of this work and limited by space, we will not discuss the details of this procedure; interested readers can refer to [174, 175] for more details.

Based on the constructed sentiment lexicon $\mathcal{L}$, a user review can be represented as a list of (f, o, s) tuples. It is possible that a user mentions a particular feature multiple times in the same review but with phrases of different sentiment polarities. To denote the overall sentiment, we calculate the summation of all sentiment polarities that user i has expressed on item j's feature k. Suppose feature k is mentioned t_{ijk} times by user i about item j with the sentiment polarity labels $\{s_{ijk}^1, s_{ijk}^2, \ldots, s_{ijk}^{t_{ijk}}\}$, we define the resulting feature score as $\widehat{s}_{ijk} = \sum_{n=1}^{t_{ijk}} s_{ijk}^n$.

As we discussed in the introduction, a user's overall assessment of an item is usually a composition of multiple factors. In order to build the connection between a user's feature-level and overall assessments of an item, we introduce the overall assessment as a dummy feature to all items and append the overall rating matrix $A \in \mathbb{R}_+^{m \times n}$ to tensor X. This results in a new tensor $\widetilde{X} \in \mathbb{R}_+^{m \times n \times (p+1)}$. To normalize the scale between feature score $\widehat{s}_{ijk}$ and item overall rating A_{ij} in $\widetilde{X}$, we perform the following nonlinear mapping on its elements introduced by the feature scores,

$$\widetilde{X}_{ijk} = \begin{cases} 0, \text{if } f_k \text{ is not mentioned by } u_i \text{ about } i_j \\ 1 + \frac{N-1}{1+\exp(-\widehat{s}_{ijk})}, \quad \text{otherwise} \end{cases} \tag{4.1.1}$$

where N is the highest overall rating in the target domain.

Tensor $\widetilde{X}$ describes the observed affinity among users, items and features in a training data set. To predict unknown affinity among these three types of entities in testing time, we factorize $\widetilde{X}$ in a lower dimensional space to find the latent representation of these entities, and complete the missing elements in $\widetilde{X}$ based on the learnt representations. As we do not restrict these three types of entities to the same dimension of latent factors, we require a more flexible factorization scheme. Tucker decomposition [146, 176] best fits for this purpose, i.e.,

$$\begin{aligned} &\min_{\widehat{X}} \ ||\widehat{X} - \widetilde{X}||_F \\ &\text{s.t. } \widehat{X} = \sum_{r=1}^{a}\sum_{t=1}^{b}\sum_{v=1}^{c} g_{rtv}\mathbf{u}_r \otimes \mathbf{i}_t \otimes \mathbf{f}_v \\ &\quad \forall r, t, v \quad \mathbf{u}_r \geq 0, \mathbf{i}_t \geq 0, \mathbf{f}_v \geq 0, \text{and } g_{rtv} \geq 0 \end{aligned} \tag{4.1.2}$$

where $\mathbf{u}_r$ is the r-th column in the resulting user factor matrix U, $\mathbf{i}_t$ is the t-th column in the resulting item factor matrix I, $\mathbf{f}_v$ is the v-th column in the resulting feature factor matrix $\widetilde{F}$ (with dummy overall assessment feature expansion), $||\cdot||_F$ denotes the Frobenius norm over a tensor, and $\otimes$ denotes vector outer product. As we have mapped the feature scores to the same range of overall ratings in the target domain (i.e., $[1, N]$), we impose non-negative constraint over the learnt latent factors to avoid any negative predictions.

In Tucker decomposition, a core tensor $\mathcal{G} \in \mathbb{R}_+^{a \times b \times c}$ is introduced to describe how and to what extent different tensor elements interact with each other. This provides us another degree of freedom in performing the joint factorization in our multi-task learning solution. We will carefully elaborate this important advantage later when we discuss the detailed learning procedure in Section 4.1.2.

The resulting factor matrices U, I, and F are often referred to as the principal component in the respective tensor mode. And the unknown affinity among user i, item j and feature k can therefore be predicted by,

$$\widehat{X}_{ijk} = \sum_{r=1}^{a}\sum_{t=1}^{b}\sum_{v=1}^{c} g_{rtv}\mathbf{u}_{ri}\mathbf{i}_{tj}\mathbf{f}_{vk}. \tag{4.1.3}$$

The predicted user's feature-level assessment can already serve as a form of rating-based explanation [175]. In the next section, we will enhance our explanation to free text based, by learning from user-provided opinionated content about the items and features.

We should note recommendation is essentially a ranking problem, in which one needs to differentiate the relative relevance quality among a set of recommendation candidates. However, the current Tucker decomposition is

performed solely by minimizing element-wise reconstruction error, i.e., in Eq (4.1.2), which cannot directly optimize any ranking loss. To address this limitation, we introduce the Bayesian Personalized Ranking (BPR) principle [177] into our factorization of $\widetilde{X}$. Because we only have explicit user assessments at the item-level, we introduce the BPR principle in the overall rating predictions. In particular, for each user u_i we construct a pairwise order set D_i^S based on the observations about him/her in $\widetilde{X}$:

$$D_i^S := \left\{(j,l)|\, \widetilde{x}_{ij(p+1)} > \widetilde{x}_{il(p+1)}\right\}$$

where $\widetilde{x}_{ij(p+1)} > \widetilde{x}_{il(p+1)}$ indicates in the given review set R_i^U: 1) the user i gives a higher overall rating to item j than item l; or 2) item j is reviewed by user i while item l is not. Then the BPR principle can be realized by:

$$BPR\text{-}O_{PT} := -\lambda_{\mathcal{B}} \sum_{i=1}^{m} \sum_{(j,l)\in D_i^S} \ln \sigma\left(\widehat{x}_{ij(p+1)} - \widehat{x}_{il(p+1)}\right) \tag{4.1.4}$$

in which $\lambda_{\mathcal{B}}$ is a trade-off parameter and $\sigma(\cdot)$ is the logistic function. Intuitively, Eq (4.1.4) is minimized when all the pairwise ranking orders are maintained and the difference is maximized. By introducing it into the objective function of Eq (4.1.2), the decomposition is forced to not only reduce element-wise reconstruction error in $\widetilde{X}$, but also to confine with the pairwise ranking order between items.

Although we only impose ranking loss over the overall rating predictions in Eq (4.1.4), it also implicitly regularizes the feature-level predictions. To better understand this benefit, we can rewrite Eq (4.1.3) into a matrix product form,

$$\widehat{X}_{ijk} = \mathcal{G} \times_a U_i \times_b I_j \times_c \widetilde{F}_k \tag{4.1.5}$$

where $\mathcal{G} \times_n M$ denotes the n-mode product between tensor $\mathcal{G}$ and matrix M, i.e., multiply matrix M with each mode-n fiber in $\mathcal{G}$.

For a given pair of user i and item j, the first two n-mode product results in a matrix, denoted as T_{ij}, which presents a $(p+1) \times c$ dimensional space spanned by the latent factors for user i and item j. The feature scores and overall ratings are predicted by projecting the feature factors, i.e., matrix $\widetilde{F}$, onto it. To satisfy the BRP principle in Eq (4.1.4), T_{ij} has to be adjusted for each pair in D_i^S. As $\widetilde{F}$ is globally shared across users and items, this introduces the pairwise ranking loss into the gradient for all features' latent factor learning; this effect becomes more evident when we introduce the learning procedures for our joint factorization later in the Section 4.1.2.

Opinionated Content Modeling for Explanation

If an algorithm could predict the opinionated content that the user would provide on the recommended item, it is an informative explanation to reveal why the user should pay attention to those features of the recommendation. Based on this principle, we develop a companion learning task of opinionated content modeling to generate detailed textual explanations for the recommendations.

With the factorization scheme discussed in the last section, a straightforward solution for content modeling is to create a four-way tensor to summarize the complex relations among users, items, features, and opinion phrases. However, this four-way tensor would be extremely sparse in practice, as an ordinary user would only comment on a handful of items and we cannot expect their comments to be exhaustive. It is known that in natural language the distribution of words is highly skewed, e.g., Zipf's law [178]; we hypothesize that the distribution of opinion phrases that an item often receives for describing its features, and that a user often uses to describe a type of items' features are also highly skewed. In other words, the appearance of an opinion phrase towards a feature should strongly depend on the user or the item. Therefore, we approximate the four-way tensor by two three-way tensors, one summarizes the relation among user, feature and opinion phrase, and another for item, feature and opinion phrase.

This approximation is also supported by prior studies in mining opinionated text data. Amarouche [179] specifies that opinion phrase associated with a feature is apparently dependent on the opinion holder (user) as well as the target object (item) in product opinion mining. Kim and Hovy [180] focus on the importance of the opinion holder, explaining that the opinion holder's identification can be used independently to answer

several opinion questions. Ronen and Moshe [181] compare products on their features/attributes by mining user-generated opinions, and report the dependence of opinions on different products features/attributes. In our experiments, we also empirically confirmed our hypothesis for approximation via a permutation test on two large review data sets.

We denote the first tensor as $Y^U \in \mathbb{R}_+^{m\times p\times q}$. From the review set R_i^U of user i, we extract all positive phrases this user has used to describe feature k across all items, i.e., $\mathcal{R}_{i,k}^U = \big\{o|\,(f,o,s) \in R_i^U, f = k, s = +1\big\}$. We only include positive phrases, as we need to explain why a user should appreciate the feature of a recommended item, rather than avoid it; otherwise we should not recommend this item or feature at all. But our algorithm can be easily extended to the scenario where one needs to provide warning messages (e.g., include the negative phrases in the tensor). To reflect the frequency of user i uses opinion phrase o to describe feature k, and to facilitate the joint factorization later, we construct Y^U as,

$$Y_{ikw}^U = \begin{cases} 0, & \text{if } w \text{ is not in } \mathcal{R}_{i,k}^U \\ 1 + (N-1)\big(\frac{2}{1+\exp(-\Gamma)} - 1\big), & \text{otherwise} \end{cases} \tag{4.1.6}$$

where Γ is the frequency of phrase w in $\mathcal{R}_{i,k}^U$.

We construct the second tensor $Y^I \in \mathbb{R}_+^{n\times p\times q}$ in a similar way. For item j, we first obtain a collection of positive phrases about its feature k from R_j^I, i.e., $\mathcal{R}_{j,k}^I = \big\{o|(f,o,s) \in R_j^I, f = k, s = +1\big\}$, and then construct Y^I as:

$$Y_{jkw}^I = \begin{cases} 0, & \text{if } w \text{ is not in } \mathcal{R}_{j,k}^I \\ 1 + (N-1)\big(\frac{2}{1+\exp(-\Omega)} - 1\big), & \text{otherwise} \end{cases} \tag{4.1.7}$$

where Ω is the frequency of phrase w in $\mathcal{R}_{j,k}^I$.

The construction of tensor $\widetilde{X}$, Y^U and Y^I impose strong dependency between the two learning tasks of item recommendation and opinionated explanation, as every two tensors share the same two types of entities (as shown in Figure 4.2). Instead of isolating the factorization of these three tensors, we propose a joint factorization scheme, which will be discussed in detail in the next section. Once the latent factors are learnt, we can predict user i's opinionated comments on feature k by the reconstructed vector $\widehat{Y}_{i,k}^U$, which can be calculated in the same way as in Eq (4.1.3) with the corresponding latent factors. Similarly, the opinionated comments that item j will receive on its feature k can be predicted by the reconstructed vector $\widehat{Y}_{j,k}^I$. As a result, to predict the opinionated comments that user i will provide on item j's feature k, we take an element-wise product between these two vectors to construct an opinion phrase scoring vector $\widehat{Y}_{i,j,k}^{U,I}$, in which each element is computed as,

$$\widehat{Y}_{i,j,k,w}^{U,I} = \widehat{Y}_{i,k,w}^U \times \widehat{Y}_{j,k,w}^I \tag{4.1.8}$$

This estimation reflects our approximation of the original four-way tensor with two three-way tensors. Because the tensor Y^U and Y^I record the frequency of an opinion phrase used in describing the features by the user and about the item, Eq (4.1.8) prefers to choose those that are popularly used to describe this feature of the item in general, and also by this target user to describe this feature in similar items.

Multi-task Learning via a Joint Tensor Factorization

Both of our proposed learning tasks are modeled as a tensor factorization problem, and they are coupled with the shared latent factors. Ideally, the predicted users' assessment about the recommendation candidates from the first task should be supported by the predicted users' opinionated comments about the recommendations from the second task. To leverage the dependency between these two tasks, we develop a joint factorization scheme.

In Tucker decomposition, a three-way input tensor will be decomposed into a core tensor and three principle component matrices. The core tensor captures multivariate interactions among the latent factors; and the principle component matrices can be viewed as basis of the resulting latent space. Based on this property, we decide to share the principle component matrices across the three tensors of $\widetilde{X}$, Y^U and Y^I to learn the latent representations of user, item, feature and opinion phrases across the two learning tasks, and keep independent

core tensors for these tensors to capture the tasks' intrinsic variance and scaling of the shared latent factors. As a result, we devise the following joint optimization formulation,

$$\begin{aligned}\min_{\widehat{X},\widehat{Y}^U,\widehat{Y}^I} & ||\widehat{X}-\widetilde{X}||_F + ||\widehat{Y}^U - Y^U||_F + ||\widehat{Y}^I - Y^I||_F - \lambda_{\mathcal{B}} \sum_{i=1}^{m} \sum_{(j,l)\in D_i^S} \ln \sigma\big(\widehat{x}_{ij(p+1)} - \widehat{x}_{il(p+1)}\big) \\ & + \lambda_{\mathcal{F}}\big(||U||^2 + ||I||^2 + ||F||^2 + ||O||^2\big) + \lambda_{\mathcal{G}}\big(||\mathcal{G}_1||^2 + ||\mathcal{G}_2||^2 + ||\mathcal{G}_3||^2\big) \\ \text{s.t.} \quad & \widehat{X} = \mathcal{G}_1 \times_a U \times_b I \times_c \widetilde{F}, \\ & \widehat{Y}^U = \mathcal{G}_2 \times_a U \times_c F \times_d O, \\ & \widehat{Y}^I = \mathcal{G}_3 \times_b I \times_c F \times_d O, \\ & U \geq 0, I \geq 0, F \geq 0, O \geq 0, \mathcal{G}_1 \geq 0, \mathcal{G}_2 \geq 0, \mathcal{G}_3 \geq 0. \end{aligned} \tag{4.1.9}$$

where we introduce l_2 regularization over the learned latent factor matrices and core tensors to avoid over-fitting. This joint factorization ensembles the two companion learning tasks for recommendation and explanation, i.e., multi-task learning; and therefore, we name our solution as Multi-Task Explainable Recommendation, or MTER in short.

The above optimization problem can be effectively solved by stochastic gradient descent (SGD), with projected gradients for non-negative constraints. However, because the number of observations in each tensor and in the pairwise ranking constraint set varies significantly, vanilla SGD procedure suffers from local optimum. To improve the convergence, we randomly select small batches of samples from each tensor and pairwise constraint set per iteration to calculate a averaged gradient, i.e., a mini-batch SGD. And to avoid manually specifying a step size, we employ adaptive gradient descent [182], which dynamically incorporates the updating trace in earlier iterations to perform more informative and faster gradient-based learning. The parameter estimation procedure of our model is off-line, and large-scale learning tasks could be solved within reasonable periods.

Interactions between the two learning tasks become more evident in MTER when we look into the detailed gradients for model update. Denote the objective function in Eq (4.1.9) as L, and we list the gradient of F_k as an example to illustrate how the elements in three tensors $\widetilde{X}$, Y^U and Y^I contribute to it:

$$\frac{\partial L}{\partial_{F_k}} = \frac{\partial L}{\partial_{\widehat{X}_{ijk}}} \mathcal{G}_1 \times_a U_i \times_b I_j + \frac{\partial L}{\partial_{\widehat{Y}^U_{ikl}}} \mathcal{G}_2 \times_a U_i \times_d O_w + \frac{\partial L}{\partial_{\widehat{Y}^I_{jkl}}} \mathcal{G}_3 \times_b I_j \times_d O_w \tag{4.1.10}$$

In Eq (4.1.10), as F_k is shared across the decomposition of all three tensors, it bridges the other three components U_i, I_j and O_w in these two tasks. Similarly, the gradient of U_i, I_j and O_w also involves all the rest factors. Furthermore, the BPR constraint introduced on overall rating prediction indirectly affects the learning of U_i, I_j and O_w, via gradient sharing. This also helps MTER conquer data sparsity issue when we have a large number of users, items, features and opinionated phrases to model.

4.1.3 Experiment

In this section, we quantitatively evaluate our solution MTER in the tasks of item recommendation and opinionated content modeling, on two popular benchmark datasets collected from Amazon[1] [183, 184] and Yelp Dataset Challenge[2]. We perform extensive comparisons against several state-of-the-art recommendation and explainable recommendation algorithms. Improved quality in both recommendation and opinionated content prediction confirms the comprehensiveness and effectiveness of our solution.

[1] http://jmcauley.ucsd.edu/data/amazon/
[2] https://www.yelp.com/dataset

Table 4.1: Basic statistics of evaluation datasets.

Dataset	#users	#items	#features	#opinions	#reviews
Amazon	6,285	12,626	95	591	55,388
Yelp	10,719	10,410	104	1,019	285,346

Experiment Setup

Datasets and Preprocessing. To verify our model's effectiveness in different application domains, we choose *restaurant* businesses from Yelp dataset and *cellphones and accessories* category from Amazon dataset. These two datasets are very sparse: 73% users and 47% products only have one review in Amazon dataset, and 54% users only have one review in Yelp dataset. However, in the constructed sentiment lexicons, 401 features are extracted from Amazon dataset, and 1065 are extracted from Yelp dataset. It is very difficult to estimate the affinity between users and those hundreds of features from only a handful of reviews. To refine the raw datasets, we first analyze the coverage of different features in these two datasets. Within the sentiment lexicon, only a small number of features (around 15%) that are frequently covered in 90% reviews, while most of features occur rarely in the whole datasets (i.e., Zipf's law). As a result, we perform recursive filtering to alleviate the sparsity issue. We only select the features whose support is above a pre-defined threshold. With the subset features, we further filter out reviews associated with too few features and items associated with too few reviews. By fine tuning these different thresholds, we obtain two refined datasets with decent amount of users and items, whose basic statistics are reported in Table 4.1.

Baselines. To evaluate the effectiveness of our proposed explainable recommendation solution, we include the following recommendation algorithms as baselines:

- **MostPopular:** A non-personalized recommendation solution. Items are ranked by their observed frequency in the training dataset.
- **NMF:** Nonnegative Matrix Factorization [185], which is a widely applied latent factor model for recommendation.
- **BPRMF:** Bayesian Personalized Ranking on Matrix Factorization [177], which introduces BPR pairwise ranking constraint into factorization model learning (as shown in Eq (4.1.4)).
- **JMARS:** A probabilistic model that jointly models aspects, ratings, and sentiments by collaborative filtering and topic modeling [168].
- **EFM:** Explicit Factor Models [175]. A joint matrix factorization model for explainable recommendation, which considers user-feature attention and item-feature quality.
- **MTER-S(SA):** Replace Tucker decomposition with canonical decomposition [146] in our MTER solution. With canonical decomposition, each decomposed matrix is represented as a summation of a shared component and a local component. For example, user factor matrix is then represented as $U_0 + U_1$, where U_0 is shared across three tensors and U_1 is only estimated for $\widetilde{X}$. Similar decomposition structure design can be found in [149, 175]. If we do not allow the local components and assume all components are shared across tensors, we can get another variant of MTER, named as MTER-SA.

Evaluation Metric. We use Normalized Discounted Cumulative Gain (NDCG) to evaluate top-k recommendation performance. 80% of each dataset is used for training, 10% for validation and 10% for testing respectively. We use grid search to find the optimal hyper parameters in a candidate set for all baseline models.

Experiment Results

Performance of Recommendation. We report the recommendation performance of each model measured by NDCG@{10,20,50,100} in Table 4.2. Paired t-test is performed between the best and second best (MTER-S(SA) excluded) performing algorithms under each metric to confirm the significance of improvement.

Table 4.2: Comparison of recommendation performance.

NDCG @K	Amazon						
	FMF [186]	NMF [187]	BPRMF [188]	JMARS [189]	EFM [143]	MTER	FacT
10	0.1009	0.0649	0.1185	0.1064	0.1109	**0.1351**	**0.1482**
20	0.1331	0.0877	0.1490	0.1348	0.1464	**0.1653**	**0.1795**
50	0.1976	0.1601	0.2070	0.1992	0.2056	**0.2234**	**0.2367**
100	0.2529	0.2144	0.2669	0.2575	0.2772	**0.2803**	**0.2869**
NDCG@K	Yelp						
	FMF [186]	NMF [187]	BPRMF [188]	JMARS [189]	EFM [143]	MTER	FacT
10	0.0931	0.0564	0.1266	0.1155	0.1071	**0.1380**	**0.1499**
20	0.1243	0.0825	0.1643	0.1553	0.1354	**0.1825**	**0.1991**
50	0.1871	0.1345	0.2214	0.2111	0.1903	**0.2365**	**0.2488**
100	0.2509	0.2175	0.2668	0.2575	0.2674	**0.2783**	**0.2867**

* p-value < 0.05

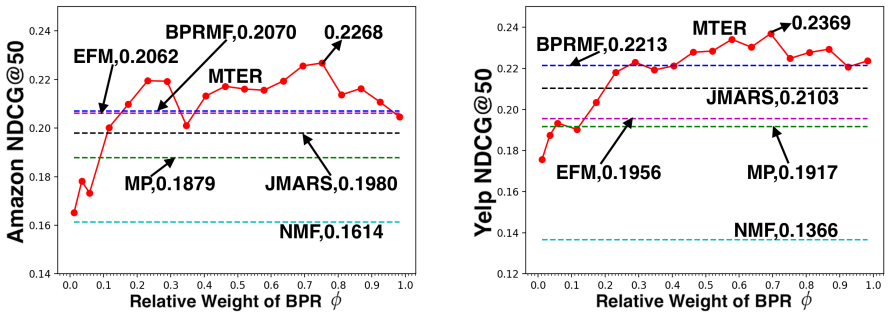


Figure 4.3: NDCG@50 $vs.$ relative weight ϕ of BPR on Amazon and Yelp datasets.

Results in Table 4.2 clearly demonstrate the advantage of MTER over the baselines. First, straightforward factorization algorithm (i.e., NMF) cannot optimize the ranking quality of the recommended items, and its performance is even worse than a simple popularity based solution, which provides generic recommendations to all users. The pairwise ranking constraints introduced by BPR greatly improve the recommendation effectiveness of BPRMF, which shares the same decomposition structure as in NMF. However, as BPRMF only models users' overall assessment on items, it cannot exploit information available in the user-provided opinionated content. Hence, its performance is generally worse than MTER and its variants. Second, comparing to JMARS and EFM, which also utilize review content for recommendation, MTER is the only model that outperforms BPRMF. JMARS models all entities in a shared topic space, which limits it resolution in modeling complex dependencies, such as users v.s., items, and users v.s., features. EFM implicitly integrates the interaction among users, items and features via three loosely coupled matrices, and it is only optimized by the reconstruction error on those three matrices. This greatly limits its recommendation quality. Third, by comparing different variants of MTER, we can recognize the advantage of Tucker decomposition in this multi-task learning setting. Because MTER-SA forces everything to be shared across three tensors, it fails to recognize task-specific variance. MTER-S enables task-specific learning, but it requires all entities to share the same dimension of latent factors. As we have observed when preprocessing the two datasets, different types of entities are associated with different number of observations, and therefore they consist of different degrees of intrinsic complexity. Forcing the latent factors to share the same structure cannot capture such intrinsic complexity, and therefore leads to sub-optimal recommendation performance. In addition, we can also observe that the best improvement from MTER is achieved at NDCG@10 (more than 15% against the best baseline on Amazon and over 11% on Yelp). This result is significant: it indicates a system equipped with MTER can provide satisfactory results earlier down the ranked list, which is crucial in all practical recommender systems.

Contribution of BPR. As the influence of BPR in our MTER training is related to both the number of pairwise

constraints selected per iteration and the trade-off coefficient λ_B, we define a relative normalized weight of BPR to analyze its contribution in our model training:

$$\phi = \frac{\lambda_B \times N_{S_{BPR}} \times T_{iter}}{m \times n^2} \tag{4.1.11}$$

where T_{iter} is the the number of iterations, $N_{S_{BPR}}$ is the number of pairwise constraints sampled for BPR in each iteration, and $m \times n^2$ is the number of all pairwise samples from a dataset of m users and n items [177]. We fix $N_{S_{BPR}}$ and tune λ_{BPR} for optimization.

We evaluate NDCG@50 on an increasing weight ϕ for BPR, while keeping all the other hyper-parameters as constant. The result is shown in Figure 4.3. We can find that when ϕ is small, the tensor reconstruction error dominates our model learning and thus its ranking performance is worse than most baselines. But thanks to the additional information introduced in opinionated reviews, MTER is still better than NMF, which is purely estimated by the reconstruction error of the overall rating matrix. With an increased ϕ, the pairwise ranking constraints help our model identify better latent factors that differentiate users' preferences over different items, which in turn lead to better modeling of dependency among users, items, features and opinionated phrases (as shown in Eq (4.1.10)). However, if ϕ goes beyond a certain threshold, the ranking quality degenerates. This is also expected: as the pairwise constraints dominate factor learning, the quality of content modeling task will be undermined, and it also increases the risk of over-fitting.

4.2 Explainable Recommendation with Factorization Tree

MTER incorporate the phrase-level sentiment analysis into latent factor learning, e.g., joint tensor factorization, for explanations. Basically, MTER maps users' feature-level opinions into the latent space and finds the most related features to the users and recommended items as explanations. However, to what extent these approximated explanations comply with the learned latent factor models is unknown, i.e., no guarantee in explanation fidelity. The fidelity of explanation and the quality of recommendation have long been considered as irreconcilable [29]: one has to trade recommendation quality for explanation. For example, it is believed that content-based collaborative filtering algorithms are easy to explain, as their underlying recommendation mechanism is straightforward. Nevertheless, we believe the tension between recommendation quality and explanation fidelity is not necessarily inevitable. We can attain both by optimizing the recommendation in accordance with the designed explanation mechanism.

In this work, we aim at explaining latent factor based recommendation algorithms with rule-based explanations. Our choice is based on the facts that 1) latent factor models have proved their effectiveness in numerous practical deployments [144, 190], and 2) prior studies show that rule-based explanations are easy to perceive and justify by the end-users [35]. However, as the latent factors are not learned by rules, it is hard to craft any rules to explain the factors afterwards. Therefore, we propose to integrate the rule-based decision making into the learning of latent factors to provide explanations for the recommendation. In particular, we treat the latent factors as a function of the rules: based on different outcome of the rules, the associated groups of users and instances should be routed to the designated latent factors, which are then optimized for recommendation. Due to similar characteristics shared by each group of users/instances created by the learned rules, the descriptive power of the learned *group-level* latent factors is enhanced, and the data sparsity problem in individual users/instances could be substantially alleviated by this group-level latent factor learning.

More specifically, we format the explanation rules based on feature-level opinions extracted from user-generated review content, e.g., whether a user holds positive opinion towards a specific feature. The rules are extracted by inductive learning on the user side and item side separately, which form a user tree and an item tree. We alternate the optimization between tree construction and latent factor estimation under a shared recommendation quality metric. An example of user tree is shown in Figure 4.4. For instance, according to the figure, if two users both expressed their preference of "*burger*" in their reviews, they should be assigned to the same node on the user tree to share the latent user factors; accordingly, if two restaurants receive similar negative comments about their "*cleanliness*", they should appear in the same node on the item tree. In testing time, the learned user and item factors are used for recommendation as in standard latent factor models, and the rules that lead to the chosen user and item factors are output as explanations: e.g., "*We recommend item X because it matches your preference on burger and cleanness of a restaurant.*"

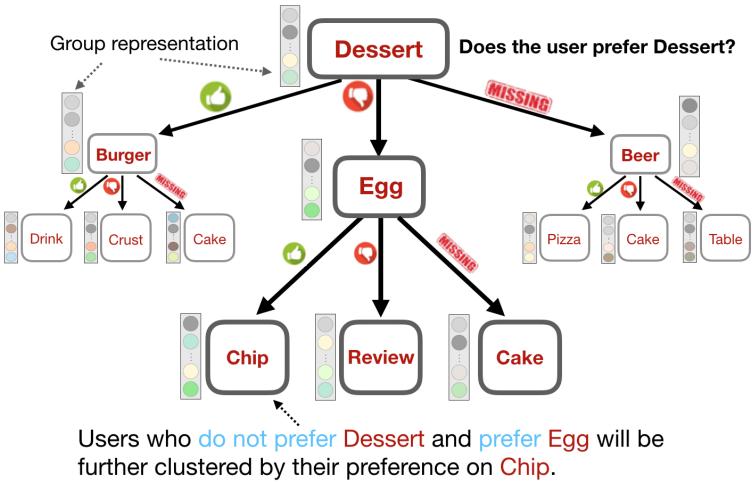


Figure 4.4: An example user tree: Top three levels of our FacT model learned for restaurant recommendations.

Extensive experiment evaluations on two large sets of reviews, i.e., Amazon reviews for product recommendation and Yelp reviews for restaurant recommendation, demonstrate improved quality in recommendation and explanation from our algorithm, compared with a set of state-of-the-art explainable recommendation algorithms. In particular, we perform serious user studies to investigate the utility of our explainable recommendation in practice, in both warm-start and cold-start settings. Positive user feedback further validates the value of our proposed solution.

4.2.1 Related Work

The idea of providing rule-based explanations was popularized in the development of expert systems [191, 192]. For example, MYCIN [193], a rule-based reasoning system, provides explanations by translating traces of rules followed from LISP to English. A user could ask both why a conclusion was arrived at and how much was known about a certain concept. But since modern recommender systems seldom use rule-based reasoning, there is very little research on explaining latent factor models with rules. We propose to embed latent factor learning under explanation rule learning, by treating the latent factors as a function of rules, such that the generated explanations can strictly adhere to the provided recommendations. On a related note, a existing work [194] uses gradient boosting decision trees (GBDT) to learn rules from the reviews and incorporate rules into an attention network. But it only uses the rules as the input of embedding models and thus isolates the learning of tree structure and embeddings. Some systems [186, 195] combine decision tree learning with matrix factorization to extract a list of interview questions for solving the cold-start problem in recommendation. But the rules are only built on the user side with their rating responses to items, i.e., the same as matrix factorization's input; it thus cannot provide any explanation to the recommended items.

4.2.2 Taming Latent Factor Models with Factorization Tree

We elaborate our solution for joint latent factor learning and explanation rule construction in this section. Briefly, we model the latent factors for both users and items as a function of the rules: users who provide the same responses to the same set of rules would share the same latent factors, and so do the items. The predicates of rules are selected among the text features extracted from user-generated reviews. For example, whether a specific user expressed his/her preference on a particular feature in reviews. And the rules are constructed by recursive inductive learning based on previously selected predicate's partition of users and items. To reflect the heterogeneity between users and items, we construct rules for users and items separately. As a result of rule induction, the latent factors for users and items are organized in a decision tree like structure accordingly, where each node on the tree represents the latent factors for the group of users or items routed to that node. We alternate the optimization of the explanation rule construction and latent factor learning

under a recommendation quality based metric. Hence, we name our solution as *Factorization Tree*, or *FacT* in short.

We start our discussion with factorization based latent factor learning, which is the basic building block of FacT. Then we provide details in rule induction based on the learned latent factors. Finally, we integrate these two learning components with an alternative optimization procedure.

Latent Factor Learning

Latent factor models [144, 190] have been widely deployed in modern recommender systems. The idea behind this family of solutions is to find vectorized representations of users and items in a lower dimensional space, which capture the affinity between users and items. Various latent factor models have been developed, such as matrix/tensor factorization [190] and factorization machines [144]. Our FacT is independent of the choice of latent factor models, as it treats the latent factor learning as a sub-routine.

Formally, denote $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ as a set of m users, $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ as a set of n items, and r_{ij} as an observed rating from user i to item j. The goal of latent factor learning is to associate each user and each item an d dimensional vector respectively, i.e., $u_i \in \mathbb{R}^d$ and $v_j \in \mathbb{R}^d$, such that the inner product between user i's factor and item j's factor predicts the rating r_{ij}. The latent factors for all the users and items, denote as $U \in \mathbb{R}^{m \times d}$ and $V \in \mathbb{R}^{n \times d}$, can thus be learned by minimizing their prediction error over a set of observed ratings $O = \{(i, j)|r_{ij}\, is\, observed\}$ as follows,

$$\mathcal{L}(U, V, O) = \min_{U,V} \sum_{(i,j)\in O} (r_{ij} - u_i^\top v_j)^2. \tag{4.2.1}$$

It is well accepted that recommendation is essentially a ranking problem [188, 196]; however, the objective function introduced in Eq (4.2.1) cannot fully characterize the need of ranking, i.e., differentiate the relative quality among candidates. To supplement information about relative item ranking into latent factor learning, Bayesian Personalized Ranking (BPR) loss [188] has been popularly adopted to enforce pairwise ranking order. To realize the BPR loss, one needs to first construct a pairwise ordered set of items D_i^o for each user i: $D_i^o := \{(j, l)|\, r_{ij} > r_{il}\}$, where $r_{ij} > r_{il}$ means that given the observations in O, either user i gives a higher rating to item j than item l, or item j is observed in user i's rating history, while item l is not. Then, the BPR loss can be measured on each user i as:

$$\mathcal{B}(u_i, V, D_i^o) = \sum_{(j,l)\in D_i^o} \log \sigma(u_i^\top v_j - u_i^\top v_l)$$

where $\sigma(\cdot)$ is a logistic function.

Putting together the pointwise rating prediction loss with the pairwise ranking loss, the latent factors for users and items can be learned by solving the following optimization problem:

$$(\widehat{U}, \widehat{V}) = \operatorname*{argmin}_{U,V} \mathcal{L}(U, V, O) - \lambda_b \sum_i \mathcal{B}(u_i, V, D_i^o) + \lambda_u \|U\|_2 + \lambda_v \|V\|_2 \tag{4.2.2}$$

where λ_b is a trade-off parameter to balance these two types of loss, $\|U\|_2$ and $\|V\|_2$ are l_2 regularization to control model complexity, and λ_u and λ_v are the corresponding coefficients. Eq (4.2.2) can be efficiently addressed by gradient-based optimization [197]. Once the user factors U and item factors V have been learned, the recommendations for user i can be generated by returning the top ranked items based on the predicted ratings $\widehat{r}_{ij} = \widehat{u}_i^\top \widehat{v}_j$.

The premise behind the aforementioned learning procedure is that there is only a small number of factors influencing users' preferences, and that a user's preference vector is determined by how each factor applies to that user and associated items. But the factors are retrieved by solving a complex optimization problem (e.g., Eq (4.2.2)), which makes the resulting recommendations hard to explain. In FacT, we embed latent factor learning under explanation rule construction, so that why a user or an item is associated to a particular latent factor can be answered by the matched rules, so do the generated recommendations.

Explanation Rule Induction

In FacT, we consider the latent factors as a function of explanation rules: the latent user factor u_i for user i is tied to the outcomes of a set of predicates applied to him/her, so does the latent item factor v_j for item j. Based on different outcomes of the rules, the associated groups of users and items should be routed to the designated latent factors. At testing time, the activated predicates on user i and item j naturally become the explanation of this recommendation.

We select the predicates among the item features extracted from user-generated reviews. User reviews provide a fine-grained understanding of a user's evaluation of an item [198]. Feature-level sentiment analysis techniques [34] can be readily applied to reviews to construct a domain-specific sentiment lexicon. Each lexicon entry takes the form of (feature, opinion, sentiment polarity), abbreviated as (f, o, s), and represents the sentiment polarity s inferred from an opinionated text phrase o describing feature f. Specifically, we label the sentiment polarity s as +1 or -1, to represent positive or negative opinions. As how to construct a sentiment lexicon with phrase-level sentiment analysis is not the focus of this work, we refer interested readers to [34, 143] for more details.

The extracted item features become candidate variables for predicate selection. To compose predicates for explanation rule construction, we first need to define the evaluation of a single variable predicate on users/items according to their association with the item features. To respect the heterogeneity between users and items, we construct the predicates for users and items separately; but the construction procedures are very similar and highly related on these two sides.

Denote $\mathcal{F} = \{f_1, f_2, ..., f_k\}$ as a set of k extracted item features. Suppose feature f_l is mentioned by user i for p^u_{il} times with a positive sentiment polarity in his/her reviews and n^u_{il} times with a negative sentiment polarity. We can construct a feature-level profile F^u_i for user i, where each element of F^u_i is defined as,

$$F^u_{il} = \begin{cases} \emptyset, & \text{if} \quad p^u_{il} = n^u_{il} = 0, \\ p^u_{il} + n^u_{il}, & \text{otherwise.} \end{cases} \tag{4.2.3}$$

Intuitively, F^u_{il} is the frequency of user i mentioning feature f_l in his/her reviews, such that it captures the relative emphasis that he/she has given to this feature. And similarly, on the item side, denote p^v_{jl} as the number of times that feature f_l is mentioned in all user-generated reviews about item j with a positive sentiment polarity, and n^v_{jl} as that with a negative sentiment polarity, we define the feature-level profile F^v_j for item j as,

$$F^v_{jl} = \begin{cases} \emptyset, & \text{if} \quad p^v_{jl} = n^v_{jl} = 0, \\ p^v_{jl} - n^v_{jl}, & \text{otherwise.} \end{cases} \tag{4.2.4}$$

Accordingly, F^v_{jl} reflects the aggregated user sentiment evaluation about feature f_l of item j.

Based on the feature-level user and item profiles, the evaluation of a single variable predicate can be easily performed by comparing the designated feature dimension in the user or item profile against a predefined threshold. For example, on the user side, if a predicate is instantiated with feature f_l and threshold t^u_l, all users can have three disjoint responses to this predicate based on their F^u_{il} values, i.e., $F^u_{il} \geq t^u_l$, or $F^u_{il} < t^u_l$, or F^u_{il} is unknown. This gives us the opportunity to model the latent factors as a function of the explanation rules: based on the evaluation results of a predicate, we allocate the input users into three separate user groups and assign one latent vector per group. We should note that other forms of predicates are also applicable for our purpose, e.g., select a list of thresholds or a nonlinear function for one variable. For simplicity, we adhere to the form of single threshold predicates, and leave the more complex forms of predicates for future exploration.

Two questions remain to be answered: First, how to select the threshold for user-side and item-side predicate creation; and second, how to assign latent vectors for each resulting user/item group. We answer the first question in this section by inductive rule learning, and leave the second to the next section, where we present an alternative optimization procedure for joint rule learning and latent factor learning. In the following discussions, we will use user-side predicate construction as an example to illustrate our rule induction method; and the same procedure directly applies to item-side predicate construction.

Intuitively, an optimal predicate should create a partition of input users where the latent factors assigned to each resulting user group lead to minimal recommendation loss defined in Eq (4.2.2). This can be achieved by exhaustively searching through the combination of all item features in $\mathcal{F}$ and all possible corresponding thresholds. This seems infeasible at a first glance, as the combinatorial search space is expected to be large. But in practice, due the sparsity of nature language (e.g., Zipf's law [199]), the mentioning of item features and its frequency in user reviews are highly concentrated at both user-level and item-level [200]. Besides, feature discretization techniques [201] can also be used to further reduce the search space.

To perform the search for optimal predicate in an input set of users $\mathcal{U}_a$, we first denote the resulting partitions of $\mathcal{U}_a$ by feature f_l and threshold t_l^u as,

$$\begin{aligned} L(f_l, t_l^u|\mathcal{U}_a) &= \{i|F_{il}^u \geq t_l^u, i \in \mathcal{U}_a\}, \\ R(f_l, t_l^u|\mathcal{U}_a) &= \{i|F_{il}^u < t_l^u, i \in \mathcal{U}_a\}, \\ E(f_l, t_l^u|\mathcal{U}_a) &= \{i|F_{il}^u = \emptyset, i \in \mathcal{U}_a\}, \end{aligned} \tag{4.2.5}$$

and the set of possible threshold t_l^u for feature f_l as T_l^u. The optimal predicate on $\mathcal{U}_a$ can then be obtained by solving the following optimization problem with respect to a given set of item factors V,

$$\begin{aligned} (\bar{f}_l, \bar{t}_l^u) = \underset{f_l \in \mathcal{F}, t_l^u \in T_l^u}{\operatorname{argmin}} \min_{u_L, u_R, u_E} & \mathcal{L}(u_L, V, O_L) - \lambda_b \sum_{i \in E(f_l, t_l^u)} \mathcal{B}(u_L, V, D_i^o) \\ & + \mathcal{L}(u_R, V, O_R) - \lambda_b \sum_{i \in R(f_l, t_l^u)} \mathcal{B}(u_R, V, D_i^o) \\ & + \mathcal{L}(u_E, V, O_E) - \lambda_b \sum_{i \in E(f_l, t_l^u)} \mathcal{B}(u_E, V, D_i^o) \\ & + \lambda_u(\|u_L\|_2 + \|u_R\|_2 + \|u_E\|_2) \end{aligned} \tag{4.2.6}$$

where O_L, O_R and O_E are the observed ratings in the resulting three partitions of $\mathcal{U}_a$, and u_L, u_R and u_E are the correspondingly assigned latent factors for the users in each of the three partitions. As users in the same partition are forced to share the same latent factors, the choice of text feature f_l and corresponding threshold t_l^u directly affect recommendation quality. In practice, considering each user and item might associate with different number of reviews, the size of user profile F_i^u and item profile F_j^v might vary significantly. Proper normalization of F_i^u and F_j^v can be performed, e.g., normalize by the total observation of feature mentioning in each user and item respectively. In this work, we follow [201] for feature value normalization and discretization.

Inside the optimization of Eq (4.2.6), a sub-routine of latent factor learning is performed to minimize recommendation loss induced by matrix factorization (as defined in Eq (4.2.2)) on the resulting partition of users. As we mentioned before, the choice of latent factor models does not affect the procedure of our predicate construction for FacT, and many other recommendation loss metrics or latent factor models can be directly plugged into Eq (4.2.2) for explainablity enhancement. We leave this exploration as our future work.

Our predicate construction can be recursively applied on the resulting user partitions $L(\bar{f}_l, \bar{t}_l^u|\mathcal{U}_a)$, $R(\bar{f}_l, \bar{t}_l^u|\mathcal{U}_a)$ and $E(\bar{f}_l, \bar{t}_l^u|\mathcal{U}_a)$ on the input user set $\mathcal{U}_a$ to extend a single variable predicate to a multi-variable one, i.e., inductive rule learning. The procedure will be terminated when, 1) the input user set cannot be further separated, e.g., all users there share the same user profile; or 2) the maximum depth has been reached. Starting the procedure from the complete set of users $\mathcal{U}$, the resulting set of multi-variable predicates form a decision tree like structure, which we refer as user tree in FacT (as shown in Figure 4.4). On the user tree, each node hosts a latent factor assigned to all its associated users, and its path to the root node presents the learned predicates for this node. The same procedure can be applied on the item side with a given set of user factors U to construct item-specific predicates, i.e., item tree.

Once the user tree and item have been constructed, explaining the recommendations generated by the latent factors becomes straightforward. Assume we recommend item j to user i: we first locate user i and item j at the leaf nodes of user tree and item tree accordingly, extract their paths back to each tree's root node, and

find the shared features on the paths to create feature-level explanations. As each branch on the selected path corresponds to a specific outcome of predicate evaluation, e.g., Eq (4.2.5), we can add predefined modifiers in front of the selected features to further elaborate the associated latent factors. For example,

- *We recommend this item to you because its [good/excellent] [feature 1] matches with your [emphasize/taste] on [feature 1], and ...*
- *We guess you would like this item because of your [preference/choice] on [feature 1], and ...*

It is also possible that the number of shared features on the two paths is low, especially when the maximum tree depth is small. In this situation, one can consider to use the union of features on these two paths, and give higher priority to the shared features and those at the lower level of the trees, as they are more specific. Another possible way of explanation generation is to use the selected features to retrieve sentences from the corresponding item reviews [171].

Alternative Optimization

The aforementioned procedure for explanation rule induction is intrinsically recursive and requires the availability of user factors for item tree construction and item factors for user tree construction. In this section, we will unify the learning of latent factors with tree construction to complete our discussion of FacT.

Define the maximum rule length, i.e., tree depth, as h. We alternate rule induction by recursively optimizing Eq (4.2.6) between user side and item side. At iteration t, we start induction from the complete user set $\mathcal{U}$ with the latest item factors V_{t-1}. For each pair of feature and threshold in the hypothesis space of Eq (4.2.6), we use gradient based optimization for learning latent factors according to Eq (4.2.2). Once the induction finishes, we collect the latent user factors U_t from the leaf nodes of the resulting user tree, and use them to execute the rule induction on the item side from the complete item set $\mathcal{V}$ to estimate V_t. This procedure is repeated until the relative change defined in Eq (4.2.2) between two consecutive iterations is smaller than a threshold, or the maximum number of iterations is reached. To break the inter-dependency between item tree and user tree construction, we first perform plain matrix factorization defined in Eq (4.2.2) to obtain the initial item factors V_0. We should note that one can also start with item tree construction from initial user factors U_0, but this does not change the nature and convergence of this alternative optimization.

The above alternative optimization is by nature greedy, and its computational complexity is potentially high. When examine the optimization steps in Eq (4.2.6), we can easily recognize that the exhaustive search of item features and thresholds can be performed in parallel in each input set of users and items. This greatly improves the efficiency of rule induction. Besides, beam search [202] can be applied in each step of predicate selection to improve the quality of learned rules and factors, but with a cost of increased computation.

One can realize that during the alternative optimization, only the latent factors learned for the leaf nodes are kept for next round of tree construction and finally the recommendation, while the factors associated with the intermediate nodes are discarded. As the procedure of inductive rule learning can be considered as a process of divisive clustering of users and items, the intermediate nodes actually capture important information about homogeneity within the identified user clusters and item clusters. To exploit such information, we introduce the learned latent factors from parent node to child node as follows,

$$u_{L,z} = \widetilde{u}_{L,z} + u_z, \;\; u_{R,z} = \widetilde{u}_{R,z} + u_z, \;\; \text{and} \;\; u_{E,z} = \widetilde{u}_{E,z} + u_z,$$

where $u_{L,z}$, $u_{R,z}$ and $u_{E,z}$ are the latent factors to be plugged into Eq (4.2.6) for the three child nodes under parent node z, and u_z is the factor already learned for the parent node z. Intuitively, $\widetilde{u}_{L,z}$, $\widetilde{u}_{L,z}$ and $\widetilde{u}_{L,z}$ can be considered as residual corrections added to the shared representation from parent nodes. Hence, the rule induction process becomes a recursive procedure of latent factor refinement. Without loss of generality, this recursive refinement can be applied to individual users and items on the leaf nodes of both user tree and item tree as well. If we refer the latent factors on the leaf node for individual users and items as personalized representations of users and items, those on the intermediate nodes could be considered as grouplized representations for the partition of users and items.

4.2.3 Experiment

We performed a set of controlled experiments on two widely used benchmark datasets collected from Amazon[3] and Yelp Dataset Challenge[4] to quantitatively evaluate our FacT model. We follow the same technique discussed in Section 4.1.3 to perform pre-processing on the dataset, and the detailed statistic can be found in Table 4.1. Besides, we compare FacT against the same baselines as that for MTER in both recommendation and explanation quality.

Top-K Recommendation

We first evaluate FacT's recommendation quality. In a good recommender system, items ranked higher in a result list should be more relevant to a user's preference. NDCG assigns higher importance to the items ranked on top. In this experiment, we fix the depth of the user tree and item tree in FacT to 6 and the size of latent dimension to 20. The recommendation performance measured by NDCG@10, 20, 50, 100 of each model is shown in Table 4.2 for Amazon and Yelp datasets, respectively.

Compared with all the baselines, FacT consistently gives better recommendation in both Amazon and Yelp datasets. Among all the baselines, NMF is widely used in practice. But it only uses dyadic user-item ratings for model learning. By introducing the pairwise ranking constraint, BPRMF improves greatly comparing to NMF. However, like reported in previous works [143, 200], BPRMF is also limited to the rating information, and cannot utilize the implicit information included in user reviews. By exploiting the review content for recommendation, JMARS and EFM gave explainable recommendation to users with comparable ranking quality with BPRMF, and MTER showed its potential in providing explanations along with decent recommendation quality. However, they are still limited for different reasons. JMARS maps users, items and features into the same topic space, where the dependency among them is not preserved. Both EMF and MTER model the users and items as individual vectors by matrix or tensor factorization, while FacT clusters users and items into groups (e.g., the intermediate nodes in user and item trees) to take advantage of in-group homogeneity for better latent factor learning. The basic intuition here is that the representations of users and items that share the same preferences or feature qualities should be pushed close to each other. And FacT enforced it by item feature based tree construction. Moreover, the personalized vectors added to the leaf nodes distinguish individual users/items, and provide accurate personalized recommendations. Besides, we observe that FacT achieves the significant improvement at NDCG@10 (9.70% against the best baseline on Amazon and 8.62% on Yelp) and NDCG@20 (8.59% against the best baseline on Amazon and 9.10% on Yelp). This is important for practical recommender system as FacT can provide more accurate recommendations earlier down the ranking list.

Latent Dimensions

The dimension of latent factors determines model capacity, and is an important hyper-parameter for factorization based methods. In this experiment, we explore the influence of latent dimension and the stability of FacT against this hyper-parameter comparing to baseline latent factor models. We varied the dimension of latent factors from 5 to 1000 and compared the results of FacT with FMF, NMF, and BPRMF, which also utilize matrix factorization as their base learning component. The results are summarized in Figure 4.5.

It is clear from the figure that FacT outperformed the other baselines with NDCG@50 under all different settings. We can also observe that for all models, the performance varied significantly when the dimension was lower. And with larger size of latent dimensions, all models' performance degenerated, as they demand more training data to fit the increasing number of parameters. The situation becomes especially worse for FacT, as we are also learning latent factors for intermediate nodes. This follows what we expected: it is generally hard for a model with a smaller dimension of latent factors to capture the affinity between users and items; but model with a higher dimension for latent factors is easier to get over-fitted with insufficient training data. Thus, in the following experiment, we choose 20 as the dimension of latent factor in FacT.

[3] http://jmcauley.ucsd.edu/data/amazon/
[4] https://www.yelp.com/dataset

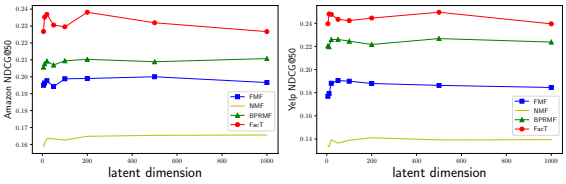


Figure 4.5: NDCG@50 vs. the size of latent dimensions.

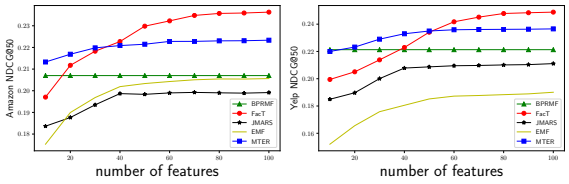


Figure 4.6: NDCG@50 v.s. the number of item features.

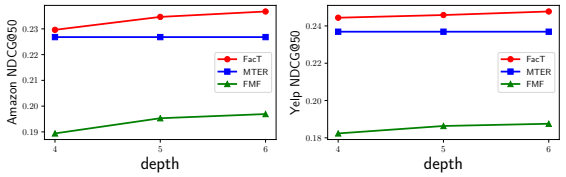


Figure 4.7: Varying the depth of user tree.

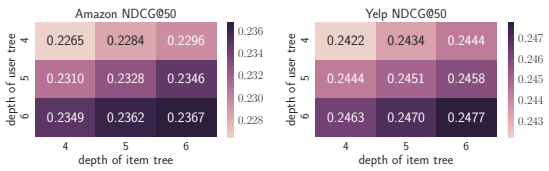


Figure 4.8: Depth of trees v.s. FacT performance.

Number of item features

As shown in Table 4.1, there are 101 item features extracted from Amazon dataset and 104 features from Yelp dataset. Though we have filtered out features with low frequency, limited by the depth of our tree structure, not all of these features will be selected for rule construction. In this analysis, we study the impact of number of features in the dataset on the performance of different models. We first ordered the features in a descending order of frequency, and then trained the models with an increasing number of features. The results are reported in Figure 4.6. From Figure 4.6, it is easy to observe that all the models get significantly improved with an increasing number of features. As the number of features got larger, the performance became stable, as more less frequent features were added. This observation suggests that features with high frequency in reviews contribute more to the feature-based recommendation algorithm learning. Especially in FacT, when the number of item features is limited, it cannot correctly create tree branches to guide latent factor learning. And more item features give FacT a higher degree of freedom to recognize the dependency between users and items.

Maximum tree depth

In FacT, we cluster the users and items along with the tree construction. The maximum tree depth controls the resolution of clusters, e.g., how many intermediate and leaf nodes will be created. We fixed all the other hyper-parameters and only tuned the maximum depth of each tree to verify the effect of it. The results are shown in Figure 4.7. We compared the performance of FacT with FMF and MTER. FMF introduces user tree construction to cluster users for cold-start recommendation. And MTER is the best baseline we had in Table 4.2, but as it does not have a tree structure, its performance remains constant in this experiment. And for FacT, we fixed the depth of item tree to 6 and varied the depth of user tree. We can observe both FMF and FacT got better performance with an increasing tree depth, which increases the granularity of the learned latent representations for users.

A more detailed result of varying the depth of both user tree and item tree is shown in Figure 4.8. From this result, we can clearly find that with a larger tree depth, FacT generated consistently better performance. We also notice that the performance change from varying the depth of item tree is much smaller than that from varying the depth of user tree. A possible explanation is that there are only a small portion of items to be recommended to users. The improved resolution on other items has little contribution to FacT's ranking quality.

Inclusion of factors from parent nodes

As discussed before, during the tree construction, the learned latent factors from parent nodes are introduced to the latent factor learning of child nodes. Thus, information about homogeneity in grouped users and items could be passed along the tree. In this experiment, we quantify the contribution of this design in FacT by

Table 4.3: NDCG@50 for FacT with/without inclusion of factors from parent node. (PF: Parent Factor)

Maximum Depth	Amazon		Yelp	
	w/ PF	w/o PF	w/ PF	w/o PF
4	0.2265	0.1811	0.2422	0.1837
5	0.2328	0.1854	0.2451	0.1906
6	0.2367	0.1892	0.2477	0.1985

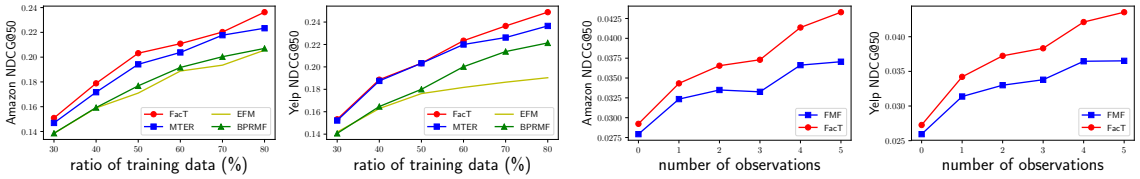


Figure 4.9: NDCG@50 v.s. the amount of training data.

Figure 4.10: NDCG@50 v.s., the # observations in cold-start.

disabling it. From Table 4.3, we can observe that with the personalized term, the model gives significantly better recommendation performance than without it. This directly demonstrates the significance of information sharing among the clustered users and items in FacT.

Dependency on training data

The last thing we investigate is different recommendation algorithms' dependency on the availability of training data. A model requiring less training data is always preferred. We used 30% to 80% of training in each fold during 5-fold cross validation in all the models, and reported the results in Figure 4.9. As expected all models performed better when more training data became available; by exploiting the shared information across users and items assigned to the same tree node, FacT better utilized the available information and stably outperformed all of the baselines.

Cold-start Recommendation

Cold-start is an well-known and challenging problem in recommender systems. Without sufficient information about a new user, it is hard for a recommender system to understand the user's interest and provide recommendations with high quality. A by-product of FacT is that the rules learned in the user tree naturally serve as a set of interview questions to solicit user preferences when a new user comes to the system, i.e., cold-start. For example, based on the user tree in Figure 4.4, the system would get a good understanding of a new user by asking just a few questions following the paths on the tree. In this experiment, we study how FacT performs on the new users. First, we separated the users into two disjoint subsets, containing 95% and 5% users, for training and testing respectively. On the training set, we learned the model and built the user tree and item tree. During testing, for each testing user, we select their first k reviews to construct his/her item feature based user profile (i.e., F_i^u as defined in Eq (4.2.3)). By matching against the user tree, we can easily find the leaf node for each testing user. Then, we use the latent factors reside in the selected leaf node to rank items for this user. We evaluate the performance in the remaining observations from the same user as ground-truth.

We compared FacT with FMF model as it is the only baseline that can handle cold-start. We varied the number of observations for each testing user from 0 to 5, and the results are shown in Figure 4.10. First, it is clear to observe that NDCG got improved with an increasing number of observations used to create the user profile for both FacT and FMF. This indicates the effectiveness of user clustering on the user tree in these two models. Second, thanks to the construction of item tree and BPR constraint, FacT got consistently better performance than FMF. In particular, NDCG@50 for FacT increases faster than FMF with more observations. We attribute this to the fact that FacT uses the item features and user opinions collected from the reviews to perform tree construction, while FMF only uses the item ratings to group users. This indicates the effectiveness of review information in modeling users.

Table 4.4: Result of warm-start user study.

Average Score	Amazon			Yelp		
	EFM	MTER	FacT	EFM	MTER	FacT
Q1	3.64	3.96	**4.45***	3.45	4.06	**4.30***
Q2	3.48	3.88	**4.03**	3.40	3.87	**4.13**
Q3	3.07	3.02	**3.88***	2.98	3.26	**3.94***

* p-value < 0.05

4.2.4 User Study

We performed serious user studies to evaluate user satisfaction of both the recommendations and explanations generated by FacT. We evaluated the performance of FacT on both warm-start users, whose ratings and reviews are known to the system beforehand, and cold-start users who are totally new to the system. The study is based on the review data in Amazon and Yelp datasets used in previous experiments. We recruited participants on Amazon Mechanical Turk to interact with our system and collected their responses. To reduce the variance and noise of the study, we required the participants to come from an English-speaking country, older than 18 years, and have online shopping experience.

Warm-start Users

In the warm-start setting, we assume user's purchase history is known to the recommender system. However, we are not able to trace the participants' purchase history on Mechanical Turk. Instead, we performed a simulation-based study, in which we asked the participants to evaluate our system from the perspective of selected users in our datasets. Specifically, for each participant, we randomly selected a user from our review dataset and presented this user's reviews for the participant to read. The participants were expected to infer this user's preferences from the review content. Then the participant will be asked several questions to evaluate the recommendation and explanation generated by our algorithm.

We designed the survey questions to evaluate different aspects of our recommender algorithm as follows:

Q1: Generally, are you satisfied with our recommendations?

Q2: Do the explanations presented to you really match your preference?

Q3: Do you have any idea about how we make recommendations for you?

We intended to use Q1 to evaluate user satisfaction of recommended items, use Q2 to judge the effectiveness of explanations, and use Q3 to evaluate the transparency of an explainable recommendation algorithm. For each question, the participants are required to choose from five rated answers: 1. Strongly negative; 2. Negative; 3. Neutral; 4. Positive; and 5. Strongly positive. We used EFM and MTER as baselines, since they both can provide textual explanations, and conducted A/B tests to ensure the evaluation is unbiased. Three hundred questionnaires were collected in total and the results are reported in Table 4.4.

From the statistics, FacT apparently outperformed both baselines in all aspects of this user study, which is further confirmed by the paired t-test. Comparing FacT with EFM and MTER on Q1, the improvement in offline validated recommendation quality directly translated into improved user satisfaction. For Q2, the advantage of FacT shows the effectiveness of our predicate selection in explanation rule construction, which captures user's underlying preferences. Moreover, the results on Q3 verified the user-perceived transparency of our tree guided recommendation and rule-based explanation mechanism.

Cold-start Users

Unlike warm-start users, cold-start users have no review history. In order to generate recommendation and explanation for these users, we progressively query user responses through an interview process. Specifically, each node of the user tree in FacT corresponds to an interview question: *"How do you like this [feature]?"*, where *[feature]* was learned to optimize the explanation rule at this node. When the user answers the interview

Table 4.5: Results of cold-start interleaved test.

number of votes	Amazon		Yelp	
	FMF	FacT	FMF	FacT
Q1	44	**63***	40	**64***
Q2	43	**64***	34	**70***
Q3	45	**62**	33	**71***

* p-value < 0.05

question designated at the current node, he/she will be directed to one of its three child nodes according to the answer. As a result, each user follows a possibly different path from the root node to a leaf node during the interview process. A user's associated latent factor is adaptively refined at each intermediate node based on the user's responses. We make recommendations and explanations according to the resulting path. For comparison, FMF is set as a baseline, since it is the only algorithm that can address the cold-start problem with the same interview process as FacT. As FMF uses items instead of features to construct the tree, the interview question there is changed to *"How do you like this [item]?"*

To interview each participant in this user study, we developed a platform to let the participant interact with our system. [5] To increase the sensitivity of comparison between two recommendation algorithms, we conduct interleaved test [203] in this cold-start study. The participant was asked to interact with two models one after the other in a random order, to compare which one is better according to our designed questions. The recommendation is interactive, based on the participants' responses to the interview questions (i.e., traversing in the user tree). There are three questions for them to answer to compare the recommendations and explanations generated by these two algorithms:

Q1: Generally, between system A and B, whose recommendations are you more satisfied with?

Q2: Between system A and B, whose explanations do you think can better help you understand the recommendations?

Q3: Between system A and B, whose explanations can better help you make a more informed decision?

We collected more than 100 valid responses on each dataset and reported the results in Table 4.5. We can find that FacT is preferred than FMF in all questions on both datasets. It suggests that: First, feature-based rule construction is more effective than item-based rule construction, which leads to improved ranking quality in FacT. Second, the feature-based explanations are preferred than the item-based ones, as the former characterizes user preferences at a finer granularity. Last, feature-based explanation rules also provide improved transparency than item-based explanations, which verifies the explainability of our solution. All the evidences from this interleaved user study demonstrate the power of FacT to address the cold-start problem.

4.3 Conclusion

In this chapter, we focus on improving the transparency of the information systems by providing explanations for the recommendations. We introduce a multi-task learning solution via a joint tensor factorization for explainable recommendation, which spearheads a new direction for boosting the performance of recommendations and explanations jointly. Both offline experiments and user study show the comprehensiveness and effectiveness of our model. Then, we present FacT model, which seamlessly integrates latent factor learning with explanation rule learning for explainable recommendation. The fidelity of explanation is guaranteed by modeling the latent factors as a function of explanation rules; and the quality of recommendation is ensured by optimizing both latent factors and rules under a recommendation based metric. Offline experiments and user studies have shown the effectiveness of our model in both aspects.

This is the first step towards exploiting explicit opinionated content for explainable recommendation; and it creates several lines of important future works. For MTER, the dependency among different entities is

[5] https://aobo-y.github.io/explanation-recommendation/

implicitly modeled via the observations in users' rating and review history. It can be further explored and more accurately modeled by incorporating external resources, such as social network structures among users and product taxonomy among items. And our tensor structure provides necessary flexibility to do so. More advanced text synthesis techniques, such as neural language models, can be utilized to generate more complete and natural sentences for explanations. For FacT, instead of using a set of single threshold predicates, we can introduce more complex forms, such as nonlinear function, for better explainability. Last, our model only uses templates to generate explanations, we believe that using features as key words and retrieving sentences from items reviews will definitely generate more natural explanations.

Chapter 5

Conclusion and Future Work

The importance of reliable decision-making in information system can never be overestimated, especially when the decisions made by the systems will directly influence human's social life, e.g., which treatment to take, which products get purchased, which opinion gets heard. In this thesis, we provide a deep and thorough understanding of the reliability of the decision-making in modern information systems, and focus on three importance aspects of the reliability: accuracy, transparency and fairness. By combining the proposed techniques, a modern information system can have a more reliable decision-making when serving both the consumer and information provider. Rigorous theoretical analysis and extensive empirical evaluation validate the proposed models' applicability in various contexts and applications.

In Chapter 2, we introduced a pairwise online learning to rank framework, PairRank, to handle the noisy and biased feedback during the online interactions with the users. We effectively explore the unknowns in the quadratic pairwise ranking space, which significantly reduced the exponentially large search space in ranking problems. We present the details of apply PairRank to linear ranker, and extend PairRank to a neural ranker, e.g., a multi-layer RankNet with the advanced theoretical analysis of neural network, e.g., neural tangent kernel technique. Rigorous theoretical analysis show that PairRank achieves sublinear upper regret bound defined on the number of mis-ordered pairs, which directly links PairRank's convergence with classical ranking evaluations. method directly optimizes the ranking-based metrics with significant improvement compared to existing baselines. To eliminate the computation overhead and make PairRank operational in practice, we developed a provable efficient exploration strategy for neural PairRank based on bootstrapping, named as PePairRank. Confidence interval based PairRank solutions are too expensive to be exactly implemented in practice. PePairRank is proved to induce a sublinear upper regret bound counted over the number of mis-ordered pairs during online result serving, and its added computational overhead is feasible. Our extensive empirical evaluations demonstrate that our perturbation-based exploration unleashes the power of neural rankers in OL2R, with minimally added computational overhead (e.g., oftentimes only one additional ranker is needed to introduce the required exploration). Overall, we show that with PairRank, the model can efficiently and effectively explore the unknowns while providing satisfying service to the users.

In Chapter 3, we focus on providing fair exposure across information providers in online learning to rank. We develop a framework to equipment the online learning to rank solutions with deterministic exploration with fair exposure guarantee. The key idea is to calibrate the explore-exploit trade-off with respect to the fairness control. In particular, the model's exploration is confined within a subset of random permutations, where fairness across groups is maintained while the feedback is still unbiased. Theoretically we prove such a strategy introduces minimum distortion in OL2R's regret to obtain fairness. Extensive empirical analysis is performed on two public learning to rank benchmark datasets to demonstrate the effectiveness of the proposed solution in balancing fairness, exploration and exploitation when compared to existing fair OL2R solutions.

In Chapter 4, we focus on improving the transparency of the existing recommendation systems by providing the explanations. We developed two different multi-task learning solutions for explainable personalization. In MTER, the task of personalization and explanation are modeled by three three-way tensors over user,

item, feature, and opinionated phrases, which are constructed from user-generated review content. Via a joint decomposition, we map user, item, feature and opinionated phrase to a shared latent representation space and provide the explanation by searching in the space. Built upon MTER, our second work, FacT, explains latent factor models with rule-based explanations. It treats the latent factors as a function of the rules. As the latent factors are learnt subject to the explanation rules, the fidelity of explanations is ensured; and because latent factors are optimized for improving personalization, the quality of personalization is also provided.

5.1 Future Work

Research in this dissertation focus on improving the reliability of the decision-making in information systems on three main aspects. There are more challenges to be conquered towards a more reliable or truthful modern systems.

5.1.1 Collaborative Learning for Personalized Online Learning to Rank

The proposed OL2R solution focuses on population-level model estimation, hence fail to recognize the heterogeneity across different users' information needs and result ranking preference. A reliable decision-making in information systems should be able to present the correct information to the right user. Given the value of personalization in modern online systems, it is of great importance to push the frontier of OL2R research to a personalized setting, where the online optimization of ranking functions is performed on a per-user basis.

Collaborative learning [31, 204–206] leverages explicit and implicit user dependency to jointly estimate individual models and is thus a promising framework for us to develop personalized OL2R solutions. It has the advantage to balance data sparsity and heterogeneity in different users' ranking preferences. We can coordinate the pairwise exploration in uncertain pairs and model learning across related users in parallel for efficient joint exploration and personalized learning. One key technical challenge is how to define the relatedness among the individualized OL2R models. Many existing collaborative learning algorithms assume the knowledge of user networks [205–207] (either social relations or similarity information). This assumption is often hard to satisfy, especially in task-specific online systems, such as a library system or a medical record search system. Even when such information is available, there is no guarantee it correlates with the underlying user ranking preferences [208]. What's more, almost all the existing collaborative learning solutions assume the dependency among users is static. This is rarely true in real-world systems, in which users frequently join and leave the system, and users' similarity and relatedness can shift over time [209]. Another unique challenge in collaborative OL2R is the data privacy. Data or model sharing across users is at the risk of revealing private information [210], and the frequent communications further amplify the risk of privacy breaches (e.g., extraction attacks) [211, 212]. We can randomize individual user's data before communicating with others to achieve strong privacy guarantees, such as local differential privacy [213, 214]. The federated learning framework [215–217] can also be leveraged to achieve private model aggregation by sharing randomized parameters for decentralized model estimation.

5.1.2 Robustness of Online Learning to Rank

In this thesis, we assume a benign learning environment: users will fully cooperate and trust the retrieval system. But real information systems operate in adversarial contexts, where users may have strong motivation to manipulate the rankings, and have reasons to fear that sensitive information about their actions will be exposed. The essence of most existing OL2R algorithms is online bandit optimization, either in the form of DBGD [12, 218], or stochastic multi-armed bandit [21–23]. Several effective poisoning attacks have been demonstrated against multi-armed bandits that only require logarithmic efforts to fully control the actions of learned models [219, 220]. Hence, it is necessary to consider the decision-making in an adversarial setting with robustness guarantee.

5.1.3 Fairness vs. Utility

Various solutions for ranking fairness have been developed, existing studies and our proposed FairExp framework believed the trade-off between result fairness and utility is inevitable, and thus the users have to

accept the deteriorated ranking utility. However, we argue that serving as the intermediary that provides the ranking and recommendation service to both users and content providers, the platform should be responsible for the satisfaction of both sides. Any downgrade of the service will lead to the dissatisfaction, and make the users or the providers leave the platform. Hence, a reliable information system should guarantee that neither of these two parties should pay for the requirement on the system by the other party.

The scarce resource that the content providers seek is the exposure of the items, i.e., the expected examination from a population of users. Most of the existing work of fair ranking is under the assumption that the exposure of each position is pre-defined and fixed, e.g., position-based examination models [5, 6, 85]. However, research shows that item position is not the only factor that affects exposure. For example, when searching on an e-commerce platform, items with promotion badges can get more attention from the users, irrespective of its position in the list [221]. This motivates us to alter the exposure of each position by incentives. The information service provided by the platform attracts the content providers and users, which in turn, can be used to generate revenue. Hence, the platform can allocate part of the revenue to the positions of the ranked list as incentives. Such incentives can alter the examination probability and lead to a different distribution of users' attention, compared to the original position-based exposure, to guarantee the fair exposure across content providers. Besides, users' dissatisfaction on the decreased ranking quality can be compensated by the provided bonus, e.g., the incentive for examining the target position.

Bibliography

[1] Kush R Varshney. Trustworthy machine learning and artificial intelligence. *XRDS: Crossroads, The ACM Magazine for Students*, 25(3):26–29, 2019.

[2] Mustafa Bilgic and Raymond J Mooney. Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI*, volume 5, 2005.

[3] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.

[4] Rashmi Sinha and Kirsten Swearingen. The role of transparency in recommender systems. In *CHI'02 extended abstracts on Human factors in computing systems*, pages 830–831. ACM, 2002.

[5] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th ACM SIGIR*, pages 154–161. ACM, 2005.

[6] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th ACM SIGIR*, pages 19–26. ACM, 2006.

[7] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM TOIS*, 25(2):7, 2007.

[8] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. Unbiased learning-to-rank with biased feedback. In *Proceedings of the 10th ACM WSDM*, pages 781–789. ACM, 2017.

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

[10] Huazheng Wang, Sonwoo Kim, Eric McCord-Snook, Qingyun Wu, and Hongning Wang. Variance reduction in gradient exploration for online learning to rank. In *SIGIR 2019*, pages 835–844, 2019.

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

[12] Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *ICML*, pages 1201–1208, 2009.

[13] Anne Schuth, Harrie Oosterhuis, Shimon Whiteson, and Maarten de Rijke. Multileave gradient descent for fast online learning to rank. In *Proceedings of the 9th ACM WSDM*, pages 457–466, 2016.

[14] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *NIPS*, pages 2312–2320, 2011.

[15] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

[16] Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural thompson sampling. *arXiv preprint arXiv:2010.00827*, 2020.

[17] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. Estimating interleaved comparison outcomes from historical click data. In *Proceedings of the 21st CIKM*, pages 1779–1783, 2012.

[18] Tong Zhao and Irwin King. Constructing reliable gradient exploration for online learning to rank. In *Proceedings of the 25th ACM CIKM*, pages 1643–1652, 2016.

[19] Huazheng Wang, Ramsey Langley, Sonwoo Kim, Eric McCord-Snook, and Hongning Wang. Efficient exploration of gradient space for online learning to rank. In *SIGIR 2018*, pages 145–154, 2018.

[20] Sumeet Katariya, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. Dcm bandits: Learning to rank with multiple clicks. In *ICML*, pages 1215–1224, 2016.

[21] Masrour Zoghi, Tomas Tunys, Mohammad Ghavamzadeh, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. Online learning to rank in stochastic click models. In *ICML 2017*, pages 4199–4208, 2017.

[22] Tor Lattimore, Branislav Kveton, Shuai Li, and Csaba Szepesvari. Toprank: A practical algorithm for online stochastic ranking. In *NIPS*, pages 3945–3954, 2018.

[23] Shuai Li, Tor Lattimore, and Csaba Szepesvári. Online learning to rank with features. *arXiv preprint arXiv:1810.02567*, 2018.

[24] Branislav Kveton, Chang Li, Tor Lattimore, Ilya Markov, Maarten de Rijke, Csaba Szepesvari, and Masrour Zoghi. Bubblerank: Safe online learning to rerank. *arXiv preprint arXiv:1806.05819*, 2018.

[25] Harrie Oosterhuis and Maarten de Rijke. Differentiable unbiased online learning to rank. In *Proceedings of the 27th ACM CIKM*, pages 1293–1302, 2018.

[26] Yiling Jia, Huazheng Wang, Stephen Guo, and Hongning Wang. Pairrank: Online pairwise learning to rank by divide-and-conquer. In *Proceedings of the Web Conference 2021*, pages 146–157, 2021.

[27] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[28] Amit Sharma and Dan Cosley. Do social explanations work?: studying and modeling the effects of social explanations in recommender systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1133–1144. International World Wide Web Conferences Steering Committee, 2013.

[29] Behnoush Abdollahi and Olfa Nasraoui. Using explainability for constrained matrix factorization. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 79–83. ACM, 2017.

[30] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92, 2014.

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

[32] Laurent Jacob, Jean-philippe Vert, and Francis R Bach. Clustered multi-task learning: A convex formulation. In *Advances in neural information processing systems*, pages 745–752, 2009.

[33] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. Explainable recommendation via multi-task learning in opinionated text data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 165–174. ACM, 2018.

[34] Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. Automatic construction of a context-aware sentiment lexicon: An optimization approach. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 347–356, New York, NY, USA, 2011. ACM.

[35] William J Clancey. The epistemology of a rule-based expert system: a framework for explanation. *Artificial intelligence*, 20(3):215–251, 1983.

[36] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Information Retrieval*, 16(1):63–90, 2013.

[37] Rolf Jagerman, Harrie Oosterhuis, and Maarten de Rijke. 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*, pages 15–24, 2019.

[38] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM TOIS*, 30(1):6, 2012.

[39] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[40] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM KDD*, pages 133–142, 2002.

[41] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *ICML*, pages 784–791, 2008.

[42] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *ICML*, pages 767–776, 2015.

[43] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Combinatorial cascading bandits. In *NIPS*, pages 1450–1458, 2015.

[44] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. Contextual combinatorial cascading bandits. In *ICML*, volume 16, pages 1245–1253, 2016.

[45] Anne Schuth, Floor Sietsma, Shimon Whiteson, Damien Lefortier, and Maarten de Rijke. Multileaved comparisons for fast online evaluation. In *Proceedings of the 23rd ACM CIKM*, pages 71–80. ACM, 2014.

[46] Harrie Oosterhuis and Maarten de Rijke. Balancing speed and quality in online learning to rank for information retrieval. In *Proceedings of the 26th 2017 ACM CIKM*, pages 277–286, 2017.

[47] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Support vector learning for ordinal regression. 1999.

[48] Shiyu Liang and R Srikant. Why deep neural networks for function approximation? *arXiv preprint arXiv:1610.04161*, 2016.

[49] Matus Telgarsky. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101*, 2015.

[50] Matus Telgarsky. Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485*, 2016.

[51] Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

[52] Dmitry Yarotsky. Optimal approximation of continuous functions by very deep ReLU networks. *arXiv preprint arXiv:1802.03620*, 2018.

[53] Boris Hanin and Mark Sellke. Approximating continuous functions by ReLU nets of minimal width. *arXiv preprint arXiv:1710.11278*, 2017.

[54] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep ReLU networks. *Machine Learning*, 2019.

[55] Difan Zou and Quanquan Gu. An improved analysis of training over-parameterized deep neural networks. In *Advances in Neural Information Processing Systems*, 2019.

[56] Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

[57] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

[58] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 109(3):467–492, 2020.

[59] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[60] Yuan Cao and Quanquan Gu. Generalization error bounds of gradient descent for learning over-parameterized deep relu networks. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[61] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, 2019.

[62] Zixiang Chen, Yuan Cao, Difan Zou, and Quanquan Gu. How much over-parameterization is sufficient to learn deep relu networks? *arXiv preprint arXiv:1911.12360*, 2019.

[63] Amit Daniely. SGD learns the conjugate kernel class of the network. In *Advances in Neural Information Processing Systems*, pages 2422–2430, 2017.

[64] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, 2019.

[65] Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. The lambdaloss framework for ranking metric optimization. In *CIKM '18*, pages 1313–1322. ACM, 2018.

[66] Vitor R Carvalho, Jonathan L Elsas, William W Cohen, and Jaime G Carbonell. Suppressing outliers in pairwise preference ranking. In *Proceedings of the 17th ACM CIKM*, pages 1487–1488, 2008.

[67] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. Click chain model in web search. In *Proceedings of the 18th WWW*, pages 11–20, 2009.

[68] Fan Guo, Chao Liu, and Yi Min Wang. Efficient multiple-click models in web search. In *Proceedings of the 2nd WSDM*, pages 124–131, 2009.

[69] Rama Kumar Pasumarthi, Sebastian Bruch, Xuanhui Wang, Cheng Li, Michael Bendersky, Marc Najork, Jan Pfeifer, Nadav Golbandi, Rohan Anil, and Stephan Wolf. Tf-ranking: Scalable tensorflow library for learning-to-rank. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2970–2978, 2019.

[70] Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, 2004.

[71] Suresh Balakrishnama and Aravind Ganapathiraju. Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, 18(1998):1–8, 1998.

[72] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[73] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pages 535–543, 2015.

[74] Tao Qin and Tie-Yan Liu. Introducing letor 4.0 datasets, 2013.

[75] Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 887–898, 2012.

[76] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2019. In press.

[77] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

[78] Branislav Kveton, Csaba Szepesvari, Sharan Vaswani, Zheng Wen, Tor Lattimore, and Mohammad Ghavamzadeh. Garbage in, reward out: Bootstrapping exploration in multi-armed bandits. In *International Conference on Machine Learning*, pages 3601–3610. PMLR, 2019.

[79] Branislav Kveton, Csaba Szepesvári, Mohammad Ghavamzadeh, and Craig Boutilier. Perturbed-history exploration in stochastic linear bandits. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence (UAI)*, page 176, 2019.

[80] Branislav Kveton, Manzil Zaheer, Csaba Szepesvári, Lihong Li, Mohammad Ghavamzadeh, and Craig Boutilier. Randomized exploration in generalized linear bandits. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 2020.

[81] Yiling Jia, Weitong Zhang, Dongruo Zhou, Quanquan Gu, and Hongning Wang. Learning contextual bandits through perturbed rewards. *arXiv preprint arXiv:2201.09910*, 2022.

[82] Jordan T Ash, Cyril Zhang, Surbhi Goel, Akshay Krishnamurthy, and Sham Kakade. Anti-concentrated confidence bonuses for scalable exploration. *arXiv preprint arXiv:2110.11202*, 2021.

[83] Haque Ishfaq, Qiwen Cui, Viet Nguyen, Alex Ayoub, Zhuoran Yang, Zhaoran Wang, Doina Precup, and Lin F Yang. Randomized exploration for reinforcement learning with general value function approximation. *arXiv preprint arXiv:2106.07841*, 2021.

[84] Yiling Jia and Hongning Wang. Learning neural ranking models online from implicit user feedback. *arXiv preprint arXiv:2201.06658*, 2022.

[85] Zhiwei Guan and Edward Cutrell. An eye tracking study of the effect of target rank on web search. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 417–420, 2007.

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

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

[88] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining*, pages 2221–2231, 2019.

[89] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[90] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[91] Meike Zehlike, Philipp Hacker, and Emil Wiedemann. Matching code and law: achieving algorithmic fairness with optimal transport. *Data Mining and Knowledge Discovery*, 34(1):163–200, 2020.

[92] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2012.

[93] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

[94] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, pages 797–806, 2017.

[95] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

[96] Mattias Rost, Louise Barkhuus, Henriette Cramer, and Barry Brown. Representation and communication: Challenges in interpreting large social media datasets. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 357–362, 2013.

[97] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 50(1):3–44, 2021.

[98] Matt J Kusner, Joshua R Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *arXiv preprint arXiv:1703.06856*, 2017.

[99] Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. A unified approach to quantifying algorithmic unfairness: Measuring individual &group unfairness via inequality indices. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2239–2248, 2018.

[100] Chris Russell, M Kusner, C Loftus, and Ricardo Silva. When worlds collide: integrating different counterfactual assumptions in fairness. In *Advances in neural information processing systems*, volume 30. NIPS Proceedings, 2017.

[101] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. Fairness testing: testing software for discrimination. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, pages 498–510, 2017.

[102] Niki Kilbertus, Manuel Gomez Rodriguez, Bernhard Schölkopf, Krikamol Muandet, and Isabel Valera. Fair decisions despite imperfect predictions. In *International Conference on Artificial Intelligence and Statistics*, pages 277–287. PMLR, 2020.

[103] Silvia Chiappa and William S Isaac. A causal bayesian networks viewpoint on fairness. In *IFIP International Summer School on Privacy and Identity Management*, pages 3–20. Springer, 2018.

[104] Bruce Glymour and Jonathan Herington. Measuring the biases that matter: The ethical and casual foundations for measures of fairness in algorithms. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 269–278, 2019.

[105] Razieh Nabi and Ilya Shpitser. Fair inference on outcomes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[106] L Elisa Celis, Amit Deshpande, Tarun Kathuria, and Nisheeth K Vishnoi. How to be fair and diverse? *arXiv preprint arXiv:1610.07183*, 2016.

[107] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017.

[108] Philip Adler, Casey Falk, Sorelle A Friedler, Tionney Nix, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54(1):95–122, 2018.

[109] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on fairness, accountability and transparency*, pages 119–133. PMLR, 2018.

[110] Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254, 2019.

[111] Vasileios Iosifidis, Besnik Fetahu, and Eirini Ntoutsi. Fae: A fairness-aware ensemble framework. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1375–1380. IEEE, 2019.

[112] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.

[113] Flavio P Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3995–4004, 2017.

[114] James E Johndrow and Kristian Lum. An algorithm for removing sensitive information: application to race-independent recidivism prediction. *The Annals of Applied Statistics*, 13(1):189–220, 2019.

[115] Naman Goel, Mohammad Yaghini, and Boi Faltings. Non-discriminatory machine learning through convex fairness criteria. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[116] Ray Jiang, Aldo Pacchiano, Tom Stepleton, Heinrich Jiang, and Silvia Chiappa. Wasserstein fair classification. In *Uncertainty in Artificial Intelligence*, pages 862–872. PMLR, 2020.

[117] Hoda Heidari, Claudio Ferrari, Krishna P Gummadi, and Andreas Krause. Fairness behind a veil of ignorance: A welfare analysis for automated decision making. *arXiv preprint arXiv:1806.04959*, 2018.

[118] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data mining and knowledge discovery*, 21(2):277–292, 2010.

[119] L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 319–328, 2019.

[120] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2212–2220, 2019.

[121] Nathan Kallus and Angela Zhou. The fairness of risk scores beyond classification: Bipartite ranking and the xauc metric. *Advances in neural information processing systems*, 32:3438–3448, 2019.

[122] Harikrishna Narasimhan, Andrew Cotter, Maya Gupta, and Serena Wang. Pairwise fairness for ranking and regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5248–5255, 2020.

[123] Ke Yang and Julia Stoyanovich. Measuring fairness in ranked outputs. In *Proceedings of the 29th international conference on scientific and statistical database management*, pages 1–6, 2017.

[124] L Elisa Celis, Damian Straszak, and Nisheeth K Vishnoi. Ranking with fairness constraints. *arXiv preprint arXiv:1704.06840*, 2017.

[125] L Elisa Celis, Anay Mehrotra, and Nisheeth K Vishnoi. Interventions for ranking in the presence of implicit bias. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 369–380, 2020.

[126] Abolfazl Asudeh, HV Jagadish, Julia Stoyanovich, and Gautam Das. Designing fair ranking schemes. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1259–1276, 2019.

[127] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1569–1578, 2017.

[128] Carlos Castillo. Fairness and transparency in ranking. In *ACM SIGIR Forum*, volume 52, pages 64–71. ACM New York, NY, USA, 2019.

[129] Meike Zehlike, Ke Yang, and Julia Stoyanovich. Fairness in ranking: A survey. *arXiv preprint arXiv:2103.14000*, 2021.

[130] Meike Zehlike and Carlos Castillo. Reducing disparate exposure in ranking: A learning to rank approach. In *Proceedings of The Web Conference 2020*, pages 2849–2855, 2020.

[131] Ashudeep Singh and Thorsten Joachims. Policy learning for fairness in ranking. *arXiv preprint arXiv:1902.04056*, 2019.

[132] Himank Yadav, Zhengxiao Du, and Thorsten Joachims. Policy-gradient training of fair and unbiased ranking functions. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1044–1053, 2021.

[133] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 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*, pages 429–438, 2020.

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

[135] Avijit Ghosh, Ritam Dutt, and Christo Wilson. When fair ranking meets uncertain inference. *arXiv preprint arXiv:2105.02091*, 2021.

[136] Ashudeep Singh, David Kempe, and Thorsten Joachims. Fairness in ranking under uncertainty. *arXiv preprint arXiv:2107.06720*, 2021.

[137] Stephen E Robertson. The probability ranking principle in ir. *Journal of documentation*, 1977.

[138] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 87–94. ACM, 2008.

[139] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. Estimating position bias without intrusive interventions. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 474–482, 2019.

[140] Zhichong Fang, Aman Agarwal, and Thorsten Joachims. Intervention harvesting for context-dependent examination-bias estimation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 825–834, 2019.

[141] Hongning Wang, ChengXiang Zhai, Anlei Dong, and Yi Chang. Content-aware click modeling. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1365–1376. ACM, 2013.

[142] Dawei Yin, Shike Mei, Bin Cao, Jian-Tao Sun, and Brian D Davison. Exploiting contextual factors for click modeling in sponsored search. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 113–122, 2014.

[143] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 83–92, New York, NY, USA, 2014. ACM.

[144] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

[145] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. Matrix factorization techniques for context aware recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, New York, NY, USA, 2011.

[146] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[147] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD*, KDD '04, 2004.

[148] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Mach. Learn.*, 73(3), December 2008.

[149] Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient l2,1-norm minimization. *CoRR*, abs/1205.2631, 2012.

[150] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning, 2017.

[151] W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications, 2019.

[152] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[153] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[154] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. *CoRR*, abs/1802.07814, 2018.

[155] Charles Taylor. The explanation of behaviour. 1964.

[156] Margaret A Boden. Purposive explanation in psychology. 1972.

[157] Wesley Salmon. Scientific explanation and the causal structure of the world. 1984.

[158] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. Explainable recommendation through attentive multi-view learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3622–3629, 2019.

[159] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. Social collaborative viewpoint regression with explainable recommendations. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 485–494, 2017.

[160] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 765–774, 2019.

[161] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 108–116, 2018.

[162] Xu Chen, Yongfeng Zhang, and Zheng Qin. Dynamic explainable recommendation based on neural attentive models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 53–60, 2019.

[163] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. *Www*, 1:285–295, 2001.

[164] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[165] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[166] Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. Learning to rank features for recommendation over multiple categories. pages 305–314, 2016.

[167] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.

[168] Qiming Diao, Minghui Qiu, Chao Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *ACM SIGKDD*, 2014.

[169] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. Social collaborative viewpoint regression with explainable recommendations. In *Proceedings of the Tenth ACM WSDM*, pages 485–494. ACM, 2017.

[170] Nava Tintarev. Explanations of recommendations. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pages 203–206, New York, NY, USA, 2007. ACM.

[171] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1583–1592. International World Wide Web Conferences Steering Committee, 2018.

[172] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), August 2009.

[173] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Sorec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM CIKM*, CIKM '08, 2008.

[174] Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. Automatic construction of a context-aware sentiment lexicon: An optimization approach. In *Proceedings of the 20th International Conference on World Wide Web*, 2011.

[175] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th ACM SIGIR*, pages 83–92, 2014.

[176] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation:n-dimensional tensor factorization for context-aware collaborative filtering. In *ACM Conference on Recommender Systems, Recsys*, 2010.

[177] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. *CoRR*, abs/1205.2618, 2012.

[178] Wentian Li. Random texts exhibit zipf's-law-like word frequency distribution. *IEEE Transactions on information theory*, 38(6):1842–1845, 1992.

[179] Kamal Amarouche, Houda Benbrahim, and Ismail Kassou. Product opinion mining for competitive intelligence. *Procedia Computer Science*, 73:358–365, 2015.

[180] Soo Min Kim and Eduard Hovy. Identifying opinion holders for question answering in opinion texts. *In Proceedings of AAAI-05 Workshop on Question Answering in Restricted Domains*, 2005.

[181] Ronen Feldman, M Fresko, Jacob Goldenberg, Oded Netzer, and Lyle Ungar. Extracting product comparisons from discussion boards. 8001:469–474, 2007.

[182] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12, 2011.

[183] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. *CoRR*, abs/1602.01585, 2016.

[184] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. *CoRR*, abs/1506.04757, 2015.

[185] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD*, 2006.

[186] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 315–324, New York, NY, USA, 2011. ACM.

[187] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 126–135, New York, NY, USA, 2006. ACM.

[188] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.

[189] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 193–202, New York, NY, USA, 2014. ACM.

[190] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[191] Robert Neches, William R. Swartout, and Johanna D. Moore. Enhanced maintenance and explanation of expert systems through explicit models of their development. *Software Engineering, IEEE Transactions on*, (11):1337–1351, 1985.

[192] Michael R Wick and William B Thompson. Reconstructive expert system explanation. *Artificial Intelligence*, 54(1):33–70, 1992.

[193] William van Melle, Edward H Shortliffe, and Bruce G Buchanan. Emycin: A knowledge engineer's tool for constructing rule-based expert systems. *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*, pages 302–313, 1984.

[194] Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. Tem: Tree-enhanced embedding model for explainable recommendation. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 1543–1552, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[195] Mingxuan Sun, Fuxin Li, Joonseok Lee, Ke Zhou, Guy Lebanon, and Hongyuan Zha. Learning multiple-question decision trees for cold-start recommendation. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 445–454, New York, NY, USA, 2013. ACM.

[196] Shuang-Hong Yang, Bo Long, Alexander J Smola, Hongyuan Zha, and Zhaohui Zheng. Collaborative competitive filtering: learning recommender using context of user choice. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 295–304. ACM, 2011.

[197] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010.

[198] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD Conference*, pages 783–792. ACM, 2010.

[199] Mark EJ Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[200] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. Explainable recommendation via multi-task learning in opinionated text data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 165–174. ACM, 2018.

[201] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 194–202. Elsevier, 1995.

[202] Peng Si Ow and Thomas E Morton. Filtered beam search in scheduling. *The International Journal Of Production Research*, 26(1):35–62, 1988.

[203] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 43–52, New York, NY, USA, 2008. ACM.

[204] Olivier Chapelle, Pannagadatta Shivaswamy, Srinivas Vadrevu, Kilian Weinberger, Ya Zhang, and Belle Tseng. Multi-task learning for boosting with application to web search ranking. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1189–1198. ACM, 2010.

[205] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. Contextual bandits in a collaborative environment. In *The 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pisa, Italy, 2016.

[206] Huazheng Wang, Qingyun Wu, and Hongning Wang. Factorization bandits for interactive recommendation. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*, pages 2695–2702, California, USA, 2017.

[207] Nicolò Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. pages 737–745, 2013.

[208] Julien Delporte, Alexandros Karatzoglou, Tomasz Matuszczyk, and Stéphane Canu. Socially enabled preference learning from implicit feedback data. In *Machine Learning and Knowledge Discovery in Databases*, pages 145–160. Springer, 2013.

[209] Chayant Tantipathananandh, Tanya Berger-Wolf, and David Kempe. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 717–726. ACM, 2007.

[210] Huazheng Wang, Qian Zhao, Qingyun Wu, Shubham Chopra, Abhinav Khaitan, and Hongning Wang. Global and local differential privacy for collaborative bandits. In *Fourteenth ACM Conference on Recommender Systems*, pages 150–159, 2020.

[211] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2512–2520. IEEE, 2019.

[212] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[213] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[214] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. In *Advances in neural information processing systems*, pages 2879–2887, 2014.

[215] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[216] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, and Rachel Cummings. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[217] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[218] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

[219] Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Jerry Zhu. Adversarial attacks on stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 3640–3649, 2018.

[220] Fang Liu and Ness Shroff. Data poisoning attacks on stochastic bandits. In *International Conference on Machine Learning*, pages 4042–4050, 2019.

[221] Fatemeh Sarvi, Maria Heuss, Mohammad Aliannejadi, Sebastian Schelter, and Maarten de Rijke. Understanding and mitigating the effect of outliers in fair ranking. *arXiv preprint arXiv:2112.11251*, 2021.

[222] Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: a bibliography. In *Acm Sigir Forum*, volume 37, pages 18–28. ACM New York, NY, USA, 2003.

[223] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 43–52, 2008.

[224] Simon Caton and Christian Haas. Fairness in machine learning: A survey. *arXiv preprint arXiv:2010.04053*, 2020.

[225] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

[226] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten De Rijke. Reusing historical interaction data for faster online learning to rank for ir. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 183–192, 2013.

[227] Yongfeng Zhang and Xu Chen. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192*, 2018.

[228] Yongkai Wu, Lu Zhang, and Xintao Wu. On discrimination discovery and removal in ranked data using causal graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2536–2544, 2018.

[229] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *Proceedings of the 27th acm international conference on information and knowledge management*, pages 2243–2251, 2018.

[230] Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun, and Ma Shaoping. Fairness-aware group recommendation with pareto-efficiency. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 107–115, 2017.

[231] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

[232] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*, 2017.

[233] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016.

[234] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 560–568, 2008.

[235] Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. ifair: Learning individually fair data representations for algorithmic decision making. In *2019 ieee 35th international conference on data engineering (icde)*, pages 1334–1345. IEEE, 2019.

[236] Flavio du Pin Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Data pre-processing for discrimination prevention: Information-theoretic optimization and analysis. *IEEE Journal of Selected Topics in Signal Processing*, 12(5):1106–1119, 2018.

[237] Lequn Wang and Thorsten Joachims. Fairness and diversity for rankings in two-sided markets. *arXiv preprint arXiv:2010.01470*, 2020.

[238] Tom Sühr, Asia J Biega, Meike Zehlike, Krishna P Gummadi, and Abhijnan Chakraborty. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3082–3092, 2019.

[239] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of The Web Conference 2020*, pages 1194–1204, 2020.

[240] Robin Burke. Multisided fairness for recommendation. *arXiv preprint arXiv:1707.00093*, 2017.

[241] Andrew Stanton, Akhila Ananthram, Congzhe Su, and Liangjie Hong. Revenue, relevance, arbitrage and more: Joint optimization framework for search experiences in two-sided marketplaces. *arXiv preprint arXiv:1905.06452*, 2019.

[242] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Pizzato. Multistakeholder recommendation: Survey and research directions. *User Modeling and User-Adapted Interaction*, 30(1):127–158, 2020.

[243] Kinjal Basu, Cyrus DiCiccio, Heloise Logan, and Noureddine El Karoui. A framework for fairness in two-sided marketplaces. *arXiv preprint arXiv:2006.12756*, 2020.

[244] L Elisa Celis, Sayash Kapoor, Farnood Salehi, and Nisheeth Vishnoi. Controlling polarization in personalization: An algorithmic framework. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 160–169, 2019.

[245] Paul Lagrée, Claire Vernade, and Olivier Cappe. Multiple-play bandits in the position-based model. In *Advances in Neural Information Processing Systems*, pages 1597–1605, 2016.

[246] Aadirupa Saha and Aditya Gopalan. Regret minimization in stochastic contextual dueling bandits. *arXiv preprint arXiv:2002.08583*, 2020.

[247] Harrie Oosterhuis, Anne Schuth, and Maarten de Rijke. Probabilistic multileave gradient descent. In *European Conference on Information Retrieval*, pages 661–668. Springer, 2016.

[248] Abraham D Flaxman, Adam Tauman Kalai, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

[249] Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pages 28–40. Citeseer, 2010.

[250] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[251] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. *arXiv preprint arXiv:1911.04462*, 2019.

[252] Jerri L Ledford. *Search engine optimization bible*, volume 584. John Wiley & Sons, 2015.

[253] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *NIPS*, pages 586–594, 2010.

[254] Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. *arXiv preprint arXiv:1703.00048*, 2017.

[255] Tie-Yan Liu. Learning to rank for information retrieval. 2011.

[256] Michal Valko, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*, 2013.

[257] Wei Hu, Zhiyuan Li, and Dingli Yu. Understanding generalization of deep neural networks trained with noisy labels. *arXiv preprint arXiv:1905.11368*, 2019.

[258] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *arXiv preprint arXiv:1905.13210*, 2019.

[259] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. 2008.

[260] Lin F Yang and Mengdi Wang. Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. *arXiv preprint arXiv:1905.10389*, 2019.

[261] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[262] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

[263] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[264] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[265] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[266] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[267] Tom Zahavy and Shie Mannor. Deep neural linear bandits: Overcoming catastrophic forgetting through likelihood matching. *arXiv preprint arXiv:1901.08612*, 2019.

[268] Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through Bayesian deep Q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9. IEEE, 2018.

[269] Naoki Abe, Alan W Biermann, and Philip M Long. Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica*, 37(4):263–293, 2003.

[270] Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

[271] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[272] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[273] Ziwei Ji and Matus Telgarsky. Risk and parameter convergence of logistic regression. *arXiv preprint arXiv:1803.07300*, 2018.

[274] Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pages 2–47, 2018.

[275] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. *arXiv preprint arXiv:1802.08246*, 2018.

[276] Mor Shpigel Nacson, Nathan Srebro, and Daniel Soudry. Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate. *arXiv preprint arXiv:1806.01796*, 2018.

[277] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[278] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Implicit bias of gradient descent on linear convolutional networks. *arXiv preprint arXiv:1806.00468*, 2018.

[279] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[280] Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, pages 8157–8166, 2018.

[281] Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *Advances in neural information processing systems*, pages 1647–1655, 2011.

[282] Boris Teodorovich Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963.

[283] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013.

[284] Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. SGD learns over-parameterized networks that provably generalize on linearly separable data. *arXiv preprint arXiv:1710.10174*, 2017.

[285] Daniel Soudry, Elad Hoffer, and Nathan Srebro. The implicit bias of gradient descent on separable data. *arXiv preprint arXiv:1710.10345*, 2017.

[286] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018.

[287] David Slepian. The one-sided barrier problem for gaussian noise. *Bell Labs Technical Journal*, 41(2):463–501, 1962.

[288] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[289] Peter Bartlett, Dave Helmbold, and Phil Long. Gradient descent with identity initialization efficiently learns positive definite linear transformations. In *International Conference on Machine Learning*, pages 520–529, 2018.

[290] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018.

[291] Simon S Du, Jason D Lee, and Yuandong Tian. When is a convolutional filter easy to learn? *arXiv preprint arXiv:1709.06129*, 2017.

[292] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with ReLU activation. *arXiv preprint arXiv:1705.09886*, 2017.

[293] Mahdi Soltanolkotabi. Learning relus via gradient descent. *arXiv preprint arXiv:1705.04591*, 2017.

[294] Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. *arXiv preprint arXiv:1706.03175*, 2017.

[295] Yuandong Tian. An analytical formula of population gradient for two-layered ReLU network and its applications in convergence and critical point analysis. *arXiv preprint arXiv:1703.00560*, 2017.

[296] Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. *arXiv preprint arXiv:1702.07966*, 2017.

[297] Daniel Hsu, Sham Kakade, Tong Zhang, et al. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17, 2012.

[298] William Dally. High-performance hardware for machine learning. *NIPS Tutorial*, 2015.

[299] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[300] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[301] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[302] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[303] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[304] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard M Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In *ACL (1)*, pages 1370–1380, 2014.

[305] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[306] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[307] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[308] Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances In Neural Information Processing Systems*, pages 2253–2261, 2016.

[309] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In *Advances in neural information processing systems*, pages 6231–6239, 2017.

[310] Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, pages 698–728, 2016.

[311] Nadav Cohen and Amnon Shashua. Convolutional rectifier networks as generalized tensor decompositions. In *International Conference on Machine Learning*, pages 955–963, 2016.

[312] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. *arXiv preprint arXiv:1606.05336*, 2016.

[313] Ben Poole, Subhaneil Lahiri, Maithreyi Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances In Neural Information Processing Systems*, pages 3360–3368, 2016.

[314] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932, 2014.

[315] Jiří Šíma. Training a single sigmoidal neuron is hard. *Training*, 14(11), 2006.

[316] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pages 855–863, 2014.

[317] Ohad Shamir. Distribution-specific hardness of learning neural networks. *arXiv preprint arXiv:1609.01037*, 2016.

[318] Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In *International Conference on Machine Learning*, pages 3067–3075, 2017.

[319] Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Weight sharing is crucial to succesful optimization. *arXiv preprint arXiv:1706.00687*, 2017.

[320] Avrim Blum and Ronald L Rivest. Training a 3-node neural network is np-complete. In *Advances in neural information processing systems*, pages 494–501, 1989.

[321] Surbhi Goel, Varun Kanade, Adam Klivans, and Justin Thaler. Reliably learning the ReLU in polynomial time. *arXiv preprint arXiv:1611.10258*, 2016.

[322] Yuchen Zhang, Jason D Lee, Martin J Wainwright, and Michael I Jordan. Learning halfspaces and neural networks with random initialization. *arXiv preprint arXiv:1511.07948*, 2015.

[323] Hanie Sedghi and Anima Anandkumar. Provable methods for training neural networks with sparse connectivity. *arXiv preprint arXiv:1412.2693*, 2014.

[324] Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 2015.

[325] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842, 2015.

[326] Chi Jin, Sham M Kakade, and Praneeth Netrapalli. Provable efficient online matrix completion via non-convex stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 4520–4528, 2016.

[327] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. *arXiv preprint arXiv:1703.00887*, 2017.

[328] Kenji Kawaguchi. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*, pages 586–594, 2016.

[329] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Global optimality conditions for deep neural networks. *arXiv preprint arXiv:1707.02444*, 2017.

[330] C Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. *arXiv preprint arXiv:1611.01540*, 2016.

[331] Amirhossein Taghvaei, Jin W Kim, and Prashant G Mehta. How regularization affects the critical points in linear networks. *arXiv preprint arXiv:1709.09625*, 2017.

[332] Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. *arXiv preprint arXiv:1704.08045*, 2017.

[333] Soheil Feizi, Hamid Javadi, Jesse Zhang, and David Tse. Porcupine neural networks:(almost) all local optima are global. *arXiv preprint arXiv:1710.02196*, 2017.

[334] Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *arXiv preprint arXiv:1707.04926*, 2017.

[335] Yair Carmon, Oliver Hinder, John C Duchi, and Aaron Sidford. " convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. *arXiv preprint arXiv:1705.02766*, 2017.

[336] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204, 2015.

[337] Itay Safran and Ohad Shamir. On the quality of the initial basin in overspecified neural networks. In *International Conference on Machine Learning*, pages 774–782, 2016.

[338] Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

[339] Daniel Soudry and Yair Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361*, 2016.

[340] Sanjeev Arora, Rong Ge, Tengyu Ma, and Andrej Risteski. Provable learning of noisy-or networks. *arXiv preprint arXiv:1612.08795*, 2016.

[341] Bo Xie, Yingyu Liang, and Le Song. Diverse neural network learns true target functions. In *Artificial Intelligence and Statistics*, pages 1216–1224, 2017.

[342] Yuchen Zhang, Jason Lee, Martin Wainwright, and Michael Jordan. On the learnability of fully-connected neural networks. In *Artificial Intelligence and Statistics*, pages 83–91, 2017.

[343] Prateek Jain and Praneeth Netrapalli. Fast exact matrix completion with finite samples. In *Conference on Learning Theory*, pages 1007–1034, 2015.

[344] Xingyuan Pan and Vivek Srikumar. Expressiveness of rectifier networks. In *International Conference on Machine Learning*, pages 2427–2435, 2016.

[345] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[346] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[347] Song Mei, Yu Bai, and Andrea Montanari. The landscape of empirical risk for non-convex losses. *arXiv preprint arXiv:1607.06534*, 2016.

[348] Simon S Du, Jason D Lee, Yuandong Tian, Barnabas Poczos, and Aarti Singh. Gradient descent learns one-hidden-layer cnn: Don't be afraid of spurious local minima. *arXiv preprint arXiv:1712.00779*, 2017.

[349] Peter Auer, Mark Herbster, and Manfred K Warmuth. Exponentially many local minima for single neurons. In *Advances in neural information processing systems*, pages 316–322, 1996.

[350] Rong Ge, Jason D Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. *arXiv preprint arXiv:1711.00501*, 2017.

[351] Adam Tauman Kalai and Ravi Sastry. The isotron algorithm: High-dimensional isotonic regression. In *COLT*. Citeseer, 2009.

[352] Itay Safran and Ohad Shamir. Spurious local minima are common in two-layer ReLU neural networks. *arXiv preprint arXiv:1712.08968*, 2017.

[353] Haoyu Fu, Yuejie Chi, and Yingbin Liang. Local geometry of one-hidden-layer neural networks for logistic regression. *arXiv preprint arXiv:1802.06463*, 2018.

[354] Sham M Kakade, Varun Kanade, Ohad Shamir, and Adam Kalai. Efficient learning of generalized linear and single index models with isotonic regression. In *Advances in Neural Information Processing Systems*, pages 927–935, 2011.

[355] Simon S Du and Jason D Lee. On the power of over-parametrization in neural networks with quadratic activation. *arXiv preprint arXiv:1803.01206*, 2018.

[356] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

[357] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[358] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[359] Kai Zhong, Zhao Song, and Inderjit S Dhillon. Learning non-overlapping convolutional neural networks with multiple kernels. *arXiv preprint arXiv:1711.03440*, 2017.

[360] Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. *arXiv preprint arXiv:1802.06509*, 2018.

[361] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252, 2019.

[362] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685, 2019.

[363] John Langford and Rich Caruana. (not) bounding the true error. In *Advances in Neural Information Processing Systems*, pages 809–816, 2002.

[364] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

[365] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

[366] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

[367] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

[368] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[369] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*, 2017.

[370] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. *arXiv preprint arXiv:1802.05296*, 2018.

[371] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.

[372] Boris Hanin. Universal function approximation by deep neural nets with bounded width and ReLU activations. *arXiv preprint arXiv:1708.02691*, 2017.

[373] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. On the convergence rate of training recurrent neural networks. *arXiv preprint arXiv:1810.12065*, 2018.

[374] Xiao Zhang, Yaodong Yu, Lingxiao Wang, and Quanquan Gu. Learning one-hidden-layer ReLU networks via gradient descent. *arXiv preprint arXiv:1806.07808*, 2018.

[375] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *arXiv preprint arXiv:1811.04918*, 2018.

[376] Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 2019.

[377] Hongzhou Lin and Stefanie Jegelka. Resnet with one-neuron hidden layers is a universal approximator. In *Advances in Neural Information Processing Systems*, pages 6172–6181, 2018.

[378] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[379] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[380] Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, Nitin Motgi, Seung-Taek Park, Raghu Ramakrishnan, Scott Roy, and Joe Zachariah. Online models for content optimization. In *Advances in Neural Information Processing Systems*, pages 17–24, 2009.

[381] Lihong Li, Wei Chu, John Langford, Taesup Moon, and Xuanhui Wang. An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. In *Proceedings of the Workshop on On-line Trading of Exploration and Exploitation 2*, pages 19–36, 2012.

[382] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 1015–1022. Omnipress, 2010.

[383] Andreas Krause and Cheng S Ong. Contextual Gaussian process bandit optimization. In *Advances in neural information processing systems*, pages 2447–2455, 2011.

[384] Greg Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019.

[385] Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

[386] Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret. *arXiv preprint arXiv:1903.05594*, 2019.

[387] Robin Allesiardo, Raphaël Féraud, and Djallel Bouneffouf. A neural networks committee for the contextual bandit problem. In *International Conference on Neural Information Processing*, pages 374–381. Springer, 2014.

[388] Carlos Riquelme, George Tucker, and Jasper Snoek. Deep Bayesian bandits showdown. In *International Conference on Learning Representations*, 2018.

[389] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

[390] David S Watkins. *Fundamentals of matrix computations*, volume 64. John Wiley & Sons, 2004.

[391] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 12(May):1655–1695, 2011.

[392] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690. ACM, 2008.

[393] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135, 2013.

[394] Benedict C May, Nathan Korda, Anthony Lee, and David S Leslie. Optimistic bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research*, 13(Jun):2069–2106, 2012.

[395] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, pages 199–213. Springer, 2012.

[396] Nathaniel Korda, Emilie Kaufmann, and Remi Munos. Thompson sampling for 1-dimensional exponential family bandits. In *Advances in neural information processing systems*, pages 1448–1456, 2013.

[397] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

[398] Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471, 2016.

[399] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

[400] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[401] Kwang-Sung Jun, Aniruddha Bhargava, Robert D. Nowak, and Rebecca Willett. Scalable generalized linear bandits: Online computation and hashing. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 99–109, 2017.

[402] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[403] John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 1096–1103, 2008.

[404] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert E. Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1638–1646, 2014.

[405] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[406] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332, 2019.

[407] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. 2019.

[408] Zachary Lipton, Xiujun Li, Jianfeng Gao, Lihong Li, Faisal Ahmed, and Li Deng. BBQ-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[409] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.

[410] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[411] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[412] Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Raghu Ramakrishnan. Content recommendation on web portals. *Communications of the ACM*, 56(6):92–101, 2013.

[413] Ron Kohavi. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pages 202–207, 1996.

[414] Jock A Blackard and Denis J Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24(3):131–151, 1999.

[415] Jeffrey Curtis Schlimmer. Concept acquisition through representational adjustment. 1987.

[416] Arthur Asuncion and David Newman. UCI machine learning repository, 2007.

[417] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96, 2018.

[418] Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 19–26, 2011.

[419] Bradley Efron. *The jackknife, the bootstrap, and other resampling plans*, volume 38. Siam, 1982.

[420] Zeyuan Allen-Zhu and Yuanzhi Li. What can ResNet learn efficiently, going beyond kernels? In *Advances in Neural Information Processing Systems*, 2019.

[421] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[422] Alina Beygelzimer and John Langford. The offset tree for learning with partial labels. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 129–138, 2009.

[423] Kwang-Sung Jun, Aniruddha Bhargava, Robert Nowak, and Rebecca Willett. Scalable generalized linear bandits: Online computation and hashing. In *Advances in Neural Information Processing Systems*, pages 99–109, 2017.

[424] Lijun Zhang, Tianbao Yang, Rong Jin, Yichi Xiao, and Zhi-Hua Zhou. Online stochastic linear optimization under one-bit feedback. In *International Conference on Machine Learning*, pages 392–401, 2016.

[425] Dylan J Foster and Alexander Rakhlin. Beyond ucb: Optimal and efficient contextual bandits with regression oracles. *arXiv preprint arXiv:2002.04926*, 2020.

[426] Sham M Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Efficient bandit algorithms for online multiclass prediction. In *Proceedings of the 25th international conference on Machine learning*, pages 440–447, 2008.

[427] Raphaël Féraud, Robin Allesiardo, Tanguy Urvoy, and Fabrice Clérot. Random forest for the contextual bandit problem. In *Artificial Intelligence and Statistics*, pages 93–101, 2016.

[428] C Quoc and Viet Le. Learning to rank with nonsmooth cost functions. *Proceedings of the Advances in Neural Information Processing Systems*, 19:193–200, 2007.

[429] Jaime Teevan, Susan T Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456. ACM, 2005.

[430] Jaime Teevan, Susan T Dumais, and Eric Horvitz. Potential for personalization. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 17(1):4, 2010.

[431] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[432] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[433] Fang Liu, Clement Yu, and Weiyi Meng. Personalized web search by mapping user queries to categories. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 558–565. ACM, 2002.

[434] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.

[435] Magdalini Eirinaki and Michalis Vazirgiannis. Web mining for web personalization. *ACM Transactions on Internet Technology (TOIT)*, 3(1):1–27, 2003.

[436] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280. ACM, 2007.

[437] Yoon Ho Cho, Jae Kyeong Kim, and Soung Hie Kim. A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications*, 23(3):329–342, 2002.

[438] Fang Liu, Clement Yu, and Weiyi Meng. Personalized web search for improving retrieval effectiveness. *Knowledge and Data Engineering, IEEE transactions on*, 16(1):28–40, 2004.

[439] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[440] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50. ACM, 2005.

[441] David Carmel, Naama Zwerdling, Ido Guy, Shila Ofek-Koifman, Nadav Har'El, Inbal Ronen, Erel Uziel, Sivan Yogev, and Sergey Chernov. Personalized social search based on the user's social network. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1227–1236. ACM, 2009.

[442] Ching-Yung Lin, Nan Cao, Shi Xia Liu, Spiros Papadimitriou, Jimeng Sun, and Xifeng Yan. Smallblue: Social network analysis for expertise search and collective intelligence. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pages 1483–1486. IEEE, 2009.

[443] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[444] Jaime Teevan, Daniel J Liebling, and Gayathri Ravichandran Geetha. Understanding and predicting personal navigation. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 85–94. ACM, 2011.

[445] Bin Tan, Xuehua Shen, and ChengXiang Zhai. Mining long-term search history to improve search accuracy. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 718–723. ACM, 2006.

[446] Kirsten Swearingen and Rashmi Sinha. Beyond algorithms: An hci perspective on recommender systems. In *ACM SIGIR 2001 Workshop on Recommender Systems*, volume 13, pages 1–11. Citeseer, 2001.

[447] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Providing justifications in recommender systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(6):1262–1272, 2008.

[448] Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 47–56. ACM, 2009.

[449] William Swartout, Cecile Paris, and Johanna Moore. Explanations in knowledge systems: Design for explainable expert systems. *IEEE Expert*, 6(3):58–64, 1991.

[450] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international conference on World Wide Web*, pages 581–590. ACM, 2007.

[451] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 259–266. ACM, 2008.

[452] Naveen Farag Awad and MS Krishnan. The personalization privacy paradox: an empirical evaluation of information transparency and the willingness to be profiled online for personalization. *MIS quarterly*, pages 13–28, 2006.

[453] Hong Sheng, Fiona Fui-Hoon Nah, and Keng Siau. An experimental study on ubiquitous commerce adoption: Impact of personalization and privacy concerns. *Journal of the Association for Information Systems*, 9(6):15, 2008.

[454] Donald A Waterman. How do expert systems differ from conventional programs?*. *Expert Systems*, 3(1):16–19, 1986.

[455] Eric J Horvitz, John S Breese, and Max Henrion. Decision theory in expert systems and artificial intelligence. *International journal of approximate reasoning*, 2(3):247–302, 1988.

[456] Hilary Johnson and Peter Johnson. Explanation facilities and interactive systems. In *Proceedings of the 1st international conference on Intelligent user interfaces*, pages 159–166. ACM, 1993.

[457] Thomas W Lauer, Eileen Peacock, and Arthur C Graesser. *Questions and information systems*. Psychology Press, 2013.

[458] Arthur C Graesser. *The psychology of questions*. Lawrence Erlbaum Assoc Inc, 1985.

[459] Kar Yan Tam and Shuk Ying Ho. Understanding the impact of web personalization on user information processing and decision outcomes. *Mis Quarterly*, pages 865–890, 2006.

[460] Thomas Griffiths and Zoubin Ghahramani. Infinite latent feature models and the indian buffet process. 2005.

[461] Xavier Amatriain. Big & personal: data and models behind netflix recommendations. In *Proceedings of the 2nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, pages 1–6. ACM, 2013.

[462] Xavier Amatriain. Building industrial-scale real-world recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 7–8. ACM, 2012.

[463] Stephen Lawrence. Personalization of web search, September 30 2003. US Patent App. 10/676,711.

[464] Amit Sheth, David Avant, and Clemens Bertram. System and method for creating a semantic web and its applications in browsing, searching, profiling, personalization and advertising, October 30 2001. US Patent 6,311,194.

[465] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[466] Udi Manber, Ash Patel, and John Robison. Yahoo! *Communications of the ACM*, 43(8):35, 2000.

[467] Devin F Hosea, Richard Scott Zimmerman, Arthur P Rascon, Anthony Scott Oddo, and Nathaniel John Thurston. Method and system for web page personalization, September 26 2002. US Patent 20,020,138,331.

[468] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, pages 801–810. IEEE, 2007.

[469] Ramnath K Chellappa and Raymond G Sin. Personalization versus privacy: An empirical examination of the online consumer¡¯s dilemma. *Information Technology and Management*, 6(2-3):181–202, 2005.

[470] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[471] Zoubin Ghahramani and Matthew J Beal. Variational inference for bayesian mixtures of factor analysers. In *NIPS*, pages 449–455, 1999.

[472] T.S. Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[473] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[474] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[475] Regina Barzilay and Michael Elhadad. Using lexical chains for text summarization. *Advances in automatic text summarization*, pages 111–121, 1999.

[476] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.(JAIR)*, 22(1):457–479, 2004.

[477] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 322–331. IEEE, 1995.

[478] Joannes Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In *Machine Learning: ECML 2005*, pages 437–448. Springer, 2005.

[479] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306. ACM, 2011.

[480] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.

[481] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 208–211. ACM, 2008.

[482] Susan E Embretson and Steven P Reise. *Item response theory*. Psychology Press, 2013.

[483] Chris Jay Hoofnagle, Bart van der Sloot, and Frederik Zuiderveen Borgesius. The european union general data protection regulation: what it is and what it means. *Information & Communications Technology Law*, 28(1):65–98, 2019.

[484] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*, volume 2. MIT press Cambridge, 1998.

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

[486] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 130–137. ACM, 2017.

[487] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[488] Fred B Bryant and Paul R Yarnold. Principal-components analysis and exploratory and confirmatory factor analysis. 1995.

[489] Harry H Harman. Modern factor analysis. 1960.

[490] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[491] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017.

[492] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017.

[493] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.

[494] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press, 2009.

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

[496] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969, 2003.

[497] Zhaohui Zheng, Keke Chen, Gordon Sun, and Hongyuan Zha. A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 287–294. ACM, 2007.

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

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

[500] John Guiver and Edward Snelson. Bayesian inference for plackett-luce ranking models. In *proceedings of the 26th annual international conference on machine learning*, pages 377–384. ACM, 2009.

[501] Michalis K Titsias and Miguel Lázaro-Gredilla. Spike and slab variational inference for multi-task and multiple kernel learning. In *Advances in neural information processing systems*, pages 2339–2347, 2011.

[502] Chong Wang and David M Blei. Decoupling sparsity and smoothness in the discrete hierarchical dirichlet process. In *Advances in neural information processing systems*, pages 1982–1989, 2009.

[503] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[504] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[505] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

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

[507] Georges E Dupret and Benjamin Piwowarski. 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*, pages 331–338. ACM, 2008.

[508] Ruth Fong and Andrea Vedaldi. Explanations for attributing deep neural network predictions. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 149–167. Springer, 2019.

[509] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

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

[511] Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-order nonconvex stochastic optimization: Handling constraints, high-dimensionality, and saddle-points. pages 651–676, 2019.

[512] Sijia Liu, Jie Chen, Pin-Yu Chen, and Alfred Hero. Zeroth-order online alternating direction method of multipliers: Convergence analysis and applications, 09–11 Apr 2018.

[513] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, pages 315–323, USA, 2013. Curran Associates Inc.

[514] Yining Wang, Simon Du, Sivaraman Balakrishnan, and Aarti Singh. Stochastic zeroth-order optimization in high dimensions. In *International Conference on Artificial Intelligence and Statistics*, pages 1356–1365, 2018.

[515] Kushal Dave, Steve Lawrence, and David M Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.

[516] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[517] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

[518] Christine A Reid, Stephanie A Kolakowsky-Hayner, Allen N Lewis, and Amy J Armstrong. Modern psychometric methodology: Applications of item response theory. *Rehabilitation Counseling Bulletin*, 50(3):177–188, 2007.

[519] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

[520] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 565–573. ACM, 2018.

[521] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5):93:1–93:42, August 2018.

[522] Vicente Iván Sánchez Carmona, Tim Rocktäschel, Sebastian Riedel, and Sameer Singh. Towards extracting faithful and descriptive representations of latent variable models. In *AAAI Spring Symposia*, 2015.

[523] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets, 2014.

[524] Qingyao Ai, Yongfeng Zhang, Keping Bi, and W. Bruce Croft. Explainable product search with a dynamic relation embedding model, 2019.

[525] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 765–774, New York, NY, USA, 2019. ACM.

[526] Felipe Costa, Sixun Ouyang, Peter Dolog, and Aonghus Lawlor. Automatic generation of natural language explanations. *CoRR*, abs/1707.01561, 2017.

[527] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11:1803–1831, August 2010.

[528] Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *CoRR*, abs/1711.09404, 2017.

[529] Chun-Chen Tu, Pai-Shun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. *CoRR*, abs/1805.11770, 2018.

[530] Yong-Deok Kim and Seungjin Choi. Nonnegative tucker decomposition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[531] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.

[532] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[533] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[534] Radu Jurca, Florent Garcin, Arjun Talwar, and Boi Faltings. Reporting incentives and biases in online review forums. *ACM Transactions on the Web (TWEB)*, 4(2):5, 2010.

[535] Arjun Mukherjee, Bing Liu, and Natalie Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st international conference on World Wide Web*, pages 191–200. ACM, 2012.

[536] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 11(9):137, 2018.

[537] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362. ACM, 2016.

[538] Tommaso Di Noia, Vito Claudio Ostuni, Paolo Tomeo, and Eugenio Di Sciascio. Sprank: Semantic path-based ranking for top-n recommendations using linked open data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):9, 2016.

[539] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[540] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[541] Piji Li, Zihao Wang, Lidong Bing, and Wai Lam. Persona-aware tips generation? In *The World Wide Web Conference*, pages 1006–1016. ACM, 2019.

[542] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. pages 994–1003, 2016.

[543] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

[544] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 729–732. ACM, 2016.

[545] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[546] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[547] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[548] A Mnih and YW Teh. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, volume 2, 2012.

[549] Nicolas Fiorini and Zhiyong Lu. Personalized neural language models for real-world query auto completion. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 208–215, 2018.

[550] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[551] Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pages 72–83. Springer, 2006.

[552] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[553] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[554] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[555] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: debiasing learning and evaluation. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 1670–1679. JMLR. org, 2016.

[556] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. Unbiased learning to rank with unbiased propensity estimation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 385–394. ACM, 2018.

[557] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[558] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[559] Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1097–1104. Omnipress, 2011.

[560] Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661, 2016.

[561] Lin Gong and Hongning Wang. When sentiment analysis meets social network: A holistic user behavior modeling in opinionated data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1455–1464. ACM, 2018.

[562] Lin Gong, Lin Lu, Mike Song, and Hongning Wang. Jnet: Learning user representations via joint network embedding and topic embedding. In *The 13th ACM International Web Search and Data Mining Conference*. ACM, 2020.

[563] Qingyun Wu, Naveen Iyer, and Hongning Wang. Learning contextual bandits in a non-stationary environment. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, Ann Arbor Michigan, U.S.A., 2018.

[564] Lin Gong, Mohammad Al Boni, and Hongning Wang. Modeling social norms evolution for personalized sentiment classification. In *The 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016.

[565] Wasi Ahmad Uddin, Kai-Wei Chang, and Hongning Wang. Multi-task learning for document ranking and query suggestion. In *Sixth International Conference on Learning Representations (ICLR 2018)*, Vancouver, Canada, 2018.

[566] Mohammad Al Boni, Keira Qi Zhou, Hongning Wang, and Matthew S Gerber. Model adaptation for personalized opinion analysis. *Volume 2: Short Papers*, page 769, 2015.

[567] Lin Gong, Benjamin Haines, and Hongning Wang. Clustered model adaption for personalized sentiment analysis. In *Proceedings of the 26th International Conference on World Wide Web*, pages 937–946. International World Wide Web Conferences Steering Committee, 2017.

[568] Qingyun Wu, Hongning Wang, Liangjie Hong, and Yue Shi. Returning is believing: Optimizing long-term user engagement in recommender systems. In *25th ACM International Conference on Information and Knowledge Management (CIKM 2017)*, Singapore, 2017.

[569] Huazheng Wang, Qingyun Wu, and Hongning Wang. Learning hidden features for contextual bandits. In *25th ACM International Conference on Information and Knowledge Management (CIKM 2016)*, Indianapolis, United States, 2016.

[570] Wasi Ahmad and Hongning Wang. Topic model based privacy protection in personalized web search. In *The 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1025–1028, Pisa, Italy, 2016.

[571] Hongning Wang, Chi Wang, ChengXiang Zhai, and Jiawei Han. Learning online discussion structures by conditional random fields. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 435–444. ACM, 2011.

[572] Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ahmed Hassan, and Ryen W White. Modeling action-level satisfaction for search task satisfaction prediction. In *Proceedings of the 37th*

international ACM SIGIR conference on Research & development in information retrieval, pages 123–132. ACM, 2014.

[573] Hongning Wang, Minlie Huang, and Xiaoyan Zhu. A generative probabilistic model for multi-label classification. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 628–637. IEEE, 2008.

[574] Hongning Wang, Yue Lu, and ChengXiang Zhai. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD Conference*, pages 618–626. ACM, 2011.

[575] Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ryen W White, and Wei Chu. Learning to extract cross-session search tasks. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1353–1364. ACM, 2013.

[576] Hongning Wang, Anlei Dong, Lihong Li, Yi Chang, and Evgeniy Gabrilovich. Joint relevance and freshness learning from clickthroughs for news search. In *Proceedings of the 21st International Conference on World Wide Web*, pages 579–588. ACM, 2012.

[577] Yue Lu, Huizhong Duan, Hongning Wang, and ChengXiang Zhai. Exploiting structured ontology to organize scattered online opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 734–742. ACL, 2010.

[578] Yue Lu, Hongning Wang, ChengXiang Zhai, and Dan Roth. Unsupervised discovery of opposing opinion networks from forum discussions. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1642–1646. ACM, 2012.

[579] Mianwei Zhou, Hongning Wang, and KC-C Change. Learning to rank from distant supervision: Exploiting noisy redundancy for relational entity search. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 829–840. IEEE, 2013.

[580] Mohammad Al Boni, Keira Qi Zhou, Hongning Wang, and Matthew S Gerber. Model adaptation for personalized opinion analysis. *Volume 2: Short Papers*, pages 769–774, 2015.

[581] Hongning Wang, Xiaodong He, Ming-Wei Chang, Yang Song, Ryen W. White, and Wei Chu. Personalized ranking model adaptation for web search. In *Proceedings of the 36th International ACM SIGIR Conference*, pages 323–332. ACM, 2013.

[582] Hongning Wang, Minlie Huang, Shilin Ding, and Xiaoyan Zhu. Exploiting and integrating rich features for biological literature classification. *BMC bioinformatics*, 9(Suppl 3):S4, 2008.

[583] Yanen Li, Anlei Dong, Hongning Wang, Hongbo Deng, Yi Chang, and ChengXiang Zhai. A two-dimensional click model for query auto-completion. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 455–464. ACM, 2014.

[584] Hongning Wang, ChengXiang Zhai, Feng Liang, Anlei Dong, and Yi Chang. User modeling in search logs via a nonparametric bayesian approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 203–212. ACM, 2014.

[585] Yang Song, Hongning Wang, and Xiaodong He. Adapting deep ranknet for personalized search. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 83–92. ACM, 2014.

[586] Yang Song, Hao Ma, Hongning Wang, and Kuansan Wang. Exploring and exploiting user search behavior on mobile and tablet devices to improve search relevance. In *Proceedings of the 22nd*

international conference on World Wide Web, pages 1201–1212. International World Wide Web Conferences Steering Committee, 2013.

[587] Ryen W White, Wei Chu, Ahmed Hassan, Xiaodong He, Yang Song, and Hongning Wang. Enhancing personalized search by mining and modeling task behavior. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1411–1420. International World Wide Web Conferences Steering Committee, 2013.

[588] Chi Wang, Hongning Wang, Jialu Liu, Ming Ji, Lu Su, Yuguo Chen, and Jiawei Han. On the detectability of node grouping in networks. In *Proceedings of the 13th SIAM International Conference on Data Mining*, pages 713–721. SIAM, 2013.

[589] Hongbo Deng, Jiawei Han, Hao Li, Heng Ji, Hongning Wang, and Yue Lu. Exploring and inferring user–user pseudo-friendship for sentiment analysis with heterogeneous networks. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7(4):308–321, 2014.

[590] Hongning Wang, Minlie Huang, and Xiaoyan Zhu. Extract interaction detection methods from the biological literature. *BMC bioinformatics*, 10(Suppl 1):S55, 2009.

[591] Hongning Wang, Duo Zhang, and ChengXiang Zhai. Structural topic model for latent topical structure analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1526–1535. Association for Computational Linguistics, 2011.

[592] Minlie Huang, Shilin Ding, Hongning Wang, and Xiaoyan Zhu. Mining physical protein-protein interactions from the literature. *Genome Biol*, 9(Suppl 2):S12, 2008.

[593] Asif Salekin, Hongning Wang, Kristine Williams, and John Stankovic. Dave: Detecting agitated vocal events. In *Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2017 IEEE/ACM International Conference on*, pages 157–166. IEEE, 2017.

[594] Sarah Masud Preum, Abu Sayeed Mondol, Meiyi Ma, Hongning Wang, and John A Stankovic. Preclude: Conflict detection in textual health advice. In *Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on*, pages 286–296. IEEE, 2017.

[595] Asif Salekin, Hongning Wang, and John Stankovic. Kinvocal: Detecting agitated vocal events. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 459–460. ACM, 2015.

[596] Huazheng Wang, Sonwoo Kim, Eric McCord-Snook, Qingyun Wu, and Hongning Wang. 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*, SIGIR'19, pages 835–844. ACM, 2019.

[597] Lu Lin, Lin Gong, and Hongning Wang. Learning personalized topical compositions with item response theory. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 609–617. ACM, 2019.

[598] Yiyi Tao, Yiling Jia, Nan Wang, and Hongning Wang. The fact: Taming latent factor models for explainability with factorization trees. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 295–304, New York, NY, USA, 2019. ACM.

[599] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. Intent-aware query obfuscation for privacy protection in personalized web search. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 285–294. ACM, 2018.

[600] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. Context attentive document ranking and query suggestion. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 385–394. ACM, 2019.

[601] Puxuan Yu, Wasi Uddin Ahmad, and Hongning Wang. Hide-n-seek: An intent-aware privacy protection plugin for personalized web search. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1333–1336. ACM, 2018.

[602] Qingyun Wu, Zhige Li, Huazheng Wang, Wei Chen, and Hongning Wang. Factorization bandits for online influence maximization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 636–646. ACM, 2019.

[603] Yi Qi, Qingyun Wu, Hongning Wang, Jie Tang, and Maosong Sun. Bandit learning with implicit feedback. In *Advances in Neural Information Processing Systems*, pages 7276–7286, 2018.

[604] Qingyun Wu, Huazheng Wang, Yanen Li, and Hongning Wang. Dynamic ensemble of contextual bandits to satisfy users' changing interests. In *The World Wide Web Conference*, pages 2080–2090. ACM, 2019.

[605] Duhigg Charles. How companies learn your secrets. Feb 16, 2012.

[606] Simon P Anderson and Regis Renault. Pricing, product diversity, and search costs: A bertrand-chamberlin-diamond model. *The RAND Journal of Economics*, pages 719–735, 1999.

[607] Steven Berry, James Levinsohn, and Ariel Pakes. Automobile prices in market equilibrium. *Econometrica: Journal of the Econometric Society*, pages 841–890, 1995.

[608] W Lance Bennett. The personalization of politics political identity, social media, and changing patterns of participation. *The ANNALS of the American Academy of Political and Social Science*, 644(1):20–39, 2012.

[609] Shengliang Xu, Shenghua Bao, Ben Fei, Zhong Su, and Yong Yu. Exploring folksonomy for personalized search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 155–162. ACM, 2008.

[610] Michael J Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.

[611] Felipe Saint-Jean, Aaron Johnson, Dan Boneh, and Joan Feigenbaum. Private web search. In *Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 84–90. ACM, 2007.

[612] Josep Domingo-Ferrer, Maria Bras-Amorós, Qianhong Wu, and Jesús Manjón. User-private information retrieval based on a peer-to-peer community. *Data & Knowledge Engineering*, 68(11):1237–1252, 2009.

[613] Michael Barbaro, Tom Zeller, and Saul Hansell. A face is exposed for aol searcher no. 4417749. *New York Times*, 9(2008):8For, 2006.

[614] Jordi Castellá-Roca, Alexandre Viejo, and Jordi Herrera-Joancomartí. Preserving user's privacy in web search engines. *Computer Communications*, 32(13):1541–1551, 2009.

[615] Michael K Reiter and Aviel D Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security (TISSEC)*, 1(1):66–92, 1998.

[616] Vincent Toubiana, Arvind Narayanan, Dan Boneh, Helen Nissenbaum, and Solon Barocas. Adnostic: Privacy preserving targeted advertising. In *Proceedings Network and Distributed System Symposium*, 2010.

[617] Saikat Guha, Bin Cheng, and Paul Francis. Privad: Practical privacy in online advertising. In *NSDI*, 2011.

[618] John Canny. Collaborative filtering with privacy. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 45–57. IEEE, 2002.

[619] Business Insider. The social-media demographics report: Differences in gender, age, and income at the top networks, October 2014.

[620] Expedia Inc. Personalize expedia hotel searches - icdm 2013, 2013.

[621] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[622] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19:629–679, 1994.

[623] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[624] Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 1–9. ACM, 2008.

[625] Anthony Wood, John A Stankovic, Gilles Virone, Leo Selavo, Zhimin He, Qiuhua Cao, Thao Doan, Yafeng Wu, Lei Fang, and Radu Stoleru. Context-aware wireless sensor networks for assisted living and residential monitoring. *Network, IEEE*, 22(4):26–33, 2008.

[626] Janyce Wiebe and Rada Mihalcea. Word sense and subjectivity. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1065–1072. Association for Computational Linguistics, 2006.

[627] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, 2010.

[628] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

[629] David A Shamma, Lyndon Kennedy, and Elizabeth F Churchill. Tweet the debates: understanding community annotation of uncollected sources. In *Proceedings of the first SIGMM workshop on Social media*, pages 3–10. ACM, 2009.

[630] Ronald C Kessler and Philip S Wang. The descriptive epidemiology of commonly occurring mental disorders in the united states*. *Annu. Rev. Public Health*, 29:115–129, 2008.

[631] Ronald C Kessler, Patricia Berglund, Olga Demler, Robert Jin, Kathleen R Merikangas, and Ellen E Walters. Lifetime prevalence and age-of-onset distributions of dsm-iv disorders in the national comorbidity survey replication. *Archives of general psychiatry*, 62(6):593–602, 2005.

[632] Kenneth Wells, Ruth Klap, Alan Koike, and Cathy Sherbourne. Ethnic disparities in unmet need for alcoholism, drug abuse, and mental health care. *American Journal of Psychiatry*, 158(12):2027–2032, 2001.

[633] Patrick Corrigan. How stigma interferes with mental health care. *American Psychologist*, 59(7):614, 2004.

[634] Benjamin L Hankin, Lyn Y Abramson, Terrie E Moffitt, Phil A Silva, Rob McGee, and Kathryn E Angell. Development of depression from preadolescence to young adulthood: emerging gender differences in a 10-year longitudinal study. *Journal of abnormal psychology*, 107(1):128, 1998.

[635] David J Kupfer, Ellen Frank, and James M Perel. The advantage of early treatment intervention in recurrent depression. *Archives of General Psychiatry*, 46(9):771, 1989.

[636] Meg M Reuland and Bethany A Teachman. Interpretation bias modification for youth and their parents: A novel treatment for early adolescent social anxiety. *Journal of Anxiety Disorders*, 2014.

[637] Elise M Clerkin, Christopher R Fisher, Jeffrey W Sherman, and Bethany A Teachman. Applying the quadruple process model to evaluate change in implicit attitudinal responses during therapy for panic disorder. *Behaviour research and therapy*, 52:17–25, 2014.

[638] Shari A Steinman and Bethany A Teachman. Reaching new heights: Comparing interpretation bias modification to exposure therapy for extreme height fear. *Journal of consulting and clinical psychology*, 82(3):404, 2014.

[639] Bethany A Teachman, Craig D Marker, and Elise M Clerkin. Catastrophic misinterpretations as a predictor of symptom change during treatment for panic disorder. *Journal of consulting and clinical psychology*, 78(6):964, 2010.

[640] Sabine Wilhelm, Gail Steketee, Jeanne M Fama, Ulrike Buhlmann, Bethany A Teachman, and Elana Golan. Modular cognitive therapy for obsessive-compulsive disorder: A wait-list controlled trial. *Journal of cognitive psychotherapy*, 23(4):294, 2009.

[641] Bethany A Teachman, Craig D Marker, and Shannan B Smith-Janik. Automatic associations and panic disorder: trajectories of change over the course of treatment. *Journal of consulting and clinical psychology*, 76(6):988, 2008.

[642] Elise M Clerkin, Bethany A Teachman, and Shannan B Smith-Janik. Sudden gains in group cognitive-behavioral therapy for panic disorder. *Behaviour research and therapy*, 46(11):1244–1250, 2008.

[643] Michael W Otto, Bethany A Teachman, Lee S Cohen, Claudio N Soares, Allison F Vitonis, and Bernard L Harlow. Dysfunctional attitudes and episodes of major depression: Predictive validity and temporal stability in never-depressed, depressed, and recovered women. *Journal of Abnormal Psychology*, 116(3):475, 2007.

[644] Gerald J Haeffel, Lyn Y Abramson, Paige C Brazy, James Y Shah, Bethany A Teachman, and Brian A Nosek. Explicit and implicit cognition: A preliminary test of a dual-process theory of cognitive vulnerability to depression. *Behaviour research and therapy*, 45(6):1155–1167, 2007.

[645] Bethany A Teachman and Sheila R Woody. Automatic processing in spider phobia: implicit fear associations over the course of treatment. *Journal of Abnormal Psychology*, 112(1):100, 2003.

[646] Jena M Saporito, C Ryan, and BA Teachman. Reducing stigma toward seeking mental health treatment among adolescents. *Stigma research and action*, 1(2):9, 2011.

[647] Tara S Peris, Bethany A Teachman, and Brian A Nosek. Implicit and explicit stigma of mental illness: Links to clinical care. *The Journal of nervous and mental disease*, 196(10):752–760, 2008.

[648] Shahriar Nirjon, Robert F Dickerson, Philip Asare, Qiang Li, Dezhi Hong, John A Stankovic, Pan Hu, Guobin Shen, and Xiaofan Jiang. Auditeur: A mobile-cloud service platform for acoustic event detection on smartphones. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 403–416. ACM, 2013.

[649] Jiska Cohen-Mansfield. Instruction manual for the cohen-mansfield agitation inventory (cmai). *Research Institute of the Hebrew Home of Greater Washington*, 1991.

[650] James Baker. The dragon system–an overview. *Acoustics, speech and signal processing, IEEE transactions on*, 23(1):24–29, 1975.

[651] Steve J Young and Sj Young. *The HTK hidden Markov model toolkit: Design and philosophy*. Citeseer, 1993.

[652] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.

[653] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[654] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[655] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29. ACM, 1996.

[656] Thorsten Joachims. *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[657] Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Computational linguistics and intelligent text processing*, pages 136–145. Springer, 2002.

[658] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[659] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM, 1986.

[660] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics, 2004.

[661] Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[662] Trevor Cohen and Dominic Widdows. Empirical distributional semantics: methods and biomedical applications. *Journal of biomedical informatics*, 42(2):390–405, 2009.

[663] Enrique Alfonseca and Suresh Manandhar. Extending a lexical ontology by a combination of distributional semantics signatures. In *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, pages 1–7. Springer, 2002.

[664] L Douglas Baker and Andrew Kachites McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103. ACM, 1998.

[665] Karen Rose, Janet Specht, and Windy Forch. Correlates among nocturnal agitation, sleep, and urinary incontinence in dementia. *American journal of Alzheimer's disease and other dementias*, page 1533317514524814, 2014.

[666] Karen M Rose, Cornelia Beck, Pao-Feng Tsai, Pham H Liem, David G Davila, Morton Kleban, Nalaka S Gooneratne, Gurpreet Kalra, and Kathy Culpepper Richards. Sleep disturbances and nocturnal agitation behaviors in older adults with dementia. *Sleep*, 34(6):779, 2011.

[667] Kathy C Richards, Corinne Lambert, Cornelia K Beck, Donald L Bliwise, William J Evans, Gurpreet K Kalra, Morton H Kleban, Rebecca Lorenz, Karen Rose, and Nalaka S Gooneratne. Strength training, walking, and social activity improve sleep in nursing home and assisted living residents: randomized controlled trial. *Journal of the American Geriatrics Society*, 59(2):214–223, 2011.

[668] Heejung Kim, Mido Chang, Karen Rose, and Sunha Kim. Predictors of caregiver burden in caregivers of individuals with dementia. *Journal of advanced nursing*, 68(4):846–855, 2012.

[669] John C Loehlin. *Latent variable models: An introduction to factor, path, and structural analysis* . Lawrence Erlbaum Associates Publishers, 1998.

[670] Martin Knott and David J Bartholomew. *Latent variable models and factor analysis*. Number 7. Edward Arnold, 1999.

[671] Daniel C Fain and Jan O Pedersen. Sponsored search: A brief history. *Bulletin of the American Society for Information Science and Technology*, 32(2):12–13, 2006.

[672] Soo Young Rieh. Judgment of information quality and cognitive authority in the web. *Journal of the American Society for Information Science and Technology*, 53(2):145–161, 2002.

[673] Eran Toch, Yang Wang, and Lorrie Faith Cranor. Personalization and privacy: a survey of privacy risks and remedies in personalization-based systems. *User Modeling and User-Adapted Interaction*, 22(1-2):203–220, 2012.

[674] Jane Huffman Hayes and Alex Dekhtyar. Humans in the traceability loop: can't live with'em, can't live without'em. In *Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering*, pages 20–23. ACM, 2005.

[675] Hsin-Han Chiang, Shinq-Jen Wu, Jau-Woei Perng, Bing-Fei Wu, and Tsu-Tian Lee. The human-in-the-loop design approach to the longitudinal automation system for an intelligent vehicle. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 40(4):708–720, 2010.

[676] Randy L Sollenberger, Ben Willems, Pamela S Della Rocco, Anton Koros, and Todd Truitt. Human-in-the-loop simulation evaluating the collocation of the user request evaluation tool, traffic management advisor, and controller-pilot data link communications: Experiment i–tool combinations. 2005.

[677] Bonnie A Nardi. *Context and consciousness: activity theory and human-computer interaction*. Mit Press, 1996.

[678] Sean W Smith. Humans in the loop: Human-computer interaction and security. *Security & Privacy, IEEE*, 1(3):75–79, 2003.

[679] David Tennenhouse. Proactive computing. *Communications of the ACM*, 43(5):43–50, 2000.

[680] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[681] David Lyon. *Surveillance as social sorting: Privacy, risk, and digital discrimination*. Psychology Press, 2003.

[682] Andrew Odlyzko. Privacy, economics, and price discrimination on the internet. In *Proceedings of the 5th international conference on Electronic commerce*, pages 355–366. ACM, 2003.

[683] Avi Goldfarb and Catherine E Tucker. Privacy regulation and online advertising. *Management Science*, 57(1):57–71, 2011.

[684] Avi Goldfarb and Catherine Tucker. Online display advertising: Targeting and obtrusiveness. *Marketing Science*, 30(3):389–404, 2011.

[685] Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80. ACM, 2005.

[686] Judith L Meece, Phyllis C Blumenfeld, and Rick H Hoyle. Students' goal orientations and cognitive engagement in classroom activities. *Journal of educational psychology*, 80(4):514, 1988.

[687] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. In *ACm SIGIR Forum*, volume 33, pages 6–12. ACM, 1999.

[688] Michael Bendersky and W Bruce Croft. Analysis of long queries in a large scale search log. In *Proceedings of the 2009 workshop on Web Search Click Data*, pages 8–14. ACM, 2009.

[689] Benjamin Piwowarski and Hugo Zaragoza. Predictive user click models based on click-through history. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 175–182. ACM, 2007.

[690] Jay M Ponte and W Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998.

[691] Shin-jeng Lin and Nick Belkin. Validation of a model of information seeking over multiple search sessions. *Journal of the American Society for Information Science and Technology*, 56(4):393–415, 2005.

[692] Diane Kelly and Nicholas J Belkin. Reading time, scrolling and interaction: exploring implicit sources of user preferences for relevance feedback. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 408–409. ACM, 2001.

[693] Scott Henninger and Nick Belkin. Interface issues and interaction strategies for information retrieval systems. In *Conference companion on Human factors in computing systems*, pages 401–402. ACM, 1995.

[694] Marti Hearst. *Search user interfaces*. Cambridge University Press, 2009.

[695] George W Furnas, Thomas K Landauer, Louis M Gomez, and Susan T Dumais. Human factors and behavioral science: Statistical semantics: Analysis of the potential performance of key-word information systems. *Bell System Technical Journal, The*, 62(6):1753–1806, 1983.

[696] Gavriel Salvendy. *Handbook of human factors and ergonomics*. John Wiley & Sons, 2012.

[697] Nicholas J Belkin. Anomalous states of knowledge as a basis for information-retrieval. *Canadian Journal of Information Science-Revue Canadienne Des Sciences De L Information*, 5(MAY):133–143, 1980.

[698] Jingjing Liu and Nicholas J Belkin. Personalizing information retrieval for multi-session tasks: The roles of task stage and task type. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 26–33. ACM, 2010.

[699] Jingjing Liu, Nicholas J Belkin, Xiangmin Zhang, and Xiaojun Yuan. Examining users' knowledge change in the task completion process. *Information Processing & Management*, 49(5):1058–1074, 2013.

[700] Kristin Purcell, Joanna Brenner, and Lee Rainie. Search engine use 2012. 2012.

[701] Gediminas Adomavicius and Alexander Tuzhilin. Using data mining methods to build customer profiles. *Computer*, (2):74–82, 2001.

[702] Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, pages 675–684. ACM, 2004.

[703] Roger Dingledine. Tor and circumvention: Lessons learned. In *Advances in Cryptology–CRYPTO 2011*, pages 485–486. Springer, 2011.

[704] Liyue Fan, Luca Bonomi, Li Xiong, and Vaidy Sunderam. Monitoring web browsing behavior with differential privacy. In *Proceedings of the 23rd international conference on World wide web*, pages 177–188. ACM, 2014.

[705] Jun Yan, Ning Liu, Gang Wang, Wen Zhang, Yun Jiang, and Zheng Chen. How much can behavioral targeting help online advertising? In *Proceedings of the 18th international conference on World wide web*, pages 261–270. ACM, 2009.

[706] Ye Chen, Dmitry Pavlov, and John F Canny. Large-scale behavioral targeting. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 209–218. ACM, 2009.

[707] Michael G Reed, Paul F Syverson, and David M Goldschlag. Anonymous connections and onion routing. *Selected Areas in Communications, IEEE Journal on*, 16(4):482–494, 1998.

[708] Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *Parallel and Distributed Systems, IEEE Transactions on*, 25(1):222–233, 2014.

[709] Ali Inan, Murat Kantarcioglu, Gabriel Ghinita, and Elisa Bertino. Private record matching using differential privacy. In *Proceedings of the 13th International Conference on Extending Database Technology*, pages 123–134. ACM, 2010.

[710] Cynthia Dwork. Differential privacy. In *Encyclopedia of Cryptography and Security*, pages 338–340. Springer, 2011.

[711] Rosie Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. "i know what you did last summer" – query logs and user privacy. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 909–914. ACM, 2007.

[712] Flávio du Pin Calmon and Nadia Fawaz. Privacy against statistical inference. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 1401–1408. IEEE, 2012.

[713] Salman Salamatian, Amy Zhang, Flavio du Pin Calmon, Sandilya Bhamidipati, Nadia Fawaz, Branislav Kveton, Pedro Oliveira, and Nina Taft. How to hide the elephant-or the donkey-in the room: Practical privacy against statistical inference for large data. *IEEE GlobalSIP*, 2013.

[714] Inc. Amazon.com. Amazon.com privacy notice. `http://www.amazon.com/gp/help/customer/display.html?nodeId=468496`, March 3, 2014.

[715] Reuters. European court says google must respect 'right to be forgotten', May 13, 2014. `http://www.reuters.com/article/2014/05/13/us-eu-google-dataprotection-idUSBREA4C07120140513`.

[716] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[717] Mummoorthy Murugesan and Chris Clifton. Plausibly deniable search. In *Workshop on Secure Knowledge Management (SKM 2008)*, 2008.

[718] Mummoorthy Murugesan and Chris Clifton. Providing privacy through plausibly deniable search. In *SDM*, pages 768–779. SIAM, 2009.

[719] Benny Chor and Niv Gilboa. Computationally private information retrieval. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 304–313. ACM, 1997.

[720] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *Journal of the ACM (JACM)*, 45(6):965–981, 1998.

[721] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *focs*, page 364. IEEE, 1997.

[722] William Gasarch. A survey on private information retrieval. In *Bulletin of the EATCS*. Citeseer, 2004.

[723] David Sánchez, Jordi Castellà-Roca, and Alexandre Viejo. Knowledge-based scheme to create privacy-preserving but semantically-related queries for web search engines. *Information Sciences*, 218:17–30, 2013.

[724] HweeHwa Pang, Xuhua Ding, and Xiaokui Xiao. Embellishing text search queries to protect user privacy. *Proceedings of the VLDB Endowment*, 3(1-2):598–607, 2010.

[725] John Bethencourt, Dawn Song, and Brent Waters. New constructions and practical applications for private stream searching. In *Security and Privacy, 2006 IEEE Symposium on*, pages 134–139. IEEE, 2006.

[726] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *Advances in Cryptology-Eurocrypt 2004*, pages 506–522. Springer, 2004.

[727] Michael J Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In *Theory of Cryptography*, pages 303–324. Springer, 2005.

[728] Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*, pages 44–55. IEEE, 2000.

[729] Yun Zhu, Li Xiong, and Christopher Verdery. Anonymizing user profiles for personalized web search. In *Proceedings of the 19th international conference on World wide web*, pages 1225–1226. ACM, 2010.

[730] Daniel C Howe and Helen Nissenbaum. Trackmenot: Resisting surveillance in web search. *Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society*, 23:417–436, 2009.

[731] Dan Cvrcek, Marek Kumpost, Vashek Matyas, and George Danezis. A study on the value of location privacy. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 109–118. ACM, 2006.

[732] Omer Tene. What google knows: Privacy and internet search engines. *Utah Law Review, Forthcoming*, 2007.

[733] Katie Hafner and Matt Richtel. Google resists us subpoena of search data. *New York Times*, 20:A1, 2006.

[734] EU data protection working party. Opinion on data protection issues related to search engines. 2008.

[735] Meredith Ringel Morris, Jaime Teevan, and Steve Bush. Enhancing collaborative web search with personalization: groupization, smart splitting, and group hit-highlighting. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 481–484. ACM, 2008.

[736] Yuchen Zhang, Weizhu Chen, Dong Wang, and Qiang Yang. User-click modeling for understanding and predicting search-behavior. In *KDD'2011*, pages 1388–1396. ACM, 2011.

[737] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR'06*, pages 3–10. ACM, 2006.

[738] R.M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

[739] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[740] Antonio Gulli and Alessio Signorini. The indexable web is more than 11.5 billion pages. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 902–903. ACM, 2005.

[741] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[742] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 697–702. IEEE, 2007.

[743] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[744] Peter Bruza, Robert McArthur, and Simon Dennis. Interactive internet search: keyword, directory and query reformulation mechanisms compared. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 280–287. ACM, 2000.

[745] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[746] Huseyin Polat and Wenliang Du. Privacy-preserving collaborative filtering using randomized perturbation techniques. *Electrical Engineering and Computer Science*, 2003.

[747] Kuansan Wang, Christopher Thrasher, Evelyne Viegas, Xiaolong Li, and Bo-june Paul Hsu. An overview of microsoft web n-gram corpus and applications. In *Proceedings of the NAACL HLT 2010 Demonstration Session*, pages 45–48. Association for Computational Linguistics, 2010.

[748] Soo Young Rieh and Hong Xieb. Analysis of multiple query reformulations on the web: The interactive information retrieval context. *Information Processing & Management*, 42(3):751–768, 2006.

[749] Tessa Lau and Eric Horvitz. *Patterns of search: analyzing and modeling web query refinement*. Springer, 1999.

[750] Xuanhui Wang and ChengXiang Zhai. Mining term association patterns from search logs for effective query reformulation. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 479–488. ACM, 2008.

[751] Lynda Tamine-Lechani, Mohand Boughanem, and Mariam Daoud. Evaluation of contextual information retrieval effectiveness: overview of issues and research. *Knowledge and Information Systems*, 24(1):1–34, 2010.

[752] Catherine Mulwa, Seamus Lawless, Mary Sharp, and Vincent Wade. The evaluation of adaptive and personalised information retrieval systems: a review. *International Journal of Knowledge and Web Intelligence*, 2(2-3):138–156, 2011.

[753] Guillermo Navarro-Arribas, Vicenç Torra, Arnau Erola, and Jordi Castellà-Roca. User k-anonymity for privacy preserving data mining of query logs. *Information Processing & Management*, 48(3):476–487, 2012.

[754] Peter McCullagh and John A Nelder. *Generalized linear models*, volume 37. CRC press, 1989.

[755] John A Nelder and RJ Baker. Generalized linear models. *Encyclopedia of Statistical Sciences*, 1972.

[756] Paul N Bennett, Krysta Svore, and Susan T Dumais. Classification-enhanced ranking. In *Proceedings of the 19th international conference on World wide web*, pages 111–120. ACM, 2010.

[757] Avi Goldfarb and Catherine E Tucker. Privacy regulation and online advertising. *Management science*, 57(1):57–71, 2011.

[758] Anja Lambrecht and Catherine Tucker. When does retargeting work? information specificity in online advertising. *Journal of Marketing Research*, 50(5):561–576, 2013.

[759] Catherine E Tucker. Social networks, personalized advertising, and privacy controls. *Journal of Marketing Research*, 51(5):546–562, 2014.

[760] Randall A Lewis and Justin M Rao. The unfavorable economics of measuring the returns to advertising. *The Quarterly Journal of Economics*, page qjv023, 2015.

[761] Thomas Blake, Chris Nosko, and Steven Tadelis. Consumer heterogeneity and paid search effectiveness: A large-scale field experiment. *Econometrica*, 83(1):155–174, 2015.

[762] Randall Lewis and Dan Nguyen. Display advertising's competitive spillovers to consumer search. *Quantitative Marketing and Economics*, 13(2):93–115, 2015.

[763] Avi Goldfarb and Catherine Tucker. Online display advertising: Targeting and obtrusiveness. *Marketing Science*, 30(3):389–404, 2011.

[764] Han Hong, Aprajit Mahajan, and Denis Nekipelov. Extremum estimation and numerical derivatives. *Journal of Econometrics*, 188(1):250–263, 2015.

[765] Patrick Bajari, Han Hong, and Denis Nekipelov. Game theory and econometrics: A survey of some recent research. In *Advances in Economics and Econometrics, 10th World Congress*, volume 3, pages 3–52, 2013.

[766] Shuchi Chawla, Jason Hartline, and Denis Nekipelov. Mechanism design for data science. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 711–712. ACM, 2014.

[767] Patrick Bajari, Denis Nekipelov, Stephen P Ryan, and Miaoyu Yang. Machine learning methods for demand estimation. *The American Economic Review*, 105(5):481–485, 2015.

[768] Darrell Hoy, Denis Nekipelov, and Vasilis Syrgkanis. Robust data-driven guarantees in auctions. *SSRN 2601928*, 2015.

[769] Shakeeb Khan and Denis Nekipelov. Information structure and statistical information in discrete response models. *Economic Research Initiatives at Duke (ERID) Working Paper*, (110), 2011.

[770] Yiqun Liu, Yupeng Fu, Min Zhang, Shaoping Ma, and Liyun Ru. Automatic search engine performance evaluation with click-through data analysis. In *Proceedings of the 16th international conference on World Wide Web*, pages 1133–1134. ACM, 2007.

[771] Pavel Serdyukov, Georges Dupret, and Nick Craswell. Log-based personalization: The 4th web search click data (wscd) workshop. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 685–686. ACM, 2014.

[772] Sonia Livingstone. Taking risky opportunities in youthful content creation: teenagers' use of social networking sites for intimacy, privacy and self-expression. *New media & society*, 10(3):393–411, 2008.

[773] Mary Leary. Reasonable expectations of privacy for youth in a digital age. *Mississippi Law Journal*, 80(3):1033–1092, 2011.

[774] Mary Madden, Amanda Lenhart, Sandra Cortesi, Urs Gasser, Maeve Duggan, Aaron Smith, and Meredith Beaton. Teens, social media, and privacy. *Pew Research Center*, 21, 2013.

[775] Y. Zhang and L. El Ghaoui. Large-scale sparse principal component analysis and application to text data. In *Proc.* Conf. Neural Information and Processing Systems, December 2011.

[776] L. El Ghaoui, G. Li, V.A. Duong, V. Pham, A. Srivastava, and K. Bhaduri. Sparse machine learning methods for understanding large text corpora. In *Conference on Intelligent Data Understanding*, October 2011.

[777] L. El Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination for the LASSO. *Pacific Journal of Optimization*, November 2011. Special Issue on Conic Optimization.

[778] B. Gawalt, J. Jia, L. Miratrix, L. El Ghaoui, B. Yu, and S. Clavier. Discovering word associations in news media via feature selection and sparse classification. In *Proc. 11th ACM SIGMM International Conference on Multimedia Information Retrieval*, 2010.

[779] J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals. *IEEE Trans. Inform. Theory*, 51(3):1030–1051, March 2006.

[780] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

[781] A. d'Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet. A direct formulation of sparse PCA using semidefinite programming. *SIAM Review*, 49(3), 2007.

[782] F. Bach, S. D. Ahipasaoglu, and A. d'Aspremont. Convex relaxations for subset selection. *working paper*, 2010.

[783] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[784] S. Berry, J. Levinsohn, and A. Pakes. Automobile prices in market equilibrium. *Econometrica*, 63(4):841–890, 1995.

[785] A. Petrin. Quantifying the benefits of new products: The case of the minivan. *Journal of Political Economy*, 110(4):705–729, 2002.

[786] S. Berry, J. Levinsohn, and A. Pakes. Estimating differentiated product demand systems from a combination of micro and macro data: The market for new vehicles. *Journal of Political Economy*, 112(1):68–105, 2004.

[787] J-P. Dube, J. Fox, and C-L. Su. Improving the numerical performance of blp static and dynamic discrete choice random coefficients demand. *Econometrica*, forthcoming, 2012.

[788] J. Albert and S. Chib. Bayesian analysis of binary and polychotomous data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

[789] P. Rossi, G. Allenby, and R. McCulloch. *Bayesian Statistics and Marketing*. John Wiley & Sons, 2005.

[790] C.C. Aggarwal and C. Zhai. *Mining Text Data*. Springer-Verlag New York Inc, 2012.

[791] D. McFadden. Econometric models for probabilistic choice among products. *Journal of Business*, pages 13–29, 1980.

[792] J.S. Deogun, S.K. Choubey, V.V. Raghavan, and H. Sever. Feature selection and effective classifiers. *Journal of the American Society for Information Science*, 49(5):423–434, 1998.

[793] C.E. Brodley and M.A. Friedl. Identifying and eliminating mislabeled training instances. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 799–805. Citeseer, 1996.

[794] S.K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Arxiv preprint cs/9408103*, 1994.

[795] B.D. Ripley. *Pattern recognition and neural networks*. Cambridge University Press, 2008.

[796] C.J.C. Burges. *Advances in kernel methods: support vector learning*. The MIT press, 1999.

[797] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[798] T. Dietterich. Machine learning for sequential data: A review. *Structural, syntactic, and statistical pattern recognition*, pages 227–246, 2002.

[799] W.K. Newey. The asymptotic variance of semiparametric estimators. *Econometrica: Journal of the Econometric Society*, pages 1349–1382, 1994.

[800] X. Chen, O. Linton, and I. Van Keilegom. Estimation of semiparametric models when the criterion function is not smooth. *Econometrica*, 71(5):1591–1608, 2003.

[801] C. Ai and X. Chen. Efficient estimation of models with conditional moment restrictions containing unknown functions. *Econometrica*, 71(6):1795–1843, 2003.

[802] T. Amemiya. *Advanced econometrics*. Harvard University Press, 1985.

[803] I.A. Ibragimov and R.Z. Khasminskii. Asymptotic theory of estimation. *Nauka, Moscow*, 1979.

[804] P.J. Bickel, CAJ Klaassen, Y. Ritov, and JA Wellner. *Efficient and adaptive inference in semiparametric models*. Johns Hopkins University Press, Baltimore, 1993.

[805] R. Dhar. Consumer preference for a no-choice option. *Journal of Consumer Research*, 24(2):215–231, 1997.

[806] C.J. Anderson. The psychology of doing nothing: Forms of decision avoidance result from reason and emotion. *Psychological Bulletin*, 129(1):139, 2003.

[807] A. Chernev. When more is less and less is more: The role of ideal point availability and assortment in consumer choice. *Journal of Consumer Research*, 30(2):170–183, 2003.

[808] X. Chen, H. Hong, and D. Nekipelov. Nonlinear models of measurement errors. *Journal of Economic Literature*, 49(4):901–937, 2011.

[809] P. Bajari, H. Hong, J. Krainer, and D. Nekipelov. Estimating static models of strategic interactions. *Journal of Business and Economic Statistics*, 28(4):469–482, 2010.

[810] H. Hong and D. Nekipelov. Semiparametric efficiency in nonlinear late models. *Quantitative Economics*, 1(2):279–304, 2010.

[811] H. Hong and D. Nekipelov. Efficient local iv estimation of an empirical auction model. *Journal of Econometrics*, 2011.

[812] A. Beck and M. Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22:557, 2012.

[813] A. Hong, H.and Mahajan and D. Nekipelov. Extremum estimation and numerical derivatives. *Stanford and UC Berkeley Working paper*, 2011.

[814] A. Hong, H.and Mahajan and D. Nekipelov. Numerical differentiation of u-statistics. *Stanford and UC Berkeley Working paper*, 2012.

[815] Khan S. and D. Nekipelov. Information structure and statistical information in discrete response models. *Duke University and UC Berkeley Working paper*, 2012.

[816] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

[817] HB Newcombe, JM Kennedy, SJ Axford, and AP James. Automatic linkage of vital and health records. *Science*, 130:954–959, 1959.

[818] I.P. Fellegi and A.B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, pages 1183–1210, 1969.

[819] D. Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge University Press, 1997.

[820] M.A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, pages 414–420, 1989.

[821] W.E. Winkler. The state of record linkage and current research problems. In *Statistical Research Division, US Census Bureau*. Citeseer, 1999.

[822] G. Salton and D. Harman. *Information retrieval*. John Wiley and Sons Ltd., 2003.

[823] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani. Robust and efficient fuzzy match for online data cleaning. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 313–324. ACM, 2003.

[824] G. Ridder and R. Moffitt. The econometrics of data combination. *Handbook of Econometrics*, 6:5469–5547, 2007.

[825] A. Acquisti. Privacy and security of personal information. *Economics of Information Security*, pages 179–186, 2004.

[826] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty Fuzziness and Knowledge Based Systems*, 10(5):557–570, 2002.

[827] A. Goldfarb and C. Tucker. Online display advertising: Targeting and obtrusiveness. *Marketing Science*, 2010.

[828] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60. ACM, 2005.

[829] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference*, pages 25–25. IEEE, 2006.

[830] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 10(5):571–588, 2002.

[831] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation algorithms for k-anonymity. *Journal of Privacy Technology*, 2005112001, 2005.

[832] V. Ciriani, S.D.C. di Vimercati, S. Foresti, and P. Samarati. k-anonymity. *Secure Data Management in Decentralized Systems. Springer-Verlag*, 2007.

[833] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, Citeseer, 1998.

[834] G.T. Duncan and D. Lambert. Disclosure-limited data dissemination. *Journal of the American statistical association*, 81(393):10–18, 1986.

[835] G.T. Duncan and S. Mukherjee. Microdata disclosure limitation in statistical databases: Query sizeand random sample query control. 1991.

[836] G.T. Duncan and R.W. Pearson. Enhancing access to microdata while protecting confidentiality: Prospects for the future. *Statistical Science*, pages 219–232, 1991.

[837] S.E. Fienberg. Conflicts between the needs for access to statistical information and demands for confidentiality. *Journal of Official Statistics*, 10:115–115, 1994.

[838] S.E. Fienberg. Statistical perspectives on confidentiality and data access in public health. *Statistics in medicine*, 20(9-10):1347–1356, 2001.

[839] G.T. Duncan, S.E. Fienberg, R. Krishnan, R. Padman, and S.F. Roehrig. Disclosure limitation methods and information loss for tabular data. *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies*, pages 135–166, 2001.

[840] J.L. Horowitz and C.F. Manski. Identification and estimation of statistical functionals using incomplete data. *Journal of Econometrics*, 132(2):445–459, 2006.

[841] J.L. Horowitz, C.F. Manski, M. Ponomareva, and J. Stoye. Computation of bounds on population parameters when the data are incomplete. *Reliable computing*, 9(6):419–440, 2003.

[842] J.M. Abowd and S.D. Woodcock. Disclosure limitation in longitudinal linked data. *Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies*, pages 215–277, 2001.

[843] C. Dwork. Differential privacy. *Automata, languages and programming*, pages 1–12, 2006.

[844] J. Abowd and L. Vilhuber. How protective are synthetic data? In *Privacy in Statistical Databases*, pages 239–246. Springer, 2008.

[845] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Advances in Cryptology–CRYPTO 2004*, pages 134–138. Springer, 2004.

[846] D. Lambert. Measures of disclosure risk and harm. *Journal of Official Statistics*, 9:313–313, 1993.

[847] H.R. Varian. Economic aspects of personal privacy. *Internet Policy and Economics*, pages 101–109, 2009.

[848] A. Acquisti and H.R. Varian. Conditioning prices on purchase history. *Marketing Science*, pages 367–381, 2005.

[849] A. Acquisti and J. Grossklags. What can behavioral economics teach us about privacy. *Digital Privacy: Theory, Technologies, and Practices*, pages 363–377, 2008.

[850] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80. ACM, 2005.

[851] A. Acquisti, A. Friedman, and R. Telang. Is there a cost to privacy breaches? an event study. In *Fifth Workshop on the Economics of Information Security*. Citeseer, 2006.

[852] A.R. Miller and C. Tucker. Privacy protection and technology diffusion: The case of electronic medical records. *Management Science*, 55(7):1077–1093, 2009.

[853] G. Calzolari and A. Pavan. On the optimality of privacy in sequential contracting. *Journal of Economic Theory*, 130(1):168–204, 2006.

[854] C.R. Taylor. Consumer privacy and the market for customer information. *RAND Journal of Economics*, pages 631–650, 2004.

[855] A. Korolova. Privacy violations using microtargeted ads: A case study. In *IEEE International Workshop on Privacy Aspects of Data Mining (PADM'2010)*, pages 474–482, 2010.

[856] C.F. Manski. *Partial identification of probability distributions*. Springer Verlag, 2003.

[857] T. Magnac and E. Maurin. Partial identification in monotone binary models: discrete regressors and interval data. *Review of Economic Studies*, 75(3):835–864, 2008.

[858] I.P. Fellegi and A.B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, pages 1183–1210, 1969.

[859] Glenn Wright. Probabilistic record linkage in sas®. *Keiser Permanente, Oakland, CA*, 2010.

[860] C.J. Bradley, L. Penberthy, K.J. Devers, and D.J. Holden. Health services research and data linkages: issues, methods, and directions for the future. *Health services research*, 45(5(2)):1468–1488, 2010.

[861] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J.V. Pearson, D.A. Stephan, S.F. Nelson, and D.W. Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS Genetics*, 4(8):e1000167, 2008.

[862] A.G. Wilson, T.L. Graves, M.S. Hamada, and C.S. Reese. Advances in data combination, analysis and collection for system reliability assessment. *Statistical Science*, 21(4):514–531, 2006.

[863] T. Komarova, D. Nekipelov, and E. Yakovlev. Identification, data combination and the risk of disclosure. *CeMMAP working papers*, 2011.

[864] I. Molchanov. *Theory of random sets*. Springer, 2005.

[865] A. Beresteanu, I. Molchanov, and F. Molinari. Sharp identification regions in models with convex moment predictions. *Econometrica*, 79(6):1785–1821, 2011.

[866] A. Beresteanu, I. Molchanov, and F. Molinari. Partial identification using random set theory. *Journal of Econometrics*, 166(1):17–32, 2012.

[867] A. Beresteanu and F. Molinari. Asymptotic properties for a class of partially identified models. *Econometrica*, 76(4):763–814, 2008.

[868] F. Molinari. Missing treatments. *Journal of Business & Economic Statistics*, 28(1):82–95, 2010.

[869] C.F. Manski and F. Molinari. Rounding probabilistic expectations in surveys. *Journal of Business & Economic Statistics*, 28(2):219–231, 2010.

[870] F. Molinari. Partial identification of probability distributions with misclassified data. *Journal of Econometrics*, 144(1):81–117, 2008.

[871] C.F. Manski and F. Molinari. Skip sequencing: A decision problem in questionnaire design. *The annals of applied statistics*, 2(1):264, 2008.

[872] F. Molinari and A.M. Rosen. Comment on “the identification power of equilibrium in simple games". *Journal of Business and Economic Statistics*, 26(3):297–302, 2008.

[873] L. Barseghyan, F. Molinari, T. O’Donoghue, and J. Teitelbaum. The nature of risk preferences: Evidence from insurance choices. *American Economic Review (forthcoming)*, 2012.

[874] T. Komarova. Binary choice models with discrete regressors: Identification and misspecification. *Journal of Econometrics (revise and resubmit)*, 2012.

[875] J.J. Kim. A method for limiting disclosure in microdata based on random noise and transformation. *Proceedings on the Section on Survey Research Methods*, pages 303–308, 1986.

[876] A. Oganian. Multiplicative noise protocols. 6344:107–117, 2010.

[877] A. Oganian. Multiplicative noise for masking numerical microdata with constraints. *SORT-Statistics and Operations Research Transactions*, 35:99–112, 2011.

[878] W.A. Fuller. Masking procedures for microdata disclosure limitation. *Journal of Official Statistics*, 9(2):383–406, 1993.

[879] C. Skinner. Statistical disclosure control for survey data. 29A:381–396, 2009.

[880] D.B. Rubin. *Multiple Imputation for Nonresponse in Surveys*, volume 29A. Wiley, New York, 1987.

[881] D.B. Rubin. Discussion of statistical disclosure limitation. *Journal of Official Statistics*, 9(2):461–468, 1993.

[882] J.P. Reiter. Satisfying disclosure restriction with synthetic data sets. *Journal of Official Statistics*, 18(4):531–543, 2002.

[883] Tatiana Komarova, Denis Nekipelov, and Evgeny Yakovlev. Estimation of treatment effects from combined data. *Economic Analysis of the Digital Economy*, page 279, 2015.

[884] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[885] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49. ACM, 2004.

[886] Fei Song and W Bruce Croft. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321. ACM, 1999.

[887] Michael McCandless, Erik Hatcher, and Otis Gospodnetic. *Lucene in Action: Covers Apache Lucene 3.0*. Manning Publications Co., 2010.

[888] Patrick Bajari, Han Hong, John Krainer, and Denis Nekipelov. Estimating static models of strategic interactions. *Journal of Business & Economic Statistics*, 28(4):469–482, 2010.

[889] Pooya Jalaly, Denis Nekipelov, and Eva Tardos. Learning and trust in auction markets. *arXiv preprint arXiv:1703.10672*, 2017.

[890] Aniko Hannak, Gary Soeller, David Lazer, Alan Mislove, and Christo Wilson. Measuring price discrimination and steering on e-commerce web sites. In *Proceedings of the 2014 conference on internet measurement conference*, pages 305–318. ACM, 2014.

[891] Nicholas Diakopoulos. Algorithmic accountability: Journalistic investigation of computational power structures. *Digital Journalism*, 3(3):398–415, 2015.

[892] Denis Nekipelov, Vasilis Syrgkanis, and Eva Tardos. Econometrics for learning agents. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 1–18. ACM, 2015.

[893] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[894] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 429–438. IEEE, 2013.

[895] Mauro Conti, Vittoria Cozza, Marinella Petrocchi, and Angelo Spognardi. Trap: using targeted ads to unveil google personal profiles. In *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*, pages 1–6. IEEE, 2015.

[896] Stephen R Lawrence. Personalization of web search results using term, category, and link-based user profiles, November 11 2011. US Patent App. 13/295,019.

[897] Shihao Ji, Anlei Dong, Ciya Liao, Yi Chang, Zhaohui Zheng, Olivier Chapelle, Gordon Guo-Zheng Sun, and Hongyuan Zha. Global and topical ranking of search results using user clicks, July 31 2009. US Patent App. 12/533,564.

[898] Pol Mac Aonghusa and Douglas J Leith. Plausible deniability in web search–from detection to assessment. *arXiv preprint arXiv:1703.03471*, 2017.

[899] Aniko Hannak, Piotr Sapiezynski, Arash Molavi Kakhki, Balachander Krishnamurthy, David Lazer, Alan Mislove, and Christo Wilson. Measuring personalization of web search. In *Proceedings of the 22nd international conference on World Wide Web*, pages 527–538. ACM, 2013.

[900] Guillermo A Calvo. Staggered prices in a utility-maximizing framework. *Journal of monetary Economics*, 12(3):383–398, 1983.

[901] Yiling Chen and David M Pennock. A utility framework for bounded-loss market makers. *arXiv preprint arXiv:1206.5252*, 2012.

[902] Jinxi Xu and W Bruce Croft. Quary expansion using local and global document analysis. In *ACM SIGIR Forum*, volume 51, pages 168–175. ACM, 2017.

[903] Ellen M Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69. Springer-Verlag New York, Inc., 1994.

[904] Bernard Widrow and Michael A Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442, 1990.

[905] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[906] Yinglian Xie and David O'Hallaron. Locality in search engine queries and its implications for caching. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1238–1247. IEEE, 2002.

[907] Hal R Varian. Economic aspects of personal privacy. In *Internet Policy and Economics*, pages 101–109. Springer, 2009.

[908] Marcel K Richter. Revealed preference theory. *Econometrica*, 34(3):635, 1966.

[909] Patrick Bajari, Chenghuan Sean Chu, Denis Nekipelov, and Minjung Park. Identification and semiparametric estimation of a finite horizon dynamic discrete choice model with a terminating action. *Quantitative Marketing and Economics*, 14(4):271–323, 2016.

[910] Michael J. Pazzani and Daniel Billsus. The adaptive web. chapter Content-based Recommendation Systems, pages 325–341. Springer-Verlag, Berlin, Heidelberg, 2007.

[911] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[912] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 165–172, New York, NY, USA, 2013. ACM.

[913] Xiaolin Zheng, Weifeng Ding, Zhen Lin, and Chaochao Chen. Topic tensor factorization for recommender system. *Inf. Sci.*, 372(C):276–293, December 2016.

[914] Lin Xiao, Zhang Min, and Yongfeng Zhang. *Joint Factorizational Topic Models for Cross-City Recommendation*. 07 2017.

[915] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, pages 535–541, Cambridge, MA, USA, 2000. MIT Press.

[916] Jasson D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 713–719, New York, NY, USA, 2005. ACM.

[917] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 880–887, New York, NY, USA, 2008. ACM.

[918] Yongfeng Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. Improve collaborative filtering through bordered block diagonal form matrices. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 313–322, New York, NY, USA, 2013. ACM.

[919] Yongfeng Zhang, Min Zhang, Yiqun Liu, Shaoping Ma, and Shi Feng. Localized matrix factorization for recommendation based on matrix block diagonal forms. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1511–1520, New York, NY, USA, 2013. ACM.

[920] Behnoush Abdollahi and Olfa Nasraoui. Explainable restricted boltzmann machines for collaborative filtering. *CoRR*, abs/1606.07129, 2016.

[921] K. Amarouche, H. Benbrahim, and I. Kassou. Product opinion mining for competitive intelligence. In *Procedia Computer Science*, volume 73, pages 358–365, 2015. cited By 1.

[922] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2, January 2009.

[923] Yongfeng Zhang and Xu Chen. Explainable recommendation: A survey and new perspectives. *CoRR*, abs/1804.11192, 2018.

[924] Amit Sharma and Dan Cosley. Do social explanations work?: Studying and modeling the effects of social explanations in recommender systems. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1133–1144, New York, NY, USA, 2013. ACM.

[925] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. Social collaborative viewpoint regression with explainable recommendations. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 485–494, New York, NY, USA, 2017. ACM.

[926] Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: Explaining recommendations using tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 47–56, New York, NY, USA, 2009. ACM.

[927] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan Kankanhalli. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 639–648, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[928] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. Social collaborative viewpoint regression with explainable recommendations. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 485–494, New York, NY, USA, 2017. ACM.

[929] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[930] Xiaowen Ding, Bing Liu, and Philip S. Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 231–240, New York, NY, USA, 2008. ACM.

[931] Owen Phelan, Kevin McCarthy, Mike Bennett, and Barry Smyth. Terms of a feather: Content-based news recommendation and discovery using twitter. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, pages 448–459, Berlin, Heidelberg, 2011. Springer-Verlag.

[932] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1661–1670. ACM, 2015.

[933] Nava Tintarev and Judith Masthoff. Designing and evaluating explanations for recommender systems. In *Recommender systems handbook*, pages 479–510. Springer, 2011.

[934] Rich Caruana. *Multitask Learning*. 1998.

[935] Bart Bakker and Tom Heskes. Task clustering and gating for bayesian multitask learning. *J. Mach. Learn. Res.*, 4, December 2003.

[936] Shai Ben-David and Reba Schuller. *Exploiting Task Relatedness for Multiple Task Learning*, pages 567–580. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[937] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6, December 2005.

[938] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. *Learning from Incomplete Ratings Using Non-negative Matrix Factorization*.

[939] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug 2009.

[940] Aonghus Lawlor, Khalil Muhammad, Rachael Rafter, and Barry Smyth. *Opinionated Explanations for Recommendation Systems*. Springer International Publishing, Cham, 2015.

[941] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, 2001.

[942] Nava Tintarev and Judith Masthoff. Effective explanations of recommendations: User-centered design. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, New York, NY, USA, 2007. ACM.

[943] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, 2008.

[944] Weiquan Wang and Izak Benbasat. Recommendation agents for electronic commerce: Effects of explanation facilities on trusting beliefs. *J. Manage. Inf. Syst.*, 23(4), May 2007.

[945] Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: Explaining recommendations using tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, 2009.

[946] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, ICDEW '07, 2007.

[947] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD*, KDD '09, 2009.

[948] Machine Learning, Computer Science, and R Caruana. Multitask learning *. *Betascript Publishing*, 1997.

[949] Jonathan Baxter. A model of inductive bias learning. *CoRR*, abs/1106.0245, 2011.

[950] S. Thrun. Is learning the n-th thing any easier than learning the first? In D. Touretzky and M Mozer, editors, *Advances in Neural Information Processing Systems (NIPS) 8*, pages 640–646, Cambridge, MA, 1996. MIT Press.

[951] Kevin Hsin-Yih Lin and Hsin-Hsi Chen. Ranking reader emotions using pairwise loss minimization and emotional distribution regression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, 2008.

[952] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent dirichlet allocation for tag recommendation. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, 2009.

[953] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, 2009.

[954] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, 2010.

[955] Asela Gunawardana and Guy Shani. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10:2935–2962, 2009.

[956] Amr Ahmed and Eric P. Xing. Staying informed: Supervised and semi-supervised multi-view topical analysis of ideological perspective. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, 2010.

[957] Yang Song, Lu Zhang, and C. Lee Giles. Automatic tag recommendation algorithms for social recommender systems. *ACM Trans. Web*, 5(1), February 2011.

[958] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 635–644, 2011.

[959] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th International ACM SIGIR*, SIGIR '11, 2011.

[960] Yunzhi Tan, Yongfeng Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. *A Unified Framework for Emotional Elements Extraction Based on Finite State Matching Machine*. 2013.

[961] Yongfeng Zhang. Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, 2015.

[962] Guang Ling, Michael R. Lyu, and Irwin King. Ratings meet reviews, a combined approach to recommend. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, 2014.

[963] Muthusamy Chelliah and Sudeshna Sarkar. Product recommendations enhanced with reviews. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, 2017.

[964] Xiaowen Ding, Bing Liu, and Philip S. Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, 2008.

[965] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Comput. Linguist.*, 37(2), June 2011.

[966] Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, 2012.

[967] Amit Sharma and Dan Cosley. Do social explanations work?: Studying and modeling the effects of social explanations in recommender systems. In *Proceedings of the 22nd WWW*, WWW '13, 2013.

[968] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, 2013.

[969] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM RecSys*, RecSys '13, 2013.

[970] Yongfeng Zhang, Haochen Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. Do users rate or review?: Boost phrase-level sentiment labeling with review-level sentiment classification. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, 2014.

[971] Xiangnan He, Ming Gao, Min-Yen Kan, Yiqun Liu, and Kazunari Sugiyama. Predicting the popularity of web 2.0 items based on user comments. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, 2014.

[972] Yongfeng Zhang. Browser-oriented universal cross-site recommendation and explanation based on user browsing logs. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, 2014.

[973] Yao Wu and Martin Ester. Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering. In *Proceedings of the Eighth ACM WSDM*, WSDM '15, 2015.

[974] Yongfeng Zhang, Min Zhang, Yi Zhang, Guokun Lai, Yiqun Liu, Honghui Zhang, and Shaoping Ma. Daily-aware personalized recommendation based on feature-level time series analysis. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, 2015.

[975] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, 2015.

[976] Wayne Xin Zhao, Jinpeng Wang, Yulan He, Ji-Rong Wen, Edward Y. Chang, and Xiaoming Li. Mining product adopter information from online reviews for improving product recommendation. *ACM Trans. Knowl. Discov. Data*, 10(3), February 2016.

[977] Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. Learning to rank features for recommendation over multiple categories. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, 2016.

[978] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *20th International Conference on Neural Information Processing Systems*, 2007.

[979] Yong Wu, Yuan Yao, Feng Xu, Hanghang Tong, and Jian Lu. Tag2word: Using tags to generate words for content based tag recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, 2016.

[980] Xiaolin Zheng, Weifeng Ding, Zhen Lin, and Chaochao Chen. Topic tensor factorization for recommender system. *Inf. Sci.*, 372(C), December 2016.

[981] Qi Zhao, Yongfeng Zhang, Yi Zhang, and Daniel Friedman. Multi-product utility maximization for economic recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, 2017.

[982] Yunzhi Tan, Min Zhang, Yiqun Liu, and Shaoping Ma. Rating-boosted latent topics: Understanding users and items with ratings and reviews. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, 2016.

[983] Chunfeng Yang, Huan Yan, Donghan Yu, Yong Li, and Dah Ming Chiu. Multi-site user behavior modeling and its application in video recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, 2017.

[984] Fan Zhang, Yiqun Liu, Xin Li, Min Zhang, Yinghui Xu, and Shaoping Ma. Evaluating web search with a bejeweled player model. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, 2017.

[985] Behnoush Abdollahi and Olfa Nasraoui. Using explainability for constrained matrix factorization. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, 2017.

[986] Michal Rosen-Zvi, Thomas L. Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. *CoRR*, abs/1207.4169, 2012.

[987] Yongfeng Zhang. Explainable recommendation: Theory and applications. *CoRR*, abs/1708.06409, 2017.

[988] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 1990.

[989] Yongfeng Zhang. Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation. pages 435–439, 02 2015.

[990] Shyam Upadhyay, Ming-Wei Chang, Kai-Wei Chang, and Wen tau Yih. Learning from explicit and implicit supervision jointly for algebra word problems. In *EMNLP*, 2016.

[991] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora.

[992] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. Fake review detection: Classification and analysis of real and pseudo reviews. 2013.

[993] Alexander Eisl, Hermann W. Elendner, and Manuel Lingo. Re-Mapping Credit Ratings. Technical report, 08 2015.

[994] Renqin Cai, Chi Wang, and Hongning Wang. Accounting for the correspondence in commented data. In *The 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 2017.

[995] Xu Chen, Pengfei Wang, Zheng Qin, and Yongfeng Zhang. Hlbpr: A hybrid local bayesian personal ranking method. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, 2016.

[996] Mustafa Bilgic and Raymond Mooney. Explaining recommendations: Satisfaction vs. promotion. In *Proceedings of Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research at the 2005 International Conference on Intelligent User Interfaces*, 2005.

[997] Khairullah Khan, Baharum Baharudin, Aurnagzeb Khan, and Ashraf Ullah. Mining opinion components from unstructured reviews: A review. *Journal of King Saud University - Computer and Information Sciences*, 26(3):258–275, 2014.

[998] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[999] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[1000] Igal Hendel and Aviv Nevo. Measuring the implications of sales and consumer inventory behavior. *Econometrica*, 74(6):1637–1673, 2006.

[1001] Gautam Gowrisankaran and Marc Rysman. Dynamics of consumer demand for new durable goods. *Journal of political Economy*, 120(6):1173–1219, 2012.

[1002] Jean-Pierre Dubé, Jeremy T Fox, and Che-Lin Su. Improving the numerical performance of static and dynamic aggregate discrete choice random coefficients demand estimation. *Econometrica*, 80(5):2231–2267, 2012.

[1003] Alan T Sorensen. An empirical model of heterogeneous consumer search for retail prescription drugs. Technical report, National Bureau of Economic Research, 2001.

[1004] Bing Pan, Helene Hembrooke, Thorsten Joachims, Lori Lorigo, Geri Gay, and Laura Granka. In google we trust: Users' decisions on rank, position, and relevance. *Journal of computer-mediated communication*, 12(3):801–823, 2007.

[1005] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[1006] Jens Hainmueller. Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies. *Political analysis*, pages 25–46, 2012.

[1007] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 31–39. ACM, 2018.

[1008] Rudolf Beran and Peter Hall. Estimating coefficient distributions in random coefficient regressions. *The annals of Statistics*, 20(4):1970–1984, 1992.

[1009] Xueying Bai, Jian Guan, and Hongning Wang. A model-based reinforcement learning with adversarial training for online recommendation. In *Advances in Neural Information Processing Systems*, 2019.

[1010] Md Mustafizur Rahman and Hongning Wang. Hidden topic sentiment model. In *Proceedings of the 25th International Conference on World Wide Web*, pages 155–165. International World Wide Web Conferences Steering Committee, 2016.

[1011] T.L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

[1012] P. Auer, N. Cesa-Bianchi, Y. Freund, and Robert E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 322–331, Oct 1995.

[1013] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM J. Comput.*, 32(1):48–77, January 2003.

[1014] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, May 2002.

[1015] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *19th International Conference on World Wide Web*, WWW '10, pages 661–670, New York, NY, USA, 2010. ACM.

[1016] John C Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 148–177, 1979.

[1017] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[1018] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

[1019] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 297–306, New York, NY, USA, 2011. ACM.

[1020] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 817–824. Curran Associates, Inc., 2008.

[1021] S. Kar, H.V. Poor, and Shuguang Cui. Bandit problems in networks: Asymptotically efficient distributed allocation rules. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 1771–1778, Dec 2011.

[1022] Swapna Buccapatnam, Atilla Eryilmaz, and Ness B. Shroff. Stochastic bandits with side observations on networks. *SIGMETRICS Perform. Eval. Rev.*, 42(1):289–300, June 2014.

[1023] Swapna Buccapatnam, Atilla Eryilmaz, and Ness B Shroff. Multi-armed bandits in the presence of side observations in social networks. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 7309–7314. IEEE, 2013.

[1024] Aleksandrs Slivkins. Contextual bandits with similarity information. *The Journal of Machine Learning Research*, 15(1):2533–2568, 2014.

[1025] Kareem Amin, Michael Kearns, and Umar Syed. Graphical models for bandit problems. *Twenty-Seventh Conference Uncertainty in Artificial Intelligence (UAI 2011)*, 2011.

[1026] Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *31st International Conference on Machine Learning (ICML-14)*, pages 757–765, 2014.

[1027] Shuai Li, Wei Chen, and Kwong-Sak Leung. Improved algorithm on online clustering of bandits. *arXiv preprint arXiv:1902.09162*, 2019.

[1028] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10:3152676, 2017.

[1029] Abhimanyu Dubey and Alex Pentland. Differentially-private federated linear bandits. *arXiv preprint arXiv:2010.11425*, 2020.

[1030] Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. Distributed bandit learning: Near-optimal regret with efficient communication. *arXiv preprint arXiv:1904.06309*, 2019.

[1031] Nathan Korda, Balazs Szorenyi, and Shuai Li. Distributed clustering of linear bandits in peer to peer networks. In *International conference on machine learning*, pages 1301–1309. PMLR, 2016.

[1032] T-H Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):1–24, 2011.

[1033] Ido Guy, Naama Zwerdling, David Carmel, Inbal Ronen, Erel Uziel, Sivan Yogev, and Shila Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *Third ACM Conference on Recommender Systems*, RecSys '09, pages 53–60, New York, NY, USA, 2009. ACM.

[1034] Stephen M. Stigler. The asymptotic distribution of the trimmed mean. *Ann. Statist.*, 1(3):472–477, 05 1973.

[1035] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, New York, NY, USA, 1994. ACM.

[1036] Eshcar Hillel, Zohar S Karnin, Tomer Koren, Ronny Lempel, and Oren Somekh. Distributed exploration in multi-armed bandits. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and

K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 854–862. Curran Associates, Inc., 2013.

[1037] Baruch Awerbuch and Robert Kleinberg. Competitive collaborative learning. *J. Comput. Syst. Sci.*, 74(8):1271–1288, December 2008.

[1038] J. Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *1st ACM Conference on Electronic Commerce*, EC '99, pages 158–166, New York, NY, USA, 1999. ACM.

[1039] Benjamin Marlin and Richard S. Zemel. The multiple multiplicative factor model for collaborative filtering. In *Twenty-first International Conference on Machine Learning*, ICML '04, pages 73–, New York, NY, USA, 2004. ACM.

[1040] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. Technical Report MSR-TR-98-12, Microsoft Research, May 1998.

[1041] Robert B Cialdini and Melanie R Trost. *Social influence: Social norms, conformity and compliance.* McGraw-Hill, 1998.

[1042] Thomas J Walsh, István Szita, Carlos Diuk, and Michael L Littman. Exploring compact reinforcement-learning representations with linear regression. In *Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 591–598. AUAI Press, 2009.

[1043] Paul M Weichsel. The kronecker product of graphs. *Proceedings of the American Mathematical Society*, 13(1):47–52, 1962.

[1044] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Fourth International Conference on Web Search and Web Data Mining (WSDM-11)*, pages 297–306, 2011.

[1045] Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2312–2320. Curran Associates, Inc., 2011.

[1046] Wei Li, Xuerui Wang, Ruofei Zhang, Ying Cui, Jianchang Mao, and Rong Jin. Exploitation and exploration in a performance based contextual advertising system. In *16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 27–36. ACM, 2010.

[1047] Djallel Bouneffouf, Amel Bouzeghoub, and Alda Lopes Gançarski. A contextual-bandit algorithm for mobile context-aware recommender system. In *Neural Information Processing*, pages 324–331. Springer, 2012.

[1048] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944–1957, November 2007.

[1049] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE, 2008.

[1050] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

[1051] Zan Huang, Hsinchun Chen, and Daniel Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):116–142, 2004.

[1052] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H. Byers. Big data: The next frontier for innovation, competition, and productivity, May 2011.

[1053] Viktor Mayer-Schönberger and Kenneth Cukier. *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.

[1054] Andrew McAfee, Erik Brynjolfsson, Thomas H Davenport, DJ Patil, and Dominic Barton. Big data. *The management revolution. Harvard Bus Rev*, 90(10):61–67, 2012.

[1055] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[1056] Divyakant Agrawal, Sudipto Das, and Amr El Abbadi. Big data and cloud computing: current state and future opportunities. In *14th International Conference on Extending Database Technology*, pages 530–533. ACM, 2011.

[1057] Eric E Schadt, Michael D Linderman, Jon Sorenson, Lawrence Lee, and Garry P Nolan. Computational solutions to large-scale data management and analysis. *Nature Reviews Genetics*, 11(9):647–657, 2010.

[1058] Answers. `http://wiki.answers.com/Q/How_many_transactions_does_eBay_process_per_day?#slide2`, 2013.

[1059] Internet Live Stats. Twitter usage statistics. `http://www.internetlivestats.com/twitter-statistics/`, 2015.

[1060] Statistic Brain. Google annual search statistics. `http://www.statisticbrain.com/google-searches`, 2015.

[1061] Jianqing Fan, Fang Han, and Han Liu. Challenges of big data analysis. *National science review*, 1(2):293–314, 2014.

[1062] Jeff Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[1063] Daren C Brabham. Crowdsourcing as a model for problem solving an introduction and cases. *Convergence: the international journal of research into new media technologies*, 14(1):75–90, 2008.

[1064] Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. Data quality from crowdsourcing: a study of annotation selection criteria. In *NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35. Association for Computational Linguistics, 2009.

[1065] Peter Welinder and Pietro Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 25–32. IEEE, 2010.

[1066] Jennie Si and Yu-Tsung Wang. Online learning control by association and reinforcement. *Neural Networks, IEEE Transactions on*, 12(2):264–276, 2001.

[1067] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[1068] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.

[1069] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[1070] Gerhard Fischer. User modeling in human–computer interaction. *User modeling and user-adapted interaction*, 11(1-2):65–86, 2001.

[1071] Daniel Billsus and Michael J Pazzani. User modeling for adaptive news access. *User modeling and user-adapted interaction*, 10(2-3):147–180, 2000.

[1072] Cody Kwok, Oren Etzioni, and Daniel S Weld. Scaling question answering to the web. *ACM Transactions on Information Systems (TOIS)*, 19(3):242–262, 2001.

[1073] Pawel Jurczyk and Eugene Agichtein. Discovering authorities in question answer communities by using link analysis. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 919–922. ACM, 2007.

[1074] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. Discovery and evaluation of aggregate usage profiles for web personalization. *Data mining and knowledge discovery*, 6(1):61–82, 2002.

[1075] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[1076] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pages 1–9. IEEE, 2010.

[1077] Cem Tekin and Mingyan Liu. Online learning in opportunistic spectrum access: A restless bandit approach. In *INFOCOM, 2011 Proceedings IEEE*, pages 2462–2470. IEEE, 2011.

[1078] Debadeepta Dey, Tian Yu Liu, Martial Hebert, and J Andrew Bagnell. Contextual sequence prediction with application to control library optimization. *Robotics*, page 49, 2013.

[1079] Ari Weinstein and Michael L Littman. Bandit-based planning and learning in continuous-action markov decision processes. In *ICAPS*, 2012.

[1080] Ruth Urner, Shai Ben-David, and Ohad Shamir. Learning from weak teachers. In *International Conference on Artificial Intelligence and Statistics*, pages 1252–1260, 2012.

[1081] Luigi Malago, Nicolo Cesa-Bianchi, and Jean-Michel Renders. Online active learning with strong and weak annotators.

[1082] Ofer Dekel, Claudio Gentile, and Karthik Sridharan. Selective sampling and active learning from single and multiple teachers. *The Journal of Machine Learning Research*, 13(1):2655–2697, 2012.

[1083] Yan Yan, Glenn M Fung, Rómer Rosales, and Jennifer G Dy. Active learning from crowds. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1161–1168, 2011.

[1084] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[1085] Nicolo Cesa-Bianchi, Claudio Gentile, and Francesco Orabona. Robust bounds for classification via selective sampling. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 121–128. ACM, 2009.

[1086] Francesco Orabona and Nicolo Cesa-Bianchi. Better algorithms for selective sampling. In *Proceedings of the 28th international conference on Machine learning (ICML-11)*, pages 433–440, 2011.

[1087] Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[1088] David A. Cohn, Les E. Atlas, and Richard E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[1089] Pinar Donmez and Jaime G Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 619–628. ACM, 2008.

[1090] Yan Yan, Rómer Rosales, Glenn Fung, Faisal Farooq, Bharat Rao, and Jennifer G Dy. Active learning from multiple knowledge sources. In *International Conference on Artificial Intelligence and Statistics*, pages 1350–1357, 2012.

[1091] Chicheng Zhang and Kamalika Chaudhuri. Active learning from weak and strong labelers. *arXiv preprint arXiv:1510.02847*, 2015.

[1092] Aleksandrs Slivkins and Eli Upfal. Adapting to a changing environment: the brownian restless bandits. In *COLT*, pages 343–354, 2008.

[1093] Hamid Nazerzadeh, Amin Saberi, and Rakesh Vohra. Dynamic cost-per-action mechanisms and applications to online advertising. In *Proceedings of the 17th international conference on World Wide Web*, pages 179–188. ACM, 2008.

[1094] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, pages 282–293. Springer, 2006.

[1095] Vikram Krishnamurthy and Robin J Evans. Hidden markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking. *Signal Processing, IEEE Transactions on*, 49(12):2893–2908, 2001.

[1096] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133–168, 1997.

[1097] Pinar Donmez, Jaime G Carbonell, and Jeff Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 259–268. ACM, 2009.

[1098] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *Proc. OSDI*, pages 583–598, 2014.

[1099] Mu Li, Li Zhou, Zichao Yang, Aaron Li, Fei Xia, David G Andersen, and Alexander Smola. Parameter server for distributed machine learning. In *Big Learning NIPS Workshop*, 2013.

[1100] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, pages 521–530. ACM, 2007.

[1101] P Bryan Heidorn. Shedding light on the dark data in the long tail of science. *Library Trends*, 57(2):280–299, 2008.

[1102] Yoon-Joo Park and Alexander Tuzhilin. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 11–18. ACM, 2008.

[1103] Quanquan Gu and Rong Jin. Stochastic optimization of partially strongly convex functions. 2013.

[1104] Quanquan Gu, Tong Zhang, and Jiawei Han. Batch-mode active learning via error bound minimization. 2014.

[1105] Quanquan Gu, Charu C. Aggarwal, and Jiawei Han. Unsupervised link selection in networks. In *AISTATS*, 2013.

[1106] Quanquan Gu and Jiawei Han. Clustered support vector machines. In *AISTATS*, 2013.

[1107] Quanquan Gu and Jiawei Han. Online spectral learning on a graph with bandit feedback. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 833–838. IEEE, 2014.

[1108] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning*, pages 151–159, 2013.

[1109] Dezhi Hong, Hongning Wang, and Kamin Whitehouse. Clustering-based active learning on sensor type classification in buildings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 363–372. ACM, 2015.

[1110] Jacques Bughin, Laura Corb, James Manyika, Olivia Nottebohm, Michael Chui, Borja de Muller Barbat, and Remi Said. The impact of internet technologies: Search. *High Tech Practice. McKinsey&Company*, 2011.

[1111] Gerard Salton and Michael J McGill. *Introduction to modern information retrieval*. mcgraw-hill, 1983.

[1112] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. Introduction to information retrieval. In *Proceedings of the international communication of association for computing machinery conference*, page 260, 2008.

[1113] Hemant Jain, Cheng Thao, and Huimin Zhao. Enhancing electronic medical record retrieval through semantic query expansion. *Information systems and e-business management*, 10(2):165–181, 2012.

[1114] Henning Müller, Jayashree Kalpathy-Cramer, Ivan Eggel, Steven Bedrick, Saïd Radhouani, Brian Bakke, Charles E Kahn, and William Hersh. Overview of the clef 2009 medical image retrieval track. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 72–84. Springer, 2009.

[1115] Howard D White. Scientific communication and literature retrieval. *The handbook of research synthesis and meta-analysis*, 2:51–71, 2009.

[1116] Steve Lawrence, Kurt Bollacker, and C Lee Giles. Indexing and retrieval of scientific literature. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 139–146. ACM, 1999.

[1117] Bruce R Schatz. Information retrieval in digital libraries: Bringing search to the net. *Science*, 275(5298):327–334, 1997.

[1118] David Hawking. Challenges in enterprise search. In *Proceedings of the 15th Australasian database conference-Volume 27*, pages 15–24. Australian Computer Society, Inc., 2004.

[1119] Rajat Mukherjee and Jianchang Mao. Enterprise search: Tough stuff. *Queue*, 2(2):36, 2004.

[1120] Mark Sanderson and W Bruce Croft. The history of information retrieval research. *Proceedings of the IEEE*, 100(Special Centennial Issue):1444–1451, 2012.

[1121] Venkat N Gudivada and Vijay V Raghavan. Content based image retrieval systems. *Computer*, 28(9):18–22, 1995.

[1122] Nicola Orio. Music retrieval: A tutorial and review. *Foundations and Trends® in Information Retrieval*, 1(1):1–90, 2006.

[1123] Sara Cohen, Carmel Domshlak, and Naama Zwerdling. On ranking techniques for desktop search. *ACM Transactions on Information Systems (TOIS)*, 26(2):11, 2008.

[1124] Hang Li. Learning to rank for information retrieval and natural language processing. *Synthesis Lectures on Human Language Technologies*, 4(1):1–113, 2011.

[1125] Jun Xu and Hang Li. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 391–398. ACM, 2007.

[1126] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 55–64. ACM, 2017.

[1127] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74. ACM, 2017.

[1128] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278. ACM, 2007.

[1129] Yisong Yue and Thorsten Joachims. Beat the mean bandit. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 241–248, 2011.

[1130] Artem Grotov and Maarten de Rijke. Online learning to rank for information retrieval: Sigir 2016 tutorial. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1215–1218. ACM, 2016.

[1131] Dawei Yin, Yuening Hu, Jiliang Tang, Tim Daly, Mianwei Zhou, Hua Ouyang, Jianhui Chen, Changsung Kang, Hongbo Deng, and Chikashi Nobata. Ranking relevance in yahoo search. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–332. ACM, 2016.

[1132] Alexandros Karatzoglou, Linas Baltrunas, and Yue Shi. Learning to rank for recommender systems. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 493–494. ACM, 2013.

[1133] Weiguo Fan, Michael Gordon, and Praveen Pathak. Personalization of search engine services for effective retrieval and knowledge management. *ICIS 2000 Proceedings*, page 4, 2000.

[1134] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[1135] Grace Hui Yang and Sicong Zhang. Differential privacy for information retrieval. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 325–326. ACM, 2017.

[1136] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1, 2012.

[1137] Branislav Kveton, Manzil Zaheer, Csaba Szepesvari, Lihong Li, Mohammad Ghavamzadeh, and Craig Boutilier. Randomized exploration in generalized linear bandits. *arXiv preprint arXiv:1906.08947*, 2019.

[1138] Branislav Kveton, Csaba Szepesvari, Mohammad Ghavamzadeh, and Craig Boutilier. Perturbed-history exploration in stochastic linear bandits. *arXiv preprint arXiv:1903.09132*, 2019.

[1139] Muzafer Sherif. *The psychology of social norms.* Harper, 1936.

[1140] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[1141] Bhaskar Mitra, Eric Nalisnick, Nick Craswell, and Rich Caruana. A dual embedding space model for document ranking. *arXiv preprint arXiv:1602.01137*, 2016.

[1142] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1291–1299. International World Wide Web Conferences Steering Committee, 2017.

[1143] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 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*, pages 610–618. ACM, 2018.

[1144] Abhradeep Guha Thakurta and Adam Smith. (nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems*, pages 2733–2741, 2013.

[1145] Naman Agarwal and Karan Singh. The price of differential privacy for online learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 32–40. JMLR. org, 2017.

[1146] Jacob Abernethy, Chansoo Lee, Audra McMillan, and Ambuj Tewari. Online learning via differential privacy. *arXiv preprint arXiv:1711.10019*, 2017.

[1147] Nikita Mishra and Abhradeep Thakurta. (nearly) optimal differentially private stochastic multi-arm bandits. In *Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 592–601. AUAI Press, 2015.

[1148] Aristide Charles Yedia Tossou and Christos Dimitrakakis. Achieving privacy in the adversarial multi-armed bandit. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[1149] Roshan Shariff and Or Sheffet. Differentially private contextual linear bandits. In *Advances in Neural Information Processing Systems*, pages 4296–4306, 2018.

[1150] Seth Neel and Aaron Roth. Mitigating bias in adaptive data gathering via differential privacy. *arXiv preprint arXiv:1806.02329*, 2018.

[1151] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724. ACM, 2010.

[1152] Yisong Yue, Yue Gao, Oliver Chapelle, Ya Zhang, and Thorsten Joachims. Learning more powerful test statistics for click-based retrieval evaluation. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 507–514. ACM, 2010.

[1153] Donald Metzler, Susan Dumais, and Christopher Meek. Similarity measures for short segments of text. In *European conference on information retrieval*, pages 16–27. Springer, 2007.

[1154] Venu Satuluri and Srinivasan Parthasarathy. Bayesian locality sensitive hashing for fast similarity search. *Proceedings of the VLDB Endowment*, 5(5):430–441, 2012.

[1155] Benjamin Van Durme and Ashwin Lall. Online generation of locality sensitive hash signatures. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 231–235. Association for Computational Linguistics, 2010.

[1156] Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[1157] Shing-Tung Yau. Some function-theoretic properties of complete riemannian manifold and their applications to geometry. *Indiana University Mathematics Journal*, 25(7):659–670, 1976.

[1158] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[1159] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[1160] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[1161] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. *arXiv preprint arXiv:1807.00459*, 2018.

[1162] Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. On context-dependent clustering of bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1253–1262. JMLR. org, 2017.

[1163] David Cossock and Tong Zhang. Subset ranking using regression. In *International Conference on Computational Learning Theory*, pages 605–619. Springer, 2006.

[1164] Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 525–533, 2011.

[1165] Joseph A Calandrino, Ann Kilzer, Arvind Narayanan, Edward W Felten, and Vitaly Shmatikov. " you might also like:" privacy risks of collaborative filtering. In *2011 IEEE Symposium on Security and Privacy*, pages 231–246. IEEE, 2011.

[1166] Aleksandra Korolova. Privacy violations using microtargeted ads: A case study. In *2010 IEEE International Conference on Data Mining Workshops*, pages 474–482. IEEE, 2010.

[1167] Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1054–1062, 2016.

[1168] Aristide CY Tossou and Christos Dimitrakakis. Algorithms for differentially private multi-armed bandits. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[1169] Alina Beygelzimer and John Langford. The offset tree for learning with partial labels. In *15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 129–138. ACM, 2009.

[1170] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[1171] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[1172] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013.

[1173] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018.

[1174] Pavel Laskov. Practical evasion of a learning-based classifier: A case study. In *2014 IEEE symposium on security and privacy*, pages 197–211. IEEE, 2014.

[1175] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24–1, 2012.

[1176] Florian Mueller and Andrea Lockerd. Cheese: tracking mouse movement activity on websites, a tool for user modeling. In *CHI'01 extended abstracts on Human factors in computing systems*, pages 279–280. ACM, 2001.

[1177] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[1178] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[1179] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

[1180] Katja Hofmann, Shimon Whiteson, and Maarten De Rijke. A probabilistic method for inferring preferences from clicks. In *20th ACM international conference on Information and knowledge management*, pages 249–258. ACM, 2011.

[1181] Eugene Kharitonov. Federated online learning to rank with evolution strategies. In *WSDM*, 2019.

[1182] Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval*, volume 310, 2007.

[1183] Krishnaram Kenthapadi, Benjamin Le, and Ganesh Venkataraman. Personalized job recommendation system at linkedin: Practical challenges and lessons learned. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 346–347, 2017.

[1184] Aditya Parameswaran, Petros Venetis, and Hector Garcia-Molina. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems (TOIS)*, 29(4):1–33, 2011.

[1185] Xia Jing and Jie Tang. Guess you like: course recommendation in moocs. In *Proceedings of the International Conference on Web Intelligence*, pages 783–789, 2017.

[1186] Tao Yang and Qingyao Ai. Maximizing marginal fairness for dynamic learning to rank. *arXiv preprint arXiv:2102.09670*, 2021.

[1187] Piotr Sapiezynski, Wesley Zeng, Ronald E Robertson, Alan Mislove, and Christo Wilson. Quantifying the impact of user attentionon fair group representation in ranked lists. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 553–562, 2019.

[1188] Bhaskar Mitra and Nick Craswell. Neural models for information retrieval. *arXiv preprint arXiv:1705.01509*, 2017.

[1189] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 55–64, 2016.

[1190] Kuang-Hua Chang. Chapter 19 - multiobjective optimization and advanced topics. In Kuang-Hua Chang, editor, *e-Design*, pages 1105 – 1173. Academic Press, Boston, 2015.

[1191] Ralph L. Keeney and Howard Raiffa. *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. Wiley, New York, 1976.

[1192] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[1193] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, 2019.

[1194] Qingyao Ai, Tao Yang, Huazheng Wang, and Jiaxin Mao. Unbiased learning to rank: Online or offline? *ACM Trans. Inf. Syst.*, 39(2), February.

[1195] N. Meinshausen and B. Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, 37(1):246–270, 2009.

[1196] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

[1197] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[1198] Filip Radlinski and Nick Craswell. Comparing the sensitivity of information retrieval metrics. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 667–674, 2010.

[1199] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

[1200] Michinari Momma, Alireza Bagheri Garakani, Nanxun Ma, and Yi Sun. Multi-objective ranking via constrained optimization, 2020.

[1201] David Carmel, Elad Haramaty, Arnon Lazerson, and Liane Lewin-Eytan. Multi-objective ranking optimization for product search using stochastic label aggregation. In *Proceedings of The Web Conference 2020*, WWW '20, page 373–383, New York, NY, USA, 2020. Association for Computing Machinery.

[1202] Miettinen Kaisa. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, Boston, USA, 1999.

[1203] Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Słowiński, editors. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer-Verlag, Berlin, Heidelberg, 2008.

[1204] Matthew J Salganik, Peter Sheridan Dodds, and Duncan J Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *science*, 311(5762):854–856, 2006.

[1205] Lada A Adamic, Bernardo A Huberman, AL Barabási, R Albert, H Jeong, and G Bianconi. Power-law distribution of the world wide web. *science*, 287(5461):2115–2115, 2000.

[1206] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs, 2014.

[1207] Marc Abeille and Alessandro Lazaric. Linear thompson sampling revisited. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 176–184, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

[1208] Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *COLT*, pages 41–53, 2010.

[1209] Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *NIPS-Twenty-Sixth Annual Conference on Neural Information Processing Systems*, 2012.

[1210] Ashley F Emery and Aleksey V Nenarokomov. Optimal experiment design. *Measurement Science and Technology*, 9(6):864, 1998.

[1211] Cristian R Rojas, James S Welsh, Graham C Goodwin, and Arie Feuer. Robust optimal experiment design for system identification. *Automatica*, 43(6):993–1008, 2007.

[1212] Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *Artificial Intelligence and Statistics*, pages 1–9. PMLR, 2012.

[1213] Davis Gilton and Rebecca Willett. Sparse linear contextual bandits via relevance vector machines. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, pages 518–522. IEEE, 2017.

[1214] Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems 27*, pages 828–836, 2014.

[1215] Yassir Jedra and Alexandre Proutiere. Optimal best-arm identification in linear bandits, 2020.

[1216] Matthew Hoffman, Bobak Shahriari, and Nando Freitas. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 365–374, Reykjavik, Iceland, 22–25 Apr 2014. PMLR.

[1217] Ben Carterette. An analysis of np-completeness in novelty and diversity ranking. *Information Retrieval*, 14(1):89–106, 2011.

[1218] Yao Wu, Jian Cao, Guandong Xu, and Yudong Tan. Tfrom: A two-sided fairness-aware recommendation model for both customers and providers. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1013–1022, 2021.

[1219] Haolun Wu, Chen Ma, Bhaskar Mitra, Fernando Diaz, and Xue Liu. Multi-fr: A multi-objective optimization method for achieving two-sided fairness in e-commerce recommendation. *arXiv preprint arXiv:2105.02951*, 2021.

[1220] Lequn Wang and Thorsten Joachims. User fairness, item fairness, and diversity for rankings in two-sided markets. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 23–41, 2021.

[1221] Gourab K Patro, Abhijnan Chakraborty, Niloy Ganguly, and Krishna Gummadi. Incremental fairness in two-sided market platforms: On smoothly updating recommendations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 181–188, 2020.

[1222] Abhijnan Chakraborty, Aniko Hannak, Asia J Biega, and Krishna Gummadi. Fair sharing for sharing economy platforms. In *Fairness, Accountability and Transparency in Recommender Systems-Workshop on Responsible Recommendation*, 2017.

[1223] Himan Abdollahpouri and Robin Burke. Multi-stakeholder recommendation and its connection to multi-sided fairness. *arXiv preprint arXiv:1907.13158*, 2019.

[1224] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338, 2013.

[1225] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W Bruce Croft, and Xueqi Cheng. A deep look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6):102067, 2020.

[1226] Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 373–382, 2015.

Appendix

A Background and Notations

Table 1: Notations used in this paper.

Notation	Description
$\mathbf{x}_i^t$, $\mathbf{x}_j^t$	feature vector of document i and j under query q_t at round t.
m	the width of a DNN in each layer.
L	the number of layers of a DNN.
p	the total number of parameters in a DNN.
$h(\cdot), \mathbf{h}$	the underlying optimal ranking score function.
$\boldsymbol{\gamma}^*$	optimal model for the underlying scoring function, $\boldsymbol{\gamma}^* = \boldsymbol{\theta}^* - \boldsymbol{\theta}_0$.
$\widehat{\boldsymbol{\gamma}}_t$	solution of the cross-entropy loss with the linearized neural network at round t.
$\mathbf{H}$	the neural tangent kernel matrix for all possible query-document features.
λ_0	$\mathbf{H} \succ \lambda_0 \mathbf{I}$ minimum eigen-value of $\mathbf{H}$
$V_t, V_{\max}$	the number of candidate documents at round t, and the maximum V_t across all queries.
Ψ_t	the set of all possible document pairs at round t, e.g., $\Psi_t = \{(i,j) \in \Psi_t, i \neq j\}$
ω_t	the set of certain rank orders at round t.
n_t	the total number of query-document feature vectors until round t, which satisfies $n_t = \sum_{s=1}^{t} V_s \leq t V_{\max}$.
n_t^P	the total number of training document pairs. As we only show top-K documents to the users, n_t^P satisfies that $n_t^P \leq \lceil \frac{tK}{2} \rceil$
$f(\mathbf{x}_i; \boldsymbol{\theta}_t)$	estimated ranking score of document $\mathbf{x}_i$ at round t
f_{ij}^t	the difference between the estimated ranking scores, $f_{ij}^t = f(\mathbf{x}_i^t; \boldsymbol{\theta}_{t-1}) - f(\mathbf{x}_j^t; \boldsymbol{\theta}_{t-1})$.
$\mathbf{g}(\mathbf{x}_i; \boldsymbol{\theta}_t)$	the gradient of the neural network function at time t, $\mathbf{g}(\mathbf{x}; \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta}) \in \mathbb{R}^p$
$\mathbf{g}_{ij}^t$	the difference between the gradients at time t, $\mathbf{g}_{ij}^t = \mathbf{g}(\mathbf{x}_i^t; \boldsymbol{\theta}_t) - \mathbf{g}(\mathbf{x}_j^t; \boldsymbol{\theta}_t)$.
$\mathbf{g}_{ij}^{t,0}$	the difference between the gradients at time 0, $\mathbf{g}_{ij}^{t,0} = \mathbf{g}(\mathbf{x}_i^t; \boldsymbol{\theta}_0) - \mathbf{g}(\mathbf{x}_j^t; \boldsymbol{\theta}_0)$.
η	step size for gradient descent in neural network optimization.
J	the number of gradient descent steps.
ξ	pairwise noise in the click feedback.
ν	sub-Gaussian variable for the pairwise noise ξ.
S	norm parameter for neural tangent kernel.
λ	regularization parameter for loss function.
$\boldsymbol{\zeta}$	pseudo noise for perturbation
υ	variance of the pseudo noise
$\mathbf{A}_t^{lin}$	$\mathbf{A}_t^{lin} = \sum_{s=1}^{t-1} \sum_{(i,j) \in \Omega_s^{ind}} \mathbf{x}_{ij}^s \mathbf{x}_{ij}^{s\top} + \lambda \mathbf{I}$
$\mathbf{A}_t^{neu}$	$\mathbf{A}_t^{neu} = \sum_{s=1}^{t-1} \sum_{(i,j) \in \Omega_s^{ind}} \mathbf{g}_{ij}^s \mathbf{g}_{ij}^{s\top} / m + \lambda \mathbf{I}$.
$\bar{\mathbf{A}}_t^{neu}$	$\bar{\mathbf{A}}_t^{neu} = \sum_{s=1}^{t-1} \sum_{(i,j) \in \Omega_s^{ind}} \mathbf{g}_{ij}^{s,0} \mathbf{g}_{ij}^{s,0\top} / m + \lambda \mathbf{I}$.
$\Delta_{\min}$	$\Delta_{\min} = \min_{t \in T, (i,j) \in \Psi_t} \lvert \sigma(h_{ij}) - 1/2 \rvert$.

Before we provide the detailed proofs, we first assume that there are n possible documents to be evaluated during the model learning. It is easy to conclude that $n \leq TV_{\max}$.

First, we introduce the neural tangent kernel matrix defined on the n possible query-document feature vectors across T rounds, $\{\mathbf{x}_i\}_{i=1}^n$.

Definition A.1 ([59,61]). Let $\{\mathbf{x}_i\}_{i=1}^{n_T}$ be the set of all pairwise document feature vectors. Define

$$\begin{aligned}
\widetilde{\mathbf{H}}_{i,j}^{(1)} &= \mathbf{\Sigma}_{i,j}^{(1)} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \qquad \mathbf{B}_{i,j}^{(l)} = \begin{pmatrix} \mathbf{\Sigma}_{i,i}^{(l)} & \mathbf{\Sigma}_{i,j}^{(l)} \\ \mathbf{\Sigma}_{i,j}^{(l)} & \mathbf{\Sigma}_{j,j}^{(l)} \end{pmatrix}, \\
\mathbf{\Sigma}_{i,j}^{(l+1)} &= 2\mathbb{E}_{(u,v)\sim N(\mathbf{0},\mathbf{B}_{i,j}^{(l)})} \left[\phi(u)\phi(v)\right], \\
\widetilde{\mathbf{H}}_{i,j}^{(l+1)} &= 2\widetilde{\mathbf{H}}_{i,j}^{(l)} \mathbb{E}_{(u,v)\sim N(\mathbf{0},\mathbf{B}_{i,j}^{(l)})} \left[\dot{\phi}(u)\dot{\phi}(v)\right] + \mathbf{\Sigma}_{i,j}^{(l+1)}.
\end{aligned}$$

Then, $\mathbf{H} = (\widetilde{\mathbf{H}}^{(L)} + \mathbf{\Sigma}^{(L)})/2$ is called the *neural tangent kernel (NTK)* matrix on the context set $\{x_i\}_{i=1}^{n_T}$, where $n_T = \sum_{s=1}^{T} V_s \leq TV_{\max}$.

We also need the following assumption on the NTK matrix and the corresponding feature set.

Assumption A.2. $\mathbf{H} \succeq \lambda_0 \mathbf{I}$; moreover, for any $1 \leq i \leq n$, $\|\mathbf{x}_i\|_2 = 1$ and $[\mathbf{x}_i]_j = [\mathbf{x}_i]_{j+d/2}$.

With this assumption, the NTK matrix is assumed to be non-singular , which is mild and commonly made in literature [61, 64, 362]. As the query-document features are manually crafted ranking features, it can be easily satisfied when no two feature vectors are in parallel. The second assumption is for convenience in analysis and can be easily satisfied by: for any context $\mathbf{x}$, $\|\mathbf{x}\|_2 = 1$, we can construct a new context $\mathbf{x}' = [\mathbf{x}^\top, \mathbf{x}^\top]^\top/\sqrt{2}$. Equipped with this assumption, it can be verified that with $\boldsymbol{\theta}_0$ initialized as in Algorithm 1, $f(\mathbf{x}_i; \boldsymbol{\theta}_0) = 0$ for any $i \in [n_T]$.

For the sigmoid function σ applied for estimating the pairwise probability, it is well known that σ is continuously differentiable, Lipschitz with constant $k_\mu = 1/4$ and $c_\mu = \inf \dot{\sigma} > 0$.

B Proofs of Theorems and Lemma in Section 2.1

B.1 Proof of Lemma 2.1.3

The proof starts with the following technique lemmas.

Lemma B.1. For the logistic function $\sigma(x) = 1/(1 + \exp(-x))$, we have: $|\sigma(x) - \sigma(y)| < k_\mu |x - y|$ where $k_\mu = 1/4$.

Lemma B.2 (Self-normalized bound for martingales, Lemma 9 in [14]). Let t be a stopping time with respect to the filtration $\{F_\tau\}_{\tau=0}^{\infty}$. Because ξ_{ij}^s is $\frac{1}{2}$-sub-Gaussian, for $\delta > 0$, with probability $1 - \delta$,

$$\left\| \sum_{s=1}^{t-1} \sum_{(i,j)\in\Omega_s^{ind}} \xi_{ij}^s \mathbf{x}_{ij}^s \right\|_{\mathbf{A}_t^{lin-1}}^2 \leq (1/2) \log(\det(\mathbf{A}_t^{lin})^{1/2}/(\delta \det(\lambda \mathbf{I})^{1/2}))$$

With the above lemmas, we prove Lemma 2.1.3 as follows.

Proof of Lemma 2.1.3. We adopt a single layer RankNet model with sigmoid activation function as our pairwise ranker. Given the objective function, Eq (2.1.1), is log-convex with respect to $\boldsymbol{\theta}$, its solution $\widetilde{\boldsymbol{\theta}}_t$ is unique under the following estimating equation at each round t,

$$\sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}}\left(\sigma({\mathbf{x}_{i'j'}^s}^\top\boldsymbol{\theta})-y_{i'j'}^s\right)\mathbf{x}_{i'j'}^s+\lambda\boldsymbol{\theta}=0 \tag{B.1}$$

Let $g_t(\boldsymbol{\theta})=\sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}}\sigma({\mathbf{x}_{i'j'}^s}^\top\boldsymbol{\theta})\mathbf{x}_{i'j'}^s+\lambda\boldsymbol{\theta}$ be the invertible function such that the estimated parameter $\widetilde{\boldsymbol{\theta}}_t$ satisfies $g_t(\widetilde{\boldsymbol{\theta}}_t)=\sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}}y_{i'j'}^s\mathbf{x}_{i'j'}^s$. Since $\widetilde{\boldsymbol{\theta}}_t$ might be outside of the admissible set of parameter Θ that $\boldsymbol{\theta}\in\Theta$ (e.g., $\|\boldsymbol{\theta}\|_2\leq Q$), the estimation can be projected into Θ to obtain $\widehat{\boldsymbol{\theta}}_t$:

$$\widehat{\boldsymbol{\theta}}_t=\operatorname*{argmin}_{\boldsymbol{\theta}\in\Theta}\left\|g_t(\boldsymbol{\theta})-g_t(\widetilde{\boldsymbol{\theta}}_t)\right\|_{\mathbf{A}_t^{lin-1}}=\operatorname*{argmin}_{\boldsymbol{\theta}\in\Theta}\left\|g_t(\boldsymbol{\theta})-\sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}}y_{i'j'}^s\mathbf{x}_{i'j'}^s\right\|_{\mathbf{A}_t^{lin-1}} \tag{B.2}$$

where $\mathbf{A}_t^{lin}$ is defined as $\mathbf{A}_t^{lin}=\sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}}\mathbf{x}_{i'j'}^s{\mathbf{x}_{i'j'}^s}^\top+\lambda\mathbf{I}$. To summarize, $\widetilde{\boldsymbol{\theta}}_t$ is the solution of Eq (4), and $\widehat{\boldsymbol{\theta}}_t$ is the estimated model which is generated by projecting $\widetilde{\boldsymbol{\theta}}_t$ onto Θ.

At round t, for any document pair $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$, the estimation error of the pairwise preference in PairRank is defined as $|\sigma({\mathbf{x}_{ij}^t}^\top\boldsymbol{\theta}^*)-\sigma({\mathbf{x}_{ij}^t}^\top\widehat{\boldsymbol{\theta}}_t)|$, which is based on the model $\widehat{\boldsymbol{\theta}}_t$ learned from the observations until last round. According to Lemma B.1, we have $|\sigma({\mathbf{x}_{ij}^t}^\top\boldsymbol{\theta}^*)-\sigma({\mathbf{x}_{ij}^t}^\top\widehat{\boldsymbol{\theta}}_t)|\leq k_\mu|{\mathbf{x}_{ij}^t}^\top\boldsymbol{\theta}^*-{\mathbf{x}_{ij}^t}^\top\widehat{\boldsymbol{\theta}}_t|$. As logistic function $\sigma(\cdot)$ is continuously differentiable, ∇g_t is continuous. Hence, according to the Fundamental Theorem of Calculus, we have $g_t(\boldsymbol{\theta}^*)-g_t(\widehat{\boldsymbol{\theta}}_t)=\mathbf{G}_t(\boldsymbol{\theta}^*-\widehat{\boldsymbol{\theta}}_t)$, where $\mathbf{G}_t=\int_0^1\nabla g_t\left(l\boldsymbol{\theta}^*+(1-l)\widehat{\boldsymbol{\theta}}_t\right)dl$. Therefore, $\nabla g_t(\boldsymbol{\theta})=\sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}}\dot{\sigma}({\mathbf{x}_{i'j'}^s}^\top\boldsymbol{\theta})\mathbf{x}_{i'j'}^s{\mathbf{x}_{i'j'}^s}^\top+\lambda\mathbf{I}$, where $\dot{\sigma}$ is the first order derivative of σ. As $c_\mu=\inf_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\dot{\sigma}(\mathbf{x}^\top\boldsymbol{\theta})$, it is easy to verify that $c_\mu\leq 1/4$. Thus, we can conclude that $\mathbf{G}_t\succeq c_\mu\mathbf{A}_t^{lin}$. Accordingly, we have the following inequality,

$$\begin{aligned}&\left|\sigma({\mathbf{x}_{ij}^t}^\top\boldsymbol{\theta}^*)-\sigma({\mathbf{x}_{ij}^t}^\top\widehat{\boldsymbol{\theta}}_t)\right|\leq k_\mu\left|{\mathbf{x}_{ij}^t}^\top\mathbf{G}_t^{-1}\left(g_t(\boldsymbol{\theta}^*)-g_t(\widehat{\boldsymbol{\theta}}_t)\right)\right|\\ \leq& k_\mu\|\mathbf{x}_{ij}^t\|_{\mathbf{G}_t^{-1}}\left\|g_t(\boldsymbol{\theta}^*)-g_t(\widehat{\boldsymbol{\theta}}_t)\right\|_{\mathbf{G}_t^{-1}}\leq(\frac{2k_\mu}{c_\mu})\|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{lin-1}}\left\|g_t(\boldsymbol{\theta}^*)-g_t(\widetilde{\boldsymbol{\theta}}_t)\right\|_{\mathbf{A}_t^{lin-1}}\end{aligned}$$

where the first and second inequalities stand as $\mathbf{G}_t$ and $\mathbf{G}_t^{-1}$ are positive definite. The third inequality stands as $\mathbf{G}_t\succeq c_\mu\mathbf{A}_t^{lin}$, which implies that $\mathbf{G}_t^{-1}\preceq c_\mu^{-1}\mathbf{A}_t^{lin^{-1}}$ and $\|\mathbf{x}\|_{\mathbf{G}_t^{-1}}\leq\|\mathbf{x}\|_{\mathbf{A}_t^{lin^{-1}}}/\sqrt{c_\mu}$ hold for any $\mathbf{x}\in\mathbb{R}^d$. The last inequality stands as $\boldsymbol{\theta}^*\in\Theta$, and $\widehat{\boldsymbol{\theta}}_t$ is the optimum solution for Eq (B.2) at round t within Θ.

Based on the definition of $\widetilde{\boldsymbol{\theta}}_t$ and the assumption on the noisy feedback that $y^t=\sigma({\mathbf{x}_t}^\top\boldsymbol{\theta}^*)+\xi^t$, where ξ^t is the noise in user feedback defined in Section 3.2, we have

$$\begin{aligned}g_t(\widetilde{\boldsymbol{\theta}}_t)-g_t(\boldsymbol{\theta}^*)=&\sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}}(y_{i'j'}^s-\sigma({\mathbf{x}_{i'j'}^s}^\top\boldsymbol{\theta}^*))\mathbf{x}_{i'j'}^s-\lambda\boldsymbol{\theta}^*\\ =&\sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}}\xi_{i'j'}^s\mathbf{x}_{i'j'}^s-\lambda\boldsymbol{\theta}^*=S_t-\lambda\boldsymbol{\theta}^*.\end{aligned}$$

where we define $S_t = \sum_{s=1}^{t-1} \sum_{(i',j')\in\Omega_s^{ind}} \xi_{i'j'}^s \mathbf{x}_{i'j'}^s$.

Therefore, the confidence interval of the estimated pairwise preference in PairRank can be derived as:

$$\begin{aligned}\left|\sigma({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}^*) - \sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t)\right| \leq & (\frac{2k_\mu}{c_\mu}) \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{lin-1}} \|S_t - \lambda\boldsymbol{\theta}^*\|_{\mathbf{A}_t^{lin-1}} \\ \leq & (\frac{2k_\mu}{c_\mu}) \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{lin-1}} \left(\|S_t\|_{\mathbf{A}_t^{lin-1}} + \sqrt{\lambda}Q\right)\end{aligned}$$

where the second inequality is based on minimum eigenvalue $\lambda_{\min}(\mathbf{A}_t^{lin}) \geq \lambda$ and $\|\boldsymbol{\theta}\|_2 \leq Q$.

As $\xi_{i'j'}^t \sim R$-sub-Gaussian, according to Theorem 1 in [14],

$$\mathbb{P}\left[\|S_t\|^2_{\mathbf{A}_t^{lin-1}} > 2R^2 \log \frac{\det(\mathbf{A}_t^{lin})^{1/2}}{\delta \det(\lambda\mathbf{I})^{1/2}}\right] \leq \delta$$

Therefore, with probability at least $1-\delta_1$, we have

$$\left|\sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t) - \sigma({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}^*)\right| \leq \alpha_t \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{lin-1}}$$

with $\alpha_t = (2k_\mu/c_\mu)\left(\sqrt{R^2 \log \frac{\det(\mathbf{A}_t^{lin})}{\delta_1^2 \det(\lambda\mathbf{I})}} + \sqrt{\lambda}Q\right)$ □

B.2 Proof of Lemma 2.1.10

Proof. Proof of Lemma 2.1.10

In this proof, we will first provide an upper bound of the minimum eigenvalue of $\mathbf{A}_t^{lin}$, and then provide the detailed derivation of the probability's upper bound.

• **Upper bound the minimum eigenvalue of $\mathbf{A}_t^{lin}$.** As discussion in Lemma 2.1.9 , we assume that pairwise feature vectors are random vectors drawn from distribution v. With $\boldsymbol{\Sigma} = \mathbb{E}[\mathbf{x}_{i'j'}^s {\mathbf{x}_{i'j'}^s}^\top]$ as the second moment matrix, define $\boldsymbol{Z} = \boldsymbol{\Sigma}^{\frac{-1}{2}}\mathbf{X}$, where $\mathbf{X}$ is a random vector drawn from the same distribution v. Then $\boldsymbol{Z}$ is isotropic, namely $\mathbb{E}[\boldsymbol{Z}\boldsymbol{Z}^\top] = \mathbf{I}_d$.

Define $\boldsymbol{U} = \sum_{s=1}^{t-1} \sum_{(i',j')\in\Omega_s^{ind}} \boldsymbol{Z}_{i'j'}^s {\boldsymbol{Z}_{i'j'}^s}^\top = \Sigma^{\frac{-1}{2}} \bar{\mathbf{A}}_t^{lin} \Sigma^{\frac{-1}{2}}$, where $\bar{\mathbf{A}}_t^{lin} = \sum_{s=1}^{t-1} \sum_{(i',j')\in\Omega_s^{ind}} \mathbf{x}_{i'j'}^s {\mathbf{x}_{i'j'}^s}^\top = \mathbf{A}_t^{lin} - \lambda\mathbf{I}$. From Lemma 2.1.9, we know that for any l, with probability at least $1 - 2\exp(-C_2 l^2)$, $\lambda_{\min}(\boldsymbol{U}) \geq n - C_1\sigma^2\sqrt{nd} - \sigma^2 l\sqrt{n}$, where σ is the sub-Gaussian parameter of $\boldsymbol{Z}$, which is upper-bounded by $\|\boldsymbol{\Sigma}^{-1/2}\| = \lambda_{\min(\boldsymbol{\Sigma})}$, and $n = \sum_{s=1}^{t-1} |\Omega_s^{ind}|$, represents the number of observations so far. We thus can rewrite the above inequality which holds with probability $1-\delta_2$ as $\lambda_{\min}(\boldsymbol{U}) \geq n - \lambda_{\min}^{-1}(\boldsymbol{\Sigma})(C_1\sqrt{nd} + l\sqrt{n})$. We now bound the minimum eigenvalue of $\bar{\mathbf{A}}_t^{lin}$, as follows:

$$\begin{aligned}\lambda_{\min}(\bar{\mathbf{A}}_t^{lin}) &= \min_{x\in\mathbb{B}^d} x^\top \bar{\mathbf{A}}_t^{lin} x = \min_{x\in\mathbb{B}^d} x^\top \boldsymbol{\Sigma}^{1/2}\boldsymbol{U}\boldsymbol{\Sigma}^{1/2} x \geq \lambda_{\min}(\boldsymbol{U}) \min_{x\in\mathbb{B}^d} x^\top \boldsymbol{\Sigma} x \\ &= \lambda_{\min}(\boldsymbol{U})\lambda_{\min}(\boldsymbol{\Sigma}) \geq \lambda_{\min}(\boldsymbol{\Sigma})n - C_1\sqrt{nd} - C_2\sqrt{n\log(1/\delta_2)}\end{aligned}$$

According to Lemma 2.1.9, for $t \geq t'$, we have:

$$\lambda_{\min}(\mathbf{\Sigma})t - (c_1\sqrt{d} + c_2\sqrt{\log(1/\delta_2)} + abd\sqrt{o_{\max}u^2/(d\lambda)})\sqrt{t} - (ab\log(1/\delta_1^2) + 4a\lambda Q^2 - \lambda) \geq 0$$

As $n \geq t$, we have

$$\lambda_{\min}(\mathbf{A}_t^{lin}) \geq \lambda_{\min}(\bar{\mathbf{A}}_t^{lin}) + \lambda \geq abd\sqrt{\frac{o_{\max}u^2}{d\lambda}} + ab\log(\frac{1}{\delta_1^2}) + 4a\lambda Q^2 \tag{B.3}$$

• **Upper bound the probability of being uncertain rank order**. Under event E_t, based on the definition of ω_u^t in Section 3.2, we know that for any document $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$ at round t, $(i,j) \in \omega_u^t$ if and only if $\sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t) - \alpha_t\|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{lin-1}} \leq \frac{1}{2}$ and $\sigma({\mathbf{x}_{ji}^t}^\top \widehat{\boldsymbol{\theta}}_t) - \alpha_t\|\mathbf{x}_{ji}^t\|_{\mathbf{A}_t^{lin-1}} \leq \frac{1}{2}$. For a logistic function, we know that $\sigma(s) = 1 - \sigma(-s)$. Therefore, let CB_{ij}^t denote $\alpha_t\|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{lin-1}}$, we can conclude that $(i,j) \in \omega_u^t$ if and only if $|\sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t) - \frac{1}{2}| \leq CB_{ij}^t$; and accordingly, $(i,j) \in \omega_c^t$, when $|\sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t) - \frac{1}{2}| > CB_{ij}^t$. To further simplify our notations, in the following analysis, we use $\widehat{\sigma}_t$ and σ^* to present $\sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t)$ and $\sigma({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}^*)$ respectively. According to the discussion above, the probability that the estimated preference between document $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$ to be in an uncertain rank order, e.g., $(i,j) \in \omega_u^t$ can be upper bounded by:

$$\begin{aligned}\mathbb{P}\big((i,j) \in \omega_u^t\big) &= \mathbb{P}\big(|\widehat{\sigma}_t - 1/2| \leq CB_{ij}^t\big) \leq \mathbb{P}\left(||\widehat{\sigma}_t - \sigma^*| - |\sigma^* - 1/2|| \leq CB_{ij}^t\right) \\ &\leq \mathbb{P}\left(|\sigma^* - 1/2| - |\widehat{\sigma}_t - \sigma^*| \leq CB_{ij}^t\right) \leq \mathbb{P}\left(\Delta_{\min} - |\widehat{\sigma}_t - \sigma^*| \leq CB_{ij}^t\right).\end{aligned}$$

where, the first inequality is based on the reverse triangle inequality. The last inequality is based on the definition of $\Delta_{\min}$. Based on Lemma 2.1.3, the above probability can be further bounded by

$$\begin{aligned}&\mathbb{P}\left(\Delta_{\min} - |\widehat{\sigma}_t - \sigma^*| \leq CB_{ij}^t\right) = \mathbb{P}\left(|\widehat{\sigma}_t - \sigma^*| \geq \Delta_{\min} - \alpha_t\|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{lin-1}}\right) \\ \leq&\mathbb{P}\left((2k_\mu/c_\mu)||\mathbf{x}_{ij}^t||_{\mathbf{A}_t^{lin-1}}\left(\|S_t\|_{\mathbf{A}_t^{lin-1}} + \sqrt{\lambda}Q\right) \geq \Delta_{\min} - \alpha_t\|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{lin-1}}\right) \\ \leq&\mathbb{P}\left(\|S_t\|_{\mathbf{A}_t^{lin-1}} \geq \frac{c_\mu\Delta_{\min}}{2k_\mu||\mathbf{x}_{ij}^t||_{\mathbf{A}_t^{lin-1}}} - \left(\sqrt{\frac{1}{4}\log\frac{\det(\mathbf{A}_t^{lin})}{\delta^2\det(\lambda\mathbf{I})}} + 2\sqrt{\lambda}Q\right)\right)\end{aligned}$$

For the RHS of the inequality inside the probability, we know that:

$$\begin{aligned}&\left(\frac{c_\mu\Delta_{\min}}{2k_\mu||\mathbf{x}_{ij}^t||_{\mathbf{A}_t^{lin-1}}}\right)^2 - \left(\sqrt{R^2\log\frac{\det(\mathbf{A}_t^{lin})}{\delta^2\det(\lambda\mathbf{I})}} + 2\sqrt{\lambda}Q\right)^2 \\ \geq&\lambda_{\min}(\mathbf{A}_t^{lin})/a - R^2\log(\det(\mathbf{A}_t^{lin})/(\delta_1^2\det(\lambda\mathbf{I}))) - 4\lambda Q^2 - 4\sqrt{\lambda}QR\sqrt{\log(\det(\mathbf{A}_t^{lin})/\det\lambda\mathbf{I}) + \log(1/\delta_1^2)} \\ \geq&\lambda_{\min}(\mathbf{A}_t^{lin})/a - b\left(\log(\det(\mathbf{A}_t^{lin})/\det(\lambda\mathbf{I})) + \log(1/\delta_1^2)\right) - 4\lambda Q^2 \\ \geq&\frac{1}{a}\cdot\lambda_{\min}(\mathbf{A}_t^{lin}) - b\cdot\left(d\log(1 + \frac{o_{\max}tU^2}{d\lambda}) + \log(\frac{1}{\delta_1^2})\right) - 4\lambda Q^2 \\ \geq&\lambda_{\min}(\mathbf{A}_t^{lin})/a - bd\log(1 + \frac{o_{\max}tu^2}{d\lambda}) - b\log(\frac{1}{\delta_1^2}) - 4\lambda Q^2 \geq 0\end{aligned}$$

where the last inequality follows Eq (B.3). Therefore, the probability could be upper bounded as:

$$
\begin{aligned}
&\mathbb{P}\left(\Delta_{\min} - |\widehat{\sigma}_t - \sigma^*| \leq CB_{ij}^t\right) \\
\leq &\mathbb{P}\left(\|S_t\|^2_{\mathbf{A}_t^{lin-1}} \geq \left(\frac{c_\mu \Delta_{\min}}{2k_\mu \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{lin-1}}} - \left(\sqrt{R^2 \log\frac{\det(\mathbf{A}_t^{lin})}{\delta \det(\lambda \mathbf{I})}} + 2\sqrt{\lambda}Q\right)\right)^2\right) \\
\leq &\mathbb{P}\left(\|S_t\|^2_{\mathbf{A}_t^{lin-1}} \geq \frac{(1-2\beta)c_\mu^2 \Delta_{\min}^2}{4k_\mu^2 \|\mathbf{x}_{ij}^t\|^{-1}_{\mathbf{A}_t^{lin}}} + R^2 \log(\frac{\det(\mathbf{A}_t^{lin})}{\delta_1^2 \det(\lambda \mathbf{I})})\right) \\
\leq &\mathbb{P}\left(\|S_t\|^2_{\mathbf{A}_t^{lin-1}} \geq 2R^2 \log\left(\exp\left(\frac{(1-2\beta)c_\mu^2 \Delta_{\min}^2}{8R^2 k_\mu^2 \|\mathbf{x}_{ij}^t\|^{-1}_{\mathbf{A}_t^{lin}}}\right) \cdot \frac{\det(\mathbf{A}_t^{lin})}{\delta_1^2 \det(\lambda \mathbf{I})})\right)\right) \\
\leq &\delta_1 \cdot \exp^{-1}\left(\frac{(1-2\beta)c_\mu^2 \Delta_{\min}^2}{8R^2 k_\mu^2 \|\mathbf{x}_{ij}^t\|^2_{\mathbf{A}_t^{lin-1}}}\right) \leq \log(\frac{1}{\delta_1}) \cdot \frac{8R^2 k_\mu^2 \|\mathbf{x}_{ij}^t\|^2_{\mathbf{A}_t^{lin-1}}}{(1-2\beta)c_\mu^2 \Delta_{\min}^2}
\end{aligned}
$$

□

B.3 Proof of Theorem 2.1.11

Proof. Proof of Theorem 2.1.11

With δ_1 and δ_2 defined in the previous lemmas, we have the T-step regret upper bounded as:

$$
R_T = R_{t'} + R_{T-t'} \leq R' + (1-\delta_1)(1-\delta_2) \sum_{s=t'}^{T} r_s \tag{B.4}
$$

where $R' = t' \cdot V_{\max} + (T-t')\left[\delta_2 \cdot V_{\max} + (1-\delta_2) \cdot \delta_1 V_{\max}\right]$ Under event E_t^1 and E_t^2, the instantaneous regret at round s is bounded by

$$
r_s = \mathbb{E}\big[K(\tau_s, \tau_s^*)\big] = \sum_{i=1}^{d_s} \mathbb{E}\left[(N_i^s + 1)N_i^s/2\right] \leq \mathbb{E}\left[N_s(N_s+1)/2\right] \tag{B.5}
$$

where N_i^s denotes the number of uncertain rank orders in block $\mathcal{B}_i^s$ at round s and N_s denotes the total number of uncertain rank orders at round s. It follows that in the worst case scenario, when N_s uncertain rank orders are all placed into the same block, at most $(N_s^2 + N_s)/2$ mis-ordered pairs will be generated by random shuffling in PairRank. This is due to the fact that based on the block created by PairRank, with N_s uncertain rank orders in one block, this block can have at most $N_s + 1$ documents.

Therefore, the cumulative regret after t' can be bounded by:

$$
\sum_{s=t'}^{T} r_s \leq \sum_{s=t'}^{T} \mathbb{E}\left[N_s(N_s+1)/2\right] \leq \left(\sum_{s=t'}^{T} \mathbb{E}[N_s^2] + \mathbb{E}[N_s]\right)/2 \leq \left(\mathbb{E}[\sum_{s=t'}^{T} N_s]^2 + \mathbb{E}[\sum_{s=t'}^{T} N_s]\right)/2
$$

According to our previous analysis, at round $t \geq t'$, the number of uncertain rank orders can be estimated by the probability of observing an uncertain rank order, i.e., $\mathbb{P}\big((i,j) \in \omega_u^t\big) \leq \log(1/\delta_1) \cdot \frac{8R^2 k_\mu^2 \|\mathbf{x}_{ij}^t\|^2_{\mathbf{A}_t^{lin-1}}}{(1-2\beta)c_\mu^2 \Delta_{\min}^2}$.

Therefore, the cumulative number of mis-ordered pairs can be bounded by the probability of observing uncertain rank orders in each round, which shrinks with more observations become available over time,

$$\mathbb{E}\left[\sum_{s=t'}^{T} N_s\right] \leq \mathbb{E}\left[\frac{1}{2}\sum_{s=t'}^{T}\sum_{(i',j')\in[V_s]^2} \mathbb{P}\big((i',j')\in\omega_u^t\big)\right] \leq \mathbb{E}\left[a\sum_{s=t'}^{T}\sum_{(i',j')\in[V_s]^2} \|\mathbf{x}_{i'j'}^s\|^2_{\mathbf{A}_s^{lin-1}}\right]$$

where $a = \log(1/\delta_1)\cdot 4k_\mu^2/((1-2\beta)c_\mu^2\Delta_{\min}^2)$

Because $\mathbf{A}_t^{lin}$ only contains information of observed document pairs so far, PairRank guarantees the number of mis-ordered pairs among the observed documents in the above inequality is upper bounded. To reason about the number of mis-ordered pairs in those unobserved documents (i.e., from o_t to V_t for each query q_t), we leverage the constant p^*, which is defined as the minimal probability that all documents in a query are examined over time,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{s=t'}^{T}\sum_{(i',j')\in[V_s]^2} \|\mathbf{x}_{i'j'}^s\|^2_{\mathbf{A}_s^{lin-1}}\right] = \mathbb{E}\left[\sum_{s=t'}^{T}\sum_{(i',j')\in[V_s]^2} \|\mathbf{x}_{i'j'}^s\|^2_{\mathbf{A}_s^{lin-1}} \times \mathbb{E}\left[\frac{1}{p_s}\mathbf{1}\{o_s = V_s\}\right]\right] \\
\leq& {p^*}^{-1}\mathbb{E}\left[\sum_{s=t'}^{T}\sum_{(i',j')\in[V_s]^2} \|\mathbf{x}_{i'j'}^s\|^2_{\mathbf{A}_s^{lin-1}}\mathbf{1}\{o_s = V_s\}\right]
\end{aligned}$$

Besides, in PairRank, we only use the independent pairs, Ω_t^{ind} to update the model and the corresponding $\mathbf{A}_t^{lin}$ matrix. Therefore, to bound the regret, we rewrite the above equation as:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{s=t'}^{T}\sum_{(i',j')\in[V_s]^2} \|\mathbf{x}_{i'j'}^s\|^2_{\mathbf{A}_s^{lin-1}}\right] \\
=&\mathbb{E}\left[\sum_{s=t'}^{T}\sum_{(i',j')\in\Omega_t^{ind}} \left(\|\mathbf{x}_{i'j'}^s\|^2_{\mathbf{A}_s^{lin-1}} + \sum_{k\in[V_t]\setminus\{i',j'\}} \|\mathbf{x}_{i'k}^s\|^2_{\mathbf{A}_s^{lin-1}} + \|\mathbf{x}_{j'k}^s\|^2_{\mathbf{A}_s^{lin-1}}\right)\right] \\
=&\mathbb{E}\left[\sum_{s=t'}^{T}\sum_{(i',j')\in\Omega_t^{ind}} \left((V_t-1)\|\mathbf{x}_{i'j'}^s\|^2_{\mathbf{A}_s^{lin-1}} + \sum_{k\in[V_t]\setminus\{i',j'\}} 2{\mathbf{x}_{i'k}^s}^\top {\mathbf{A}_s^{lin}}^{-1}\mathbf{x}_{j'k}^s\right)\right]
\end{aligned}$$

Based on Lemma 10 and Lemma 11 in [14], the expected number of the first term can be bounded by,

$$\sum_{s=t'}^{T}\sum_{(i',j')\in\Omega_t^{ind}} (V_t-1)\|\mathbf{x}_{i'j'}^s\|^2_{\mathbf{A}_s^{lin-1}} \leq 2dL_{max}\log(1+\frac{o_{\max}Tu^2}{2d\lambda})$$

And the second term can be bounded as:

$$\sum_{s=t'}^{T} \sum_{(i',j')\in\Omega_t^{ind}} \sum_{k\in[V_t]\setminus\{i',j'\}} 2{\mathbf{x}_{i'k}^s}^\top {\mathbf{A}_s^{lin}}^{-1} \mathbf{x}_{j'k}^s \leq \sum_{s=t'}^{T} (L_{max}^2 - 2L_{max})u^2/\lambda_{\min}(\mathbf{A}_s^{lin}) = w.$$

By chaining all the inequalities, we have the regret upper bounded by:

$$R_T \leq R' + (1-\delta_1)(1-\delta_2)\frac{1}{p^{*2}}\left(2adV_{\max}\log(1+\frac{o_{\max}Tu^2}{2d\lambda}) + aw\right)^2$$

where $R' = t'V_{\max} + (T-t')(\delta_2 V_{\max} - (1-\delta_2)\delta_1 V_{\max})$, with t' defined in Lemma 2.1.10, and $V_{\max}$ representing the maximum number of documents associated to the same query over time. By choosing $\delta_1 = \delta_2 = 1/T$, the theorem shows that the expected regret is at most $R_T \leq O(d\log^4(T))$.

□

B.4 Proof of Lemma 2.1.6

In order to prove Lemma 2.1.6, we need the following technical lemmas.

Lemma B.3 (Lemma 5.1, [15])**.** There exists a positive constant $\bar{C}$ such that for any $\delta \in (0,1)$, if $m \geq \bar{C}n_T^4 L^6 \log(n_T^2 L/\delta)/\lambda_0^4$, then with probability at least $1-\delta$, there exists a $\boldsymbol{\theta}^* \in \mathbb{R}^p$ such that for any $i \in [n_T]$, with $\mathbf{h} = (h(\mathbf{x}_1), \ldots, h(\mathbf{x}_n))$.

$$h(\mathbf{x}_i) = \langle \mathbf{g}(\mathbf{x}_i;\boldsymbol{\theta}_0), \boldsymbol{\theta}^* - \boldsymbol{\theta}_0\rangle, \quad \sqrt{m}\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\|_2 \leq \sqrt{2\mathbf{h}^\top \mathbf{H}^{-1}\mathbf{h}} \leq S \tag{B.6}$$

Lemma B.4 (Lemma B.3, [15])**.** There exist constants $\{C_i^\epsilon\}_{i=1}^5 > 0$ such that for any $\delta > 0$, if m satisfies,

$$C_1^\epsilon m^{-3/2} L^{-3/2}[\log(n_T L^2/\delta)]^{3/2} \leq \tau \leq C_2^\epsilon L^{-6}[\log m]^{-3/2}$$

then with probability at least $1-\delta$, for any $t \in [T]$, we have

$$\begin{aligned}
&\|\mathbf{A}_t^{neu}\|_2 \leq \lambda + C_3^\epsilon n_t^P L,\\
&\|\bar{\mathbf{A}}_t^{neu} - \mathbf{A}_t^{neu}\|_F \leq C_4^\epsilon n_t^P \sqrt{\log(m)}\tau^{1/3} L^4,\\
&\left|\log\frac{\det(\bar{\mathbf{A}}_t^{neu})}{\det(\lambda\mathbf{I})} - \log\frac{\det(\mathbf{A}_t^{neu})}{\det(\lambda\mathbf{I})}\right| \leq C_5^\epsilon (n_t^P)^{3/2}\lambda^{-1/2}\sqrt{\log(m)}\tau^{1/3}L^4
\end{aligned}$$

For the constants, $C_3^\epsilon = 2C_3^z$, $C_4^\epsilon = 8C_3^w(C_3^z)^2$, and $C_5^\epsilon = 4C_3^w(C_3^z)^2$, with C_3^z and C_3^w from the technique lemmas, Lemma B.6 and Lemma B.7.

Lemma B.5 (Lemma B.4, [15])**.** There exist constants $\{C_i^v\}_{i=1}^3 > 0$ such that for any $\delta > 0$, if τ satisfies that

$$C_1^v m^{-3/2} L^{-3/2}[\log(n_T L^2/\delta)]^{3/2} \leq \tau \leq C_2^v L^{-6}[\log m]^{-3/2},$$

then with probability at least $1-\delta$, for any $\widetilde{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{\theta}}$ satisfying $\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\|_2 \leq \tau$, $\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\|_2 \leq \tau$ and $i \in [n_T]$ we have

$$\left|f(\mathbf{x}_i;\widetilde{\boldsymbol{\theta}}) - f(\mathbf{x}_i;\widehat{\boldsymbol{\theta}}) - \langle \mathbf{g}(\mathbf{x}_i;\widehat{\boldsymbol{\theta}}), \widetilde{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}\rangle\right| \leq C_3^v \tau^{4/3} L^3 \sqrt{m\log m}.$$

Lemma B.6 (Lemma B.5, [15])**.** There exist constants $\{C_i^w\}_{i=1}^3 > 0$ such that for any $\delta \in (0,1)$, if τ satisfies,

$$C_1^w m^{-3/2} L^{-3/2} \max\{\log^{-3/2} m, \log^{3/2}(n_T/\delta)\} \leq \tau \leq C_2^w L^{-9/2} \log^{-3} m,$$

then with probability at least $1-\delta$, for all $\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2 \leq \tau$ and $i \in [n_T]$ we have

$$\|\mathbf{g}(\mathbf{x}_i;\boldsymbol{\theta}) - \mathbf{g}(\mathbf{x}_i;\boldsymbol{\theta}_0)\|_2 \leq C_3^w \sqrt{\log m}\tau^{1/3} L^3 \|\mathbf{g}(\mathbf{x}_i;\boldsymbol{\theta}_0)\|_2.$$

Lemma B.7 (Lemma B.6, [15])**.** There exist constants $\{C_i^z\}_{i=1}^3 > 0$ such that for any $\delta > 0$, if τ satisfies that

$$C_1^z m^{-3/2} L^{-3/2} [\log(n_T L^2/\delta)]^{3/2} \leq \tau \leq C_2^z L^{-6} [\log m]^{-3/2},$$

then with probability at least $1-\delta$, for any $\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2 \leq \tau$ and $i \in [n_T]$ we have $\|\mathbf{g}(\mathbf{x}_i;\boldsymbol{\theta})\|_F \leq C_3^z \sqrt{mL}$.

We also need the following lemmas. The first lemma is based on the generalized linear bandit [253] and the analysis of linear bandit in [14]. For the second lemma, we adapted it from the original paper with our pairwise cross-entropy loss. The key difference lies in 1) the different number of observations in each round, which affects the required condition on the width of the neural network m; 2) we extend the original error bound analysis for the least square loss to the generalized linear model, e.g., logistic regression model.

Lemma B.8. For any $t \in [T]$, with $\widehat{\boldsymbol{\gamma}}_t$ defined as the solution of the following equation,

$$\widehat{\boldsymbol{\gamma}}_t = \min_{\boldsymbol{\gamma}} \sum_{s=1}^{t-1} \sum_{(i,j)\in\Omega_s^{ind}} -y_{i,j}^s \log\Big(\sigma(\langle \mathbf{g}_{i,j}^{t,0}, \boldsymbol{\gamma}\rangle)\Big) - (1-y_{i,j}^s)\log\Big(1-\sigma(\langle \mathbf{g}_{i,j}^{t,0}, \boldsymbol{\gamma}\rangle)\Big) + \frac{m\lambda}{2}\|\boldsymbol{\gamma}\|^2. \quad \text{(B.7)}$$

Then, with the pairwise noise ξ_{ij}^ssatisfying Proposition 2.1.2, for any $(i,j) \in \Psi_t$, with probability at least $1-\delta_1$, we have,

$$\|\sqrt{m}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\gamma}}_t)\|_{\bar{\mathbf{A}}_t^{neu}} \leq c_\mu^{-2}\Big(\sqrt{\nu^2 \log(\det(\bar{\mathbf{A}}_t^{neu}))/(\delta_1^2 \det(\lambda \mathbf{I}))} + \sqrt{\lambda} S\Big)$$

Lemma B.9 (Lemma B.2, [15])**.** There exist constants $\{\bar{C}_i\}_{i=1}^6 > 0$ such that for any $\delta > 0$, if for all $t \in [T]$, η and m satisfy

$$\begin{aligned}
&\sqrt{2n_t^P/(m\lambda)} \geq \bar{C}_1 m^{-3/2} L^{-3/2} [\log(n_T L^2/\delta)]^{3/2},\\
&\sqrt{2n_t^P/(m\lambda)} \leq \bar{C}_2 \min\big\{L^{-6}[\log m]^{-3/2}, \big(m(\lambda\eta)^2 L^{-6}(n_t^P)^{-1}(\log m)^{-1}\big)^{3/8}\big\},\\
&\eta \leq \bar{C}_3 (m\lambda + n_t^P m L)^{-1},\\
&m^{1/6} \geq \bar{C}_4 \sqrt{\log m} L^{7/2} (n_t^P)^{7/6} \lambda^{-7/6} (1 + \sqrt{n_t^P/\lambda}),
\end{aligned}$$

then with probability at least $1-\delta$, we have that $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0\|_2 \leq \sqrt{2n_t^P/(m\lambda)}$ and

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\gamma}}_t\|_2 \leq (1-\eta m\lambda)^{J/2}\sqrt{2n_t^P/(m\lambda)} + m^{-2/3}\sqrt{\log m} L^{7/2}(n_t^P)^{7/6}\lambda^{-7/6}(\bar{C}_5 + \bar{C}_6\sqrt{n_t^P/\lambda}).$$

Proof of Lemma 2.1.6. We first bound the estimated pairwise preference based on the Lipschitz continuity:

$$\Big|\sigma(f(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}) - f(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1})) - \sigma(h(\mathbf{x}_i^t) - h(\mathbf{x}_j^t))\Big| \leq k_\mu \Big|f(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}) - f(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1}) - \Big(h(\mathbf{x}_i^t) - h(\mathbf{x}_j^t)\Big)\Big|$$

According to Lemma B.3, and $f(\mathbf{x};\boldsymbol{\theta}_0)=0$, we could have the following equation for document $\mathbf{x}_i^t$,

$$\begin{aligned}f(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1})-h(\mathbf{x}_i^t)=&f(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1})-f(\mathbf{x}_i^t;\boldsymbol{\theta}_0)-\langle\mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}),\boldsymbol{\theta}_{t-1}-\boldsymbol{\theta}_0\rangle\\&+\langle\mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}),\boldsymbol{\theta}_{t-1}-\boldsymbol{\theta}_0\rangle-\langle\mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}),\boldsymbol{\theta}^*-\boldsymbol{\theta}_0\rangle\\&+\langle\mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}),\boldsymbol{\theta}^*-\boldsymbol{\theta}_0\rangle-\langle\mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_0),\boldsymbol{\theta}^*-\boldsymbol{\theta}_0\rangle.\end{aligned}$$

Therefore, we could have the following inequalities based on the triangle inequality.

$$\begin{aligned}&\Big|f(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1})-f(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1})-\Big(h(\mathbf{x}_i^t)-h(\mathbf{x}_j^t)\Big)\Big|\\\leq&\Big|\langle\mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1})-\mathbf{g}(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1}),\boldsymbol{\theta}_{t-1}-\boldsymbol{\theta}^*\rangle\Big|\\&+\|\boldsymbol{\theta}^*-\boldsymbol{\theta}_0\|_2\Big(\|\mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1})-\mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_0)\|_2+\|\mathbf{g}(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1})-\mathbf{g}(\mathbf{x}_j^t;\boldsymbol{\theta}_0)\|_2\Big)\\&+\Big|f(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1})-f(\mathbf{x}_i^t;\boldsymbol{\theta}_0)-\langle\mathbf{g}(\mathbf{x}_i^t;\boldsymbol{\theta}_{t-1}),\boldsymbol{\theta}_{t-1}-\boldsymbol{\theta}_0\rangle\Big|\\&+\Big|f(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1})-f(\mathbf{x}_j^t;\boldsymbol{\theta}_0)-\langle\mathbf{g}(\mathbf{x}_j^t;\boldsymbol{\theta}_{t-1}),\boldsymbol{\theta}_{t-1}-\boldsymbol{\theta}_0\rangle\Big|\\\leq&2C_3^v\tau^{4/3}L^3\sqrt{m\log m}+2C_3^zC_3^w\tau^{1/3}L^{7/2}\sqrt{\log(m)}S+\Big|\langle\mathbf{g}_{ij}^t,\boldsymbol{\theta}_{t-1}-\boldsymbol{\theta}^*\rangle\Big|,\end{aligned}$$

where the last inequality is due to Lemma B.3, B.5, B.6, B.7, with satisfied τ as the upper bound of $\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2$.

Now we start to bound the last term $\Big|\langle\mathbf{g}_{ij}^t,\boldsymbol{\theta}_{t-1}-\boldsymbol{\theta}^*\rangle\Big|$.

$$\begin{aligned}\Big|\langle\mathbf{g}_{ij}^t,\boldsymbol{\theta}_{t-1}-\boldsymbol{\theta}^*\rangle\Big|=&\Big|\langle\mathbf{g}_{ij}^t,\boldsymbol{\theta}_{t-1}-\boldsymbol{\theta}_0-\widehat{\boldsymbol{\gamma}}_t-(\boldsymbol{\theta}^*-\boldsymbol{\theta}_0-\widehat{\boldsymbol{\gamma}}_t)\rangle\Big|\\\leq&|\langle\mathbf{g}_{ij}^t,\boldsymbol{\theta}^*-\boldsymbol{\theta}_0-\widehat{\boldsymbol{\gamma}}_t\rangle|+\|\mathbf{g}_{ij}^t\|\|\boldsymbol{\theta}_{t-1}-\boldsymbol{\theta}_0-\widehat{\boldsymbol{\gamma}}_t\|\end{aligned}\tag{B.8}$$

For the first term, we have the following analysis.

$$\begin{aligned}|\langle\mathbf{g}_{ij}^t,\boldsymbol{\theta}^*-\boldsymbol{\theta}_0-\widehat{\boldsymbol{\gamma}}_t\rangle|\leq&\|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{neu-1}}\|\sqrt{m}(\boldsymbol{\theta}^*-\boldsymbol{\theta}_0-\widehat{\boldsymbol{\gamma}}_t)\|_{\mathbf{A}_t^{neu}}\\\leq&\|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{neu-1}}\sqrt{(1+\|\mathbf{A}_t^{neu}-\bar{\mathbf{A}}_t^{neu}\|_2/\lambda)}\|\sqrt{m}(\boldsymbol{\theta}^*-\boldsymbol{\theta}_0-\widehat{\boldsymbol{\gamma}}_t)\|_{\bar{\mathbf{A}}_t^{neu}}\\\leq&\sqrt{1+C_4^\epsilon n_t^P\sqrt{\log(m)}\tau^{1/3}L^4}\|\sqrt{m}(\boldsymbol{\theta}^*-\boldsymbol{\theta}_0-\widehat{\boldsymbol{\gamma}}_t)\|_{\bar{\mathbf{A}}_t^{neu}}\|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{neu-1}},\end{aligned}$$

where the first inequality is trivial, and the second inequality is due to the fact that $\mathbf{x}^\top\mathbf{P}\mathbf{x}\leq\mathbf{x}^\top\mathbf{Q}\mathbf{x}\cdot\|\mathbf{P}\|_2/\lambda_{\min}(\mathbf{Q})$, and $\lambda_{\min}(\bar{\mathbf{A}}_t^{neu})\geq\lambda$, the third inequality is based on Lemma B.4 with $\|\mathbf{A}_t^{neu}-\bar{\mathbf{A}}_t^{neu}\|_2\leq\|\mathbf{A}_t^{neu}-\bar{\mathbf{A}}_t^{neu}\|_F$. According to Lemma B.8, with probability $1-\delta_1$, we have

$$\begin{aligned}\|\sqrt{m}(\boldsymbol{\theta}^*-\boldsymbol{\theta}_0-\widehat{\boldsymbol{\gamma}}_t)\|_{\bar{\mathbf{A}}_t^{neu}}\leq&c_\mu^{-2}(\sqrt{\nu^2\log(\det(\bar{\mathbf{A}}_t^{neu}))/(\delta_1^2\det(\lambda\mathbf{I}))}+\sqrt{\lambda}S)\\\leq&c_\mu^{-2}(\sqrt{\nu^2\log(\det(\mathbf{A}_t^{neu}))/(\delta_1^2\det(\lambda\mathbf{I}))+C_5^\epsilon(n_t^P)^{3/2}\lambda^{-1/2}\sqrt{\log(m)}\tau^{1/3}L^4}+\sqrt{\lambda}S)\end{aligned}$$

where the second inequality is based on Lemma B.4. For the second term of Eq B.8, it can be bounded according to Lemma B.7 and Lemma B.9. By chaining all the inequalities, and with $\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|\leq\tau\leq\sqrt{2n_t^P/(m\lambda)}$, and the satisfied m and η, we complete the proof. □

B.5 Proof of Lemma 2.1.15

Proof of Lemma 2.1.15. In this proof we will first provide an analysis on the minimum eigenvalue of $\mathbf{A}_t^{\bar{n}eu}$, and then provide the detailed derivation of the upper bound of the probability.

At initialization, DNNs are equivalent to Gaussian processes in the infinite-width limit. Thus, we assume that the gradient differences between the documents at the initial step are random vectors drawn from some

distribution v. With $\mathbf{\Sigma} = \mathbb{E}[\mathbf{g}_{ij}^0 \mathbf{g}_{ij}^{0\top}]$ as the second moment matrix, define $\mathbf{Z} = \mathbf{\Sigma}^{-1/2}\mathbf{X}$, where $\mathbf{X}$ is a random vector drawn from the same distribution v. Then $\mathbf{Z}$ is isotropic, namely $\mathbb{E}[\mathbf{Z}\mathbf{Z}^\top] = \mathbf{I}_p$. Define $\mathbf{D} = \sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}} \mathbf{Z}_{i'j'}^s \mathbf{Z}_{i'j'}^{s\top}$, where $\mathbf{Z}_{i'j'}^s = \Sigma^{-1/2}\mathbf{g}_{i'j'}^{s,0}$. It is trivial to have $\mathbf{D} = \Sigma^{-1/2}(\mathbf{A}_t^{\bar{neu}} - \lambda\mathbf{I})\Sigma^{-1/2}$. From Lemma 2.1.9, we know that for any l, with probability at least $1 - 2\exp(-C_2 l^2)$, $\lambda_{\min}(\mathbf{D}) \geq n_t - C_1\sigma^2 n_t - \sigma^2 l\sqrt{n_t}$, where σ is the sub-Gaussian parameter of $\mathbf{Z}$, which is upper-bounded by $\|\mathbf{\Sigma}^{-1/2}\| = \lambda_{\min}(\mathbf{\Sigma})$, and $n_t = \sum_{s=1}^{t-1}|\Omega_s^{ind}|$, represents the number of pairwise observations so far. Thus, we can rewrite the above inequality which holds with probability $1-\delta_2$ as $\lambda_{\min}(\mathbf{D}) \geq n_t - \lambda_{\min}^{-1}(\mathbf{\Sigma})(C_1 n_t + l\sqrt{n_t})$, and:

$$\begin{aligned}\lambda_{\min}(\mathbf{A}_t^{\bar{neu}} - \lambda\mathbf{I}) &= \min_{x\in\mathbb{B}^p} x^\top(\mathbf{A}_t^{\bar{neu}} - \lambda\mathbf{I})x = \min_{x\in\mathbb{B}^p} x^\top \mathbf{\Sigma}^{1/2}\mathbf{D}\mathbf{\Sigma}^{1/2}x \\ &\geq \lambda_{\min}(\mathbf{D}) \min_{x\in\mathbb{B}^p} x^\top\Sigma x = \lambda_{\min}(\mathbf{D})\lambda_{\min}(\mathbf{\Sigma}) \geq \lambda_{\min}(\mathbf{\Sigma})n_t - C_1 n_t - C_2\sqrt{n_t\log(1/\delta_2)}\end{aligned}$$

Under event E_t, based on the definition of ω_t^c, we know that for any document i and j at round t, $(i,j) \notin \omega_t$ if and only if $\sigma(f_{ij}^t) - \alpha_t\|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{neu-1}} - \epsilon(m) \leq 1/2$ and $\sigma(f_{ji}^t) - \alpha_t\|\mathbf{g}_{ji}^t/\sqrt{m}\|_{\mathbf{A}_t^{neu-1}} - \epsilon(m) \leq 1/2$.

For a logistic function, we know that $\sigma(s) = 1 - \sigma(-s)$. Therefore, according to Lemma 2.1.6, let CB_{ij}^t denote $\alpha_t\|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{neu-1}} + \epsilon(m)$, we can conclude that $(i,j) \notin \omega_t$ if and only if $|\sigma(f_{ij}^t) - 1/2| \leq CB_{ij}^t$; and accordingly, $(i,j) \in \omega_t$, when $|\sigma(f_{ij}^t) - 1/2| > CB_{ij}^t$.

According to the discussion above, at round t, the probability that the estimated preference between document i and j to be in an uncertain rank order, i.e., $(i,j) \notin \omega_t$, can be upper bounded by:

$$\begin{aligned}\mathbb{P}\big((i,j)\notin\omega_t\big) =& \mathbb{P}\big(|\sigma(f_{ij}^t) - 1/2| \leq CB_{ij}^t\big) \leq \mathbb{P}\left(\big||\sigma(f_{ij}^t) - \sigma(h_{ij}^t)| - |\sigma(h_{ij}^t) - 1/2|\big| \leq CB_{ij}^t\right) \\ \leq& \mathbb{P}\left(|\sigma(h_{ij}^t) - 1/2| - |\sigma(f_{ij}^t) - \sigma(h_{ij}^t)| \leq CB_{ij}^t\right) \leq \mathbb{P}\left(\Delta_{\min} - |\sigma(f_{ij}^t) - \sigma(h_{ij}^t)| \leq CB_{ij}^t\right),\end{aligned}$$

where the first inequality is based on the reverse triangle inequality. The last inequality is based on the definition of $\Delta_{\min}$. Based on Lemma 2.1.6, the above probability can be further bounded by

$$\begin{aligned}&\mathbb{P}\left(\Delta_{\min} - |\sigma(f_{ij}^t) - \sigma(h_{ij}^t)| \leq CB_{ij}^t\right) \\ =&\mathbb{P}\left(|\sigma(f_{ij}^t) - \sigma(h_{ij}^t)| \geq \Delta_{\min} - \alpha_t\|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{neu-1}} - \epsilon(m)\right) \\ \leq&\mathbb{P}\left(\frac{2k_\mu}{c_\mu}||\mathbf{g}_{ij}^t/\sqrt{m}||_{\mathbf{A}_t^{neu-1}}\left(\|\mathbf{W}_t\|_{\mathbf{A}_t^{neu-1}} + \sqrt{\lambda}S\right) \geq \Delta_{\min} - \alpha_t\|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{neu-1}} - 2\epsilon(m)\right) \\ \leq&\mathbb{P}\left(\|\mathbf{W}_t\|_{\mathbf{A}_t^{neu-1}} \geq \frac{c_\mu(\Delta_{\min} - 2\epsilon(m))}{2k_\mu||\mathbf{g}_{ij}^t/\sqrt{m}||_{\mathbf{A}_t^{lin-1}}} - \left(\sqrt{\nu^2\log\frac{\det(\mathbf{A}_t^{neu})}{\delta_1^2\det(\lambda\mathbf{I})}} + 2\sqrt{\lambda}S\right)\right).\end{aligned}$$

where $\mathbf{W}_t = \sum_{s=1}^t\sum_{(i',j')\in\Omega_s^{ind}} \xi_{i'j'}^s \mathbf{g}_{i'j'}^s$.

For the right-hand side, we know that $\lambda_{\min}(\mathbf{A}_t^{neu}) \geq \lambda_{\min}(\mathbf{A}_t^{\bar{neu}}) + \|\mathbf{A}_t^{neu} - \mathbf{A}_t^{\bar{neu}}\| \geq \lambda_{\min}(\mathbf{A}_t^{\bar{neu}} - \lambda\mathbf{I}) + \lambda + \|\mathbf{A}_t^{neu} - \mathbf{A}_t^{\bar{neu}}\|$. With some positive constants $\{C_i^u\}_{i=1}^5$, for $t \geq t' = \left(C_1^u + C_2^u\sqrt{\log(1/\delta_2)} + C_3^u V_{\max}\right)^2 + C_4^u\log(1/\delta_1) + C_5^u$, as $n_t > t$, we have $n_t - \sqrt{n_t}\left(C_1^u + C_2^u\sqrt{\log(1/\delta_2)} + C_3^u V_{\max}\right) > C_4^u\log(1/\delta_1) + C_5^u$.

Hence, we have the following inequalities,

$$
\begin{aligned}
&\left(\frac{c_\mu(\Delta_{\min}-2\epsilon(m))}{2k_\mu\|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{lin-1}}}\right)^2-\left(\sqrt{\nu^2\log\frac{\det(\mathbf{A}_t^{neu})}{\delta_1^2\det(\lambda\mathbf{I})}}+2\sqrt{\lambda}S\right)^2\\
\geq&\lambda_{\min}(\mathbf{A}_t^{neu})c_\mu^2(\Delta_{\min}-2\epsilon(m))^2/(4C_3^zk_\mu^2L)-\nu^2\log(\det(\mathbf{A}_t^{neu})/(\delta_1^2\det(\lambda\mathbf{I})))\\
&-4\lambda S^2-4\sqrt{\lambda}S\nu\sqrt{\log(\det(\mathbf{A}_t^{neu})/\det\lambda\mathbf{I})+\log(1/\delta_1^2)}\\
\geq&(\lambda_{\min}(\mathbf{A}_t^{\bar{n}eu}-\lambda\mathbf{I})+\lambda+\|\mathbf{A}_t^{neu}-\mathbf{A}_t^{\bar{n}eu}\|)c_\mu^2(\Delta_{\min}-2\epsilon(m))^2/(4C_3^zk_\mu^2L)\\
&-(4\sqrt{\lambda}S\nu+\nu^2)\left(\log(\det(\mathbf{A}_t^{neu})/\det(\lambda\mathbf{I}))+\log(1/\delta_1^2)\right)-4\lambda S^2\\
\geq&\lambda_{\min}(\mathbf{\Sigma})(n_t-\sqrt{n_t}\big(C_1^u+C_2^u\sqrt{\log(1/\delta_2)}+C_3^uV_{\max}\big)-C_4^u\log(1/\delta_1)+C_5^u)\geq 0
\end{aligned}
$$

with corresponding positive constants $\{C_i^u\}_{i=1}^5$. Therefore, the probability could be upper bounded:

$$
\begin{aligned}
&\mathbb{P}\left(\Delta_{\min}-|\sigma(f_{ij}^t)-\sigma(h_{ij}^t)|\leq CB_{ij}^t\right)\\
\leq&\mathbb{P}\left(\|\mathbf{W}_t\|^2_{\mathbf{A}_t^{neu-1}}\geq\left(\frac{c_\mu(\Delta_{\min}-2\epsilon(m))}{2k_\mu\|\mathbf{g}_{ij}^t/\sqrt{m}\|_{\mathbf{A}_t^{lin-1}}}-\left(\sqrt{\nu^2\log\frac{\det(\mathbf{A}_t^{neu})}{\delta_1\det(\lambda\mathbf{I})}}+2\sqrt{\lambda}S\right)\right)^2\right)\\
\leq&\mathbb{P}\left(\|\mathbf{W}_t\|^2_{\mathbf{A}_t^{neu-1}}\geq\frac{c_\mu^2(\Delta_{\min}-2\epsilon(m))^2}{4k_\mu^2\|\mathbf{g}_{ij}^t/\sqrt{m}\|^{-1}_{\mathbf{A}_t^{neu}}}+\nu^2\log(\frac{\det(\mathbf{A}_t^{neu})}{\delta_1^2\det(\lambda\mathbf{I})})\right)\\
\leq&\mathbb{P}\left(\|\mathbf{W}_t\|^2_{\mathbf{A}_t^{neu-1}}\geq2\nu^2\log\left(\exp\left(\frac{c_\mu^2(\Delta_{\min}-2\epsilon(m))^2}{8\nu^2k_\mu^2\|\mathbf{g}_{ij}^t/\sqrt{m}\|^{-1}_{\mathbf{A}_t^{neu}}}\right)\cdot\frac{\det(\mathbf{A}_t^{neu})}{\delta_1^2\det(\lambda\mathbf{I})}\right)\right)\\
\leq&\delta_1\cdot\exp^{-1}\left(\frac{c_\mu^2(\Delta_{\min}-2\epsilon(m))^2}{8\nu^2k_\mu^2\|\mathbf{g}_{ij}^t/\sqrt{m}\|^2_{\mathbf{A}_t^{neu-1}}}\right)\leq C_6^u\log(1/\delta_1)\frac{\|\mathbf{g}_{ij}^t/\sqrt{m}\|^2_{\mathbf{A}_t^{neu-1}}}{(\Delta_{\min}-2\epsilon(m))^2},
\end{aligned}
$$

with an additional positive constant C_6^u. This completes the proof. □

B.6 Proof of Theorem 2.1.17

Lemma B.10. There exist positive constants $\{C_i\}_{i=1}^2$ such that for any $\delta\in(0,1)$, if $\eta\leq\bar{C}_1(TmL+m\lambda)^{-1}$ and $m\geq\bar{C}_2\max\left\{T^7\lambda^{-7}L^{21}(\log m)^3,N^6L^6(\log(n_TL^2/\delta))^{3/2}\right\}$, then with probability at least $1-\delta$, we have

$$
\sum_{t=1}^T\sum_{(i',j')\in\Omega_t^{ind}}\|\mathbf{g}_{i'j'}^t/\sqrt{m}\|_{\mathbf{A}_t^{neu-1}}\leq 2\log\frac{\det\mathbf{A}_T}{\det\lambda\mathbf{I}}\leq\widetilde{d}\log(1+TV_{\max}^2/\lambda)+1
$$

where $\widetilde{d}$ is defined as the effective dimension of $\mathbf{H}$.

Proof of Theorem 2.1.17. With δ_1 and δ_2 defined in the previous lemmas, we have with probability at least $1-\delta_1$, the T-step regret is upper bounded as:

$$
R_T=R_{t'}+R_{T-t'}\leq t'*V_{\max}^2+(T-t')\delta_2V_{\max^2}+(1-\delta_2)\sum_{t=t'}^T r_t \tag{B.9}
$$

When event E_t and the event defined in Lemma 2.1.6 both occur, the instantaneous regret at round t is bounded by $r_t=\mathbb{E}\big[K(\tau_s,\tau_s^*)\big]\leq\mathbb{E}[U_t]$, where U_t denotes the number of uncertain rank orders under the ranker at round t. As the ranked list is generated by topological sort on the certain rank orders, the random shuffling only happens between the documents that are in uncertain rank orders, which induce regret in the proposed ranked list. In each round of result serving, as the model $\boldsymbol{\theta}_t$ would not change until the next round, the expected

number of uncertain rank orders can be estimated by summing the uncertain probabilities over all possible pairwise comparisons under the current query q_t, e.g., $\mathbb{E}[U_t] = 1/2 \sum_{(i,j)\in\Psi_t} \mathbb{P}((i,j) \notin \omega_t)$.

Based on Lemma 2.1.15, the cumulative number of mis-ordered pairs can be bounded by the probability of observing uncertain rank orders in each round, which shrinks with more observations become available over time,

$$\begin{aligned}\mathbb{E}\big[\sum_{s=t'}^{T} U_t\big] \leq& \mathbb{E}\big[1/2 \sum_{s=t'}^{T} \sum_{(i',j')\in\Psi_s} \mathbb{P}((i',j') \notin \omega_t)\big] \\ \leq& \mathbb{E}\big[1/2 \sum_{s=t'}^{T} \sum_{(i',k')\in\Psi_s} C_6^u \log(1/\delta_1) \|\mathbf{g}_{i'j'}^t\|^2_{\mathbf{A}_t^{neu-1}} / (\Delta_{\min} - 2\epsilon(m))^2\big].\end{aligned}$$

Because $\mathbf{A}_t^{neu}$ only contains information of observed document pairs so far, our algorithm guarantees the number of mis-ordered pairs among the observed documents in the above inequality is upper bounded. To reason about the number of mis-ordered pairs in those unobserved documents (i.e., from o_t to V_t for each query q_t), we leverage the constant p^*, which is defined as the minimal probability that all documents in a query are examined over time,

$$\begin{aligned}\mathbb{E}\big[\sum_{t=t'} \sum_{(i',j')\in\Psi_t} \|\mathbf{g}_{i'j'}^t/\sqrt{m}\|_{\mathbf{A}_t^{lin-1}}\big] =& \mathbb{E}\big[\sum_{t=t'} \sum_{(i',j')\in\Psi_t} \|\mathbf{g}_{i'j'}^t/\sqrt{m}\|_{\mathbf{A}_t^{lin-1}} \times \mathbb{E}\big[p_t^{-1}\mathbf{1}\{o_t = V_t\}\big]\big] \\ \leq& p^{*-1}\mathbb{E}\big[\sum_{t=t'} \sum_{(i',j')\in\Psi_t} \|\mathbf{g}_{i'j'}^t/\sqrt{m}\|_{\mathbf{A}_t^{lin-1}} \mathbf{1}\{o_t = V_t\}\big]\end{aligned}$$

Besides, we only use the independent pairs, Ω_t^{ind} to update the model and the corresponding $\mathbf{A}_t^{neu}$ matrix. Therefore, to bound the regret, we rewrite the above equation as:

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=t'}^{T} \sum_{(i',j')\in\Psi_t} \|\mathbf{g}_{i'j'}^t/\sqrt{m}\|^2_{\mathbf{A}_t^{neu-1}}\right] \qquad \text{(B.10)} \\ =&\mathbb{E}\left[\sum_{t=t'}^{T} \sum_{(i',j')\in\Omega_t^{ind}} \left(\|\mathbf{g}_{i'j'}^t/\sqrt{m}\|^2_{\mathbf{A}_t^{neu-1}} + \sum_{k\in[V_t]\setminus\{i',j'\}} \|\mathbf{g}_{i'k}^t/\sqrt{m}\|^2_{\mathbf{A}_t^{neu-1}} + \|\mathbf{g}_{j'k}^t/\sqrt{m}\|^2_{\mathbf{A}_t^{neu-1}}\right)\right] \\ =&\mathbb{E}\left[\sum_{t=t'}^{T} \sum_{(i',j')\in\Omega_t^{ind}} \left((V_t-1)\|\mathbf{g}_{i'j'}^t/\sqrt{m}\|^2_{\mathbf{A}_s^{-1}} + \sum_{k\in[V_t]\setminus\{i',j'\}} (2/m){\mathbf{g}_{i'k}^t}^\top \mathbf{A}_t^{neu-1}\mathbf{g}_{j'k}^t\right)\right]\end{aligned}$$

For the second term, it can be bounded as:

$$\sum_{t=t'}^{T} \sum_{(i',j')\in\Omega_t^{ind}} \sum_{k\in[V_t]\setminus\{i',j'\}} (2/m){\mathbf{g}_{i'k}^t}^\top \mathbf{A}_t^{neu-1}\mathbf{g}_{j',k}^t \leq \sum_{t=t'}^{T} 2C_3^z(V_{\max}^2 - 2V_{\max})L^2/\lambda_{\min}(\mathbf{A}_t^{neu})$$

where the first inequality is due to Lemma B.7. According to the analysis of $\lambda_{\min}(\mathbf{A}_t^{neu})$ and $\lambda_{\min}(\bar{\mathbf{A}}_t^{neu})$, the convergence rate the above upper bound is faster than the self-normalized term in Eq B.10. Hence, by chaining all the inequalities, we have with probability at least $1-\delta_1$, the regret satisfies,

$$\begin{aligned}R_T \leq& R' + (1-\delta_2)C_6^u \log(1/\delta_1)(w + V_{\max}(\widetilde{d}\log(1 + TV_{\max}^2/\lambda) + 1)/(\Delta_{\min} - 2\epsilon(m))^2 \\ \leq& R' + (C_1^r \log(1/\delta_1)\widetilde{d}\log(1 + TV_{\max}^2/\lambda) + C_2^r)(1-\delta_2)/p^*\end{aligned}$$

where $\{C_i^r\}_{i=1}^2$ are positive constants, $R' = t'V_{\max}^2 + (T-t')\delta_2 V_{\max}^2$. By choosing $\delta_1 = \delta_2 = 1/T$, the theorem shows that the expected regret is at most $R_T \leq O(\log^4(T))$. □

C Proofs of Theorems and Lemmas in Section 2.2

C.1 Proof of Lemma 2.2.1

Proof. According to the definition of $g_t(\boldsymbol{\theta})$ in Section 4, we have the following equation for the auxiliary solution $\bar{\boldsymbol{\theta}}_t$,

$$g_t(\bar{\boldsymbol{\theta}}_t) = \sum_{s=-d+1}^{t-1} \sum_{(i,j)\in\Omega_s} y_{ij}^s \mathbf{x}_{ij}^s$$

Then, to bound the deviation caused by the observation noise, we have the following derivations for any input $\mathbf{x}$

$$\begin{aligned}
&|\sigma(\mathbf{x}^\top \bar{\boldsymbol{\theta}}_t) - \sigma(\mathbf{x}^\top \boldsymbol{\theta}^*)| \le k_\mu |\mathbf{x}^\top (\bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)| \\
=&k_\mu |\mathbf{x}^\top \mathbf{G}_t^{-1} (g_t(\bar{\boldsymbol{\theta}}_t) - g_t(\boldsymbol{\theta}^*))| \le \frac{k_\mu}{c_\mu} |\mathbf{x}^\top \mathbf{A}_t^{-1} (g_t(\bar{\boldsymbol{\theta}}_t) - g_t(\boldsymbol{\theta}^*))| \\
\le&\frac{k_\mu}{c_\mu} \|\mathbf{x}\|_{\mathbf{A}_t^{-1}} \|g_t(\bar{\boldsymbol{\theta}}_t) - g_t(\boldsymbol{\theta}^*)\|_{\mathbf{A}_t^{-1}}
\end{aligned}$$

The first inequality is due to the Lipschitz continuity of the logistic function. As logistic function is continuously differentiable, the second equality is according to the Fundamental Theorem of Calculus, where $\mathbf{G}_t = \sum_{s=1}^{t-1} \sum_{(i,j)\in\Omega_s} \dot{\sigma}({\mathbf{x}_{ij}^s}^\top \boldsymbol{\theta}) \mathbf{x}_{ij}^s {\mathbf{x}_{ij}^s}^\top + \lambda \mathbf{I}$. In the third inequality, $\mathbf{A}_t = \sum_{s=1}^{t-1} \sum_{(i,j)\in\Omega_s} \mathbf{x}_{ij}^s {\mathbf{x}_{ij}^s}^\top + \lambda \mathbf{I}$. And this inequality holds as $\mathbf{G}_t \succ c_\mu \mathbf{A}_t$, with $c_\mu = \inf_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \dot{\sigma}(\boldsymbol{x}^\top \boldsymbol{\theta})$.

Next, we will bound $\|g_t(\bar{\boldsymbol{\theta}}_t) - g_t(\boldsymbol{\theta}^*)\|_{\mathbf{A}_t^{-1}}$.

$$\begin{aligned}
&\|g_t(\bar{\boldsymbol{\theta}}_t) - g_t(\boldsymbol{\theta}^*)\|_{\mathbf{A}_t^{-1}} \\
=&\| \sum_{s=-d+1}^{t-1} y_{ij}^s \mathbf{x}_{ij}^s - \sigma({\mathbf{x}_{ij}^s}^\top \boldsymbol{\theta}^*) \mathbf{x}_{ij}^s \|_{\mathbf{A}_t^{-1}} \\
\le&\| \sum_{s=-d+1}^{0} \sigma({\mathbf{x}_{ij}^s}^\top \boldsymbol{\theta}^*) \mathbf{x}_{ij}^s \|_{\mathbf{A}_t^{-1}} + \| \sum_{s=1}^{t-1} \epsilon_{ij}^s \mathbf{x}_{ij}^s \|_{\mathbf{A}_t^{-1}} \\
\le&d + R\sqrt{\log(\frac{\det(\mathbf{A}_t)}{\delta^2 \det(\lambda \mathbf{I})})}
\end{aligned}$$

The first equality is based on the definition of function g_t, and $\bar{\boldsymbol{\theta}}_t$. The last inequality is according to the self-normalized bound for martingales in [14]. This completes the proof. □

C.2 Proof of Lemma 2.2.3

Proof. According to Lemma 2.2.1, with probability at least $1-\delta$, event E_t defined in Section 4 occurs. Under event E_t, for document pair (i,j) satisfying $i \succ j$ for the given query (e.g., $\sigma({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}^*) > \frac{1}{2}$), we first analyze the probability that at least one estimate $\sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t^{(n)}) > \frac{1}{2}$ for $n \in [N]$, e.g., $\mathbb{P}(\max_{n\in[N]} \sigma({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t^{(n)})) > \frac{1}{2}$. For simplicity, in the following analysis, we use $\widehat{\sigma}_t^{(n)}$, $\bar{\sigma}_t$and σ^* to present $\sigma\left({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t^{(n)}\right)$, $\sigma\left({\mathbf{x}_{ij}^t}^\top \bar{\boldsymbol{\theta}}_t\right)$and $\sigma\left({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}^*\right)$ respectively.

$$\mathbb{P}(\max_{n\in[N]} \widehat{\sigma}_t^{(n)} \ge \frac{1}{2}) = 1 - \prod_{n=1}^{N} \mathbb{P}(\sigma_t^{(n)} < \frac{1}{2})$$

For any $n \in [N]$, we have the following bound for $\mathbb{P}(\sigma_t^{(n)} < \frac{1}{2})$.

$$\begin{aligned}
\mathbb{P}(\sigma_t^{(n)} < \frac{1}{2}) \leq & \mathbb{P}(\sigma_t^{(n)} < \sigma^*) \\
= & \mathbb{P}(\sigma_t^{(n)} - \bar{\sigma}_t < \sigma^* - \bar{\sigma}_t) \\
\leq & \mathbb{P}(\sigma_t^{(n)} - \bar{\sigma}_t < \alpha_t \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{-1}}) \\
\leq & \mathbb{P}(c_\mu {\mathbf{x}_{ij}^t}^\top (\widehat{\boldsymbol{\theta}}_t - \bar{\boldsymbol{\theta}}) < \alpha_t \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{-1}}) \\
\leq & \mathbb{P}(c_\mu {\mathbf{x}_{ij}^t}^\top \mathbf{G}_t^{-1}(g_t(\widehat{\boldsymbol{\theta}}_t) - g_t(\bar{\boldsymbol{\theta}})) < \alpha_t \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{-1}}) \\
\leq & \mathbb{P}(\frac{c_\mu}{k_\mu} {\mathbf{x}_{ij}^t}^\top \mathbf{A}_t^{-1}(g_t(\widehat{\boldsymbol{\theta}}_t) - g_t(\bar{\boldsymbol{\theta}})) < \alpha_t \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{-1}}) \\
= & \mathbb{P}(U(\mathbf{x}_{ij}^t) < \frac{k_\mu}{c_\mu} \alpha_t \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{-1}})
\end{aligned}$$

where $U(\mathbf{x}_{ij}^t) = {\mathbf{x}_{ij}^t}^\top \mathbf{A}_t^{-1}(g_t(\widehat{\boldsymbol{\theta}}_t) - g_t(\bar{\boldsymbol{\theta}}))$. According to the definition of $\bar{\boldsymbol{\theta}}_t$, we know that,

$$U(\mathbf{x}_{ij}^t) = {\mathbf{x}_{ij}^t}^\top \mathbf{A}_t^{-1} \sum_{s=-d+1}^{t-1} \gamma_{ij}^s \mathbf{x}_{ij}^s$$

It is easy to obtain that $\mathbb{E}[U(\mathbf{x}_{ij}^t)] = 0$ with $\mathbb{E}[\gamma] = 0$. And the variance of $U(\mathbf{x}_{ij}^t)$ is,

$$\begin{aligned}
Var[U(\mathbf{x}_{ij}^t)] = & \nu^2 {\mathbf{x}_{ij}^t}^\top \mathbf{A}_t (\sum_{s=-d+1}^{t-1} \mathbf{x}_{ij}^s {\mathbf{x}_{ij}^s}^\top) \mathbf{A}_t \mathbf{x}_{ij}^t \\
= & \nu^2 {\mathbf{x}_{ij}^t}^\top \mathbf{A}_t^{-1} (\lambda \mathbf{I} + \sum_{s=1}^{t-1} \mathbf{x}_{ij}^s {\mathbf{x}_{ij}^s}^\top) \mathbf{A}_t^{-1} \mathbf{x}_{ij}^t \\
= & \nu^2 \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{-1}}^2
\end{aligned}$$

Therefore, we have $U(\mathbf{x}_{ij}^t) \sim \mathcal{N}(0, \nu^2 \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{-1}}^2)$, and the probability can be upper bounded as,

$$\begin{aligned}
\mathbb{P}(U(\mathbf{x}_{ij}^t) < \frac{k_\mu}{c_\mu} \alpha_t \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{-1}}) = & 1 - \mathbb{P}(U(\mathbf{x}_{ij}^t) > \frac{k_\mu}{c_\mu} \alpha_t \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{-1}}) \\
\leq & 1 - \frac{\exp(-\beta^2)}{4\sqrt{\pi}\beta}
\end{aligned}$$

where $\beta = \frac{k_\mu \alpha_t}{c_\mu \nu}$. By chaining all the inequalities, we have that with $N \geq \log \delta / \log(1 - \exp(-\beta^2)/(4\sqrt{\pi}\beta))$, with probability at least $1 - \delta$, $\mathbb{P}(\max_{n \in [N]} \widehat{\sigma}_t^{(n)} \geq \frac{1}{2})$.

Therefore, under event E_t, with $N \geq \log \delta / \log(1 - \exp(-\beta^2)/(4\sqrt{\pi}\beta))$, based on the definition of ω_t^u in Section 3.2, we know that for document i and j at round t, $(i, j) \in \omega_t^c$ if and only if $\min_{n \in [N]} \sigma\left({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t^{(n)}\right) > \frac{1}{2}$. For simplicity, in the following analysis, we use $\widehat{\sigma}_t^{(n)}$, $\bar{\sigma}_t$and σ^* to present $\sigma\left({\mathbf{x}_{ij}^t}^\top \widehat{\boldsymbol{\theta}}_t^{(n)}\right)$, $\sigma\left({\mathbf{x}_{ij}^t}^\top \bar{\boldsymbol{\theta}}_t\right)$and $\sigma\left({\mathbf{x}_{ij}^t}^\top \boldsymbol{\theta}^*\right)$ respectively. Then the probability of being uncertain can be bounded as,

$$\begin{aligned}
\mathbb{P}\big((i, j) \in \omega_t^u\big) = & 1 - \mathbb{P}\big(\min_{n \in [N]} \widehat{\sigma}_t^{(n)} > 1/2\big) \\
= & 1 - \prod_{n=1}^{N} \mathbb{P}\big(\widehat{\sigma}_t^{(n)} > 1/2\big) = 1 - \big(\mathbb{P}\big(\widehat{\sigma}_t^{(n)} > 1/2\big)\big)^N
\end{aligned}$$

Based on the definition of $\Delta_{\min}$, $\sigma^* - 1/2 \geq \Delta_{\min}$ and $\sigma^* - \bar{\sigma}_t \leq CB$. And according to the random

matrix theory and Lemma 2 in [26], when $t > t'$, $\Delta_{\min} - CB > 0$ which can be viewed as a constant $c\Delta_{\min}$. Therefore, we have the following inequalities,

$$\begin{aligned}
&\mathbb{P}\big(\widehat{\sigma}_t^{(m)} > 1/2\big) = \mathbb{P}\big(\widehat{\sigma}_t^{(m)} - \sigma^* + \sigma^* - \bar{\sigma}_t + \bar{\sigma}_t > 1/2\big) \\
\geq &\mathbb{P}\big(\bar{\sigma}_t - \widehat{\sigma}_t^{(m)} < \Delta_{\min} - CB\big) = 1 - \frac{1}{2}\mathbb{P}\big(|\bar{\sigma}_t - \widehat{\sigma}_t^{(m)}| \geq \Delta_{\min} - CB\big) \\
\geq &1 - \frac{1}{2}\mathbb{P}\big(k_\mu |U(x)| \geq \Delta_{\min} - CB\big) \geq 1 - \exp(-\frac{c_\mu^2 c^2 \Delta_{\min}^2}{2k_\mu^2 \nu^2 \|x\|^2_{\boldsymbol{A}_t^{-1}}})
\end{aligned}$$

Let $B = \frac{c_\mu^2 c^2 \Delta_{\min}^2}{2k_\mu^2 \nu^2 \|x\|^2_{\boldsymbol{A}_t^{-1}}}$, we have the probability of being uncertain rank order upper bounded as,

$$\begin{aligned}
\mathbb{P}\big((i,j) \in \omega_t^u\big) \leq &1 - \big(1 - \exp(-B)\big)\big)^N \\
= &1 - \exp N \log(1 - \exp(-B)) \\
\leq &- N \log(1 - \exp(-B)) \leq -N \log \exp(-1/B))
\end{aligned}$$

Therefore, $\mathbb{P}\big((i,j) \in \omega_t^u\big) \leq \min\Big\{1, \frac{2Nk_\mu^2 \nu^2 \|x_{i,j}\|^2_{\boldsymbol{A}_t^{-1}}}{c_\mu^2 c^2 \Delta_{\min}^2}\Big\}$. This completes the proof. □

C.3 Proof of Theorem 2.2.4

Proof. Once the certain and uncertain rank orders are determined, our proposed model will generate the ranked list by topological sort with respect to the certain rank orders. Therefore, the regret only comes from the uncertain rank orders. In each round of result serving, as the model $\boldsymbol{\theta}_t$ would not change until the next round, the expected number of uncertain rank orders, U_t, can be estimated by summing the uncertain probabilities over all possible pairwise comparisons under the current query q_t, e.g., $\mathbb{E}[U_t] = \frac{V_t(V_t-1)}{2}\mathbb{P}((i,j) \in \omega_t^u)$.

Based on Lemma 2.2.3, the cumulative number of mis-ordered pairs can be bounded by the probability of observing uncertain rank orders in each round, which shrinks with more observations become available over time,

$$\mathbb{E}\big[\sum\nolimits_{s=t'}^{T} U_t\big] \leq \mathbb{E}\big[\frac{1}{2}\sum\nolimits_{s=t'}^{T}\sum\nolimits_{(i,j)\in[V_t]^2} \frac{2Nk_\mu^2\nu^2\|\mathbf{x}_{ij}^s\|^2_{\boldsymbol{A}_s^{-1}}}{c_\mu^2 c^2 \Delta_{\min}^2}\big].$$

As $\mathbf{A}_t$ only contains information of observed document pairs so far, the number of mis-ordered pairs among the observed documents is guaranteed to be upper bounded. To reason about the number of mis-ordered pairs in those unobserved documents (i.e., from o_t to V_t for each query q_t), we leverage the constant p^*, which is defined as the minimal probability that all documents in a query are examined over time,

$$\begin{aligned}
&\mathbb{E}\big[\sum\nolimits_{t=t'}^{T}\sum\nolimits_{(i,j)\in[V_t]^2} \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{-1}}\big] \\
= &\mathbb{E}\big[\sum\nolimits_{t=t'}^{T}\sum\nolimits_{(i,j)\in[V_t]^2} \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{-1}} \times \mathbb{E}\big[p_t^{-1}\mathbf{1}\{o_t = V_t\}\big]\big] \\
\leq &p^{*-1}\mathbb{E}\big[\sum\nolimits_{t=t'}^{T}\sum\nolimits_{(i,j)\in[V_t]^2} \|\mathbf{x}_{ij}^t\|_{\mathbf{A}_t^{-1}}\mathbf{1}\{o_t = V_t\}\big]
\end{aligned}$$

Besides, we only use the independent pairs Ω_t to update the model and the corresponding $\mathbf{A}_t$ matrix. Therefore, to bound the regret, the pairs can be divided into two parts based on whether they are belonging to the observed

set Ω_t. Then, we have the following inequalities,

$$\begin{aligned}&\sum_{s=t'}^{T}\sum_{(i,j)\in\Omega_s}\sum_{k\in[V_t]\setminus\{i,j\}}2{\boldsymbol{x}_{ik}^s}^\top\boldsymbol{A}_s^{-1}\boldsymbol{x}_{jk}^s\\ \leq&\sum_{s=t'}^{T}(V_{\max}^2-2V_{\max})P^2/\lambda_{\min}(\boldsymbol{A}_s).\end{aligned}$$

And for the pairs belonging to the $\{\Omega_s\}_{s=1}^T$, based on Lemma 10 and Lemma 11 in [14], we have,

$$\sum_{s=t'}^{T}\sum_{(i,j)\in\Omega_s}(V_t-1)\|\mathbf{x}_{ij}^s\|_{\boldsymbol{A}_s^{-1}}^2\leq 2dV_{\max}\log(1+\frac{o_{\max}TP^2}{2d\lambda})$$

Then, chaining all the inequalities, we have when event E_t happens, the regret can be upper bounded as,

$$R_T\leq R'+\sum_{s=t'}^{T}r_s\leq R'+\frac{1}{p^*}2dV_{\max}C\log(1+\frac{o_{\max}TP^2}{2d\lambda})$$

where $C=2Nk_\mu^2\nu^2/c_\mu^2c^2\Delta_{\min}^2$, $R'=t'V_{\max}$, with t' defined in Lemma 2.2.3. This completes the proof. □

D Proofs of Lemmas in Appendix B

In this section, we provide the detailed proofs of Lemma B.9 and Lemma B.8 in Section B. For the technical lemmas, interested readers can refer to the original paper to [15] for more details.

We need the following technical lemma adopted from [15].

Lemma D.1 (Lemma 5.1, [15])**.** Let $\mathbf{G}=[\mathbf{g}(\mathbf{x}_1;\boldsymbol{\theta}_0),\ldots,\mathbf{g}(\mathbf{x}_{n_T};\boldsymbol{\theta}_0)]/\sqrt{m}\in\mathbb{R}^{p\times n_T}$. Let $\mathbf{H}$ be the NTK matrix as defined in Definition A.1. For any $\delta\in(0,1)$, if

$$m=\Omega\bigg(\frac{L^6\log(n_TL/\delta)}{\epsilon^4}\bigg),$$

then with probability at least $1-\delta$, we have $\|\mathbf{G}^\top\mathbf{G}-\mathbf{H}\|_F\leq n\epsilon$.

D.1 Proof of Lemma B.9

In this section, we will provide the detailed proof of Lemma B.9. First, assume that until round t, there are in total n_t^P observed document pairs, e.g., $\sum_{s=1}^{t-1}|\Omega_s^{ind}|=n_t^P\leq V_{\max}t$, where $|\cdot|$ represents the cardinality of the designated set, and $V_{\max}$ is the maximum number of document pairs that can be observed given query q across all queries. For simplicity, we will re-index all the observed pairs until round t from 1 to n_t in the following analysis.

Then, for round t, define the following quantities,

$$\mathbf{J}^{(j)}=\Big(\mathbf{g}(\mathbf{x}_{1,1};\boldsymbol{\theta}^{(j)})-\mathbf{g}(\mathbf{x}_{1,2};\boldsymbol{\theta}^{(j)},\ldots,\mathbf{g}(\mathbf{x}_{n_t^P,1};\boldsymbol{\theta}^{(j)})-\mathbf{g}(\mathbf{x}_{n_t^P,2};\boldsymbol{\theta}^{(j)})\Big)\in\mathbb{R}^{p\times n_t^P}\tag{D.1}$$

$$\mathbf{H}^{(j)}=\mathbf{J}^{(j)\top}\mathbf{J}^{(j)}\in\mathbb{R}^{n_t^P\times n_t^P}\tag{D.2}$$

$$\mathbf{f}^{(j)}=\Big(f(\mathbf{x}_{1,1};\boldsymbol{\theta}^{(j)})-f(\mathbf{x}_{1,2};\boldsymbol{\theta}^{(j)}),\ldots,f(\mathbf{x}_{n_t^P,1};\boldsymbol{\theta}^{(j)})-f(\mathbf{x}_{n_t^P,2};\boldsymbol{\theta}^{(j)})\Big)^\top\in\mathbb{R}^{n_t^P\times 1}\tag{D.3}$$

$$\mathbf{p}^{(j)}=\Big(\sigma(f(\mathbf{x}_{1,1};\boldsymbol{\theta}^{(j)})-f(\mathbf{x}_{1,2};\boldsymbol{\theta}^{(j)})),\ldots,\sigma(f(\mathbf{x}_{n_t^P,1};\boldsymbol{\theta}^{(j)})-f(\mathbf{x}_{n_t^P,2};\boldsymbol{\theta}^{(j)})\Big)^\top\in\mathbb{R}^{n_t^P\times 1}\tag{D.4}$$

$$\mathbf{y}=\big(y_1,\ldots y_{n_t^P}\big)^\top\in\mathbb{R}^{n_t^P\times 1}\tag{D.5}$$

According to the loss function defined in Eq (2.1.2), we have the update rule of $\boldsymbol{\theta}^{(j)}$ as follows:

$$\boldsymbol{\theta}^{(j+1)} = \boldsymbol{\theta}^{(j)} - \eta[\mathbf{J}^{(j)}(\mathbf{p}^{(j)} - \mathbf{y}) + m\lambda(\boldsymbol{\theta}^{(j)} - \boldsymbol{\theta}^{(0)})] \tag{D.6}$$

Besides, we have the following auxiliary sequence $\{\widetilde{\boldsymbol{\theta}}^{(k)}\}$,

$$\widetilde{\boldsymbol{\theta}}^{(0)} = \boldsymbol{\theta}^{(0)}, \widetilde{\boldsymbol{\theta}}^{(j+1)} = \widetilde{\boldsymbol{\theta}}^{(j)} - \eta[\mathbf{J}^{(0)}(\sigma(\mathbf{J}^{(0)\top}(\widetilde{\boldsymbol{\theta}}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(0)})) - \mathbf{y}) + m\lambda(\widetilde{\boldsymbol{\theta}}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(0)})].$$

Next lemma provides perturbation bounds for $\mathbf{J}^{(j)}, \mathbf{H}^{(j)}$ and $\|\mathbf{f}^{(j+1)} - \mathbf{f}^{(j)} - [\mathbf{J}^{(j)}]^\top(\boldsymbol{\theta}^{(j+1)} - \boldsymbol{\theta}^{(j)})\|_2$.

Lemma D.2. There exist constants $\{\widehat{C}_i\}_{i=1}^6 > 0$ such that for any $\delta > 0$, if τ satisfies that

$$\widehat{C}_1 m^{-3/2} L^{-3/2} [\log(n_T L^2/\delta)]^{3/2} \leq \tau \leq \widehat{C}_2 L^{-6} [\log m]^{-3/2},$$

then with probability at least $1 - \delta$, for any $j, s \in [J]$, $\|\boldsymbol{\theta}^{(j)} - \boldsymbol{\theta}^{(0)}\|_2 \leq \tau$ and $\|\boldsymbol{\theta}^{(s)} - \boldsymbol{\theta}^{(0)}\|_2 \leq \tau$, we have the following inequalities,

$$\left\|\mathbf{J}^{(j)}\right\|_F \leq \widehat{C}_4 \sqrt{n_t^P m L}, \tag{D.7}$$

$$\|\mathbf{J}^{(j)} - \mathbf{J}^{(0)}\|_F \leq \widehat{C}_5 \sqrt{n_t^P m \log m} \tau^{1/3} L^{7/2}, \tag{D.8}$$

$$\left\|\mathbf{f}^{(s)} - \mathbf{f}^{(j)} - [\mathbf{J}^{(j)}]^\top(\boldsymbol{\theta}^{(s)} - \boldsymbol{\theta}^{(j)})\right\|_2 \leq \widehat{C}_6 \tau^{4/3} L^3 \sqrt{n_t^P m \log m}, \tag{D.9}$$

$$\|\mathbf{y}\|_2 \leq \sqrt{n_t^P}. \tag{D.10}$$

Lemma D.3. There exist constants $\{\widetilde{C}_i\}_{i=1}^4 > 0$ such that for any $\delta > 0$, if τ, η satisfy that

$$\begin{aligned}
&\widetilde{C}_1 m^{-3/2} L^{-3/2} [\log(n_T L^2/\delta)]^{3/2} \leq \tau \leq \widetilde{C}_2 L^{-6} [\log m]^{-3/2}, , \\
&\eta \leq \widetilde{C}_3 (m\lambda + n_t^P m L)^{-1}, \tau^{8/3} \leq \widetilde{C}_4 m (\lambda\eta)^2 L^{-6} (n_t^P)^{-1} (\log m)^{-1},
\end{aligned}$$

then with probability at least $1 - \delta$, for any $j \in [J]$, $\|\boldsymbol{\theta}^{(j)} - \boldsymbol{\theta}^{(0)}\|_2 \leq \tau$, we have $\|\mathbf{p}^{(j)} - \mathbf{y}\|_2 \leq 2\sqrt{n_t^P}$.

Next lemma gives an upper bound of the distance between auxiliary sequence $\|\widetilde{\boldsymbol{\theta}}^{(j)} - \boldsymbol{\theta}^{(0)}\|_2$.

Lemma D.4. There exist constants $\{C_i\}_{i=1}^3 > 0$ such that for any $\delta \in (0, 1)$, if τ, η satisfy that

$$C_1 m^{-3/2} L^{-3/2} [\log(n_t^P L^2/\delta)]^{3/2} \leq \tau \leq C_2 L^{-6} [\log m]^{-3/2}, \eta \leq C_3 (n_t^P m L + m\lambda)^{-1},$$

then with probability at least $1 - \delta$, we have that for any $j \in [J]$,

$$\left\|\widetilde{\boldsymbol{\theta}}^{(j)} - \boldsymbol{\theta}^{(0)}\right\|_2 \leq \sqrt{2n_t^P/(m\lambda)}, \text{and } \left\|\widetilde{\boldsymbol{\theta}}^{(j)} - \boldsymbol{\theta}^{(0)} - \widehat{\mathbf{w}}_t\right\|_2 \leq (1 - \eta m\lambda)^{j/2} \sqrt{n_t^P/(m\lambda)}.$$

With above lemmas, we prove Lemma B.9 as follows.

Proof of Lemma B.9. Set $\tau = \sqrt{2n_t^P/(m\lambda)}$. First we assume that $\|\boldsymbol{\theta}^{(j)} - \boldsymbol{\theta}^{(0)}\|_2 \leq \tau$ for all $0 \leq j \leq J$. Then with this assumption and the choice of m, τ, we have that Lemma D.2, D.3 and D.4 hold. Then we have

$$
\begin{aligned}
&\big\|\boldsymbol{\theta}^{(j+1)} - \widetilde{\boldsymbol{\theta}}^{(j+1)}\big\|_2 \\
=&\big\|\boldsymbol{\theta}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(j)} - \eta\mathbf{J}^{(j)}(\mathbf{p}^{(j)} - \mathbf{y}) - \eta m\lambda(\boldsymbol{\theta}^{(j)} - \boldsymbol{\theta}^{(0)}) + \eta\mathbf{J}^{(0)}(\sigma(\mathbf{J}^{(0)\top}(\widetilde{\boldsymbol{\theta}}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(0)})) - \mathbf{y}) + \eta m\lambda(\widetilde{\boldsymbol{\theta}}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(0)})\big\|_2 \\
=&\big\|(1-\eta m\lambda)(\boldsymbol{\theta}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(j)}) - \eta(\mathbf{J}^{(j)} - \mathbf{J}^{(0)})(\mathbf{p}^{(j)} - \mathbf{y}) - \eta\mathbf{J}^{(0)}(\mathbf{p}^{(j)} - \sigma(\mathbf{J}^{(0)\top}(\widetilde{\boldsymbol{\theta}}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(0)})))\big\|_2 \\
\leq&\big\|(1-\eta m\lambda)(\boldsymbol{\theta}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(j)})\big\|_2 + \eta\big\|(\mathbf{J}^{(j)} - \mathbf{J}^{(0)})(\mathbf{p}^{(j)} - \mathbf{y})\big\|_2 + \eta\big\|\mathbf{J}^{(0)}\big\|_2\big\|\mathbf{p}^{(j)} - \sigma(\mathbf{J}^{(0)\top}(\widetilde{\boldsymbol{\theta}}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(0)}))\big\|_2 \\
\leq&\big\|(1-\eta m\lambda)(\boldsymbol{\theta}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(j)})\big\|_2 + \eta\big\|(\mathbf{J}^{(j)} - \mathbf{J}^{(0)})(\mathbf{p}^{(j)} - \mathbf{y})\big\|_2 + k_\mu\eta\big\|\mathbf{J}^{(0)}\big\|_2\big\|\mathbf{f}^{(j)} - \mathbf{J}^{(0)\top}(\widetilde{\boldsymbol{\theta}}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(0)})\big\|_2 \\
\leq&\big\|(\mathbf{I} - \eta(m\lambda\mathbf{I} + k_\mu\mathbf{H}^{(0)}))\big\|_2\big\|\boldsymbol{\theta}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(j)}\big\|_2 + \eta\big\|\mathbf{J}^{(j)} - \mathbf{J}^{(0)}\big\|_2\big\|\mathbf{p}^{(j)} - \mathbf{y}\big\|_2 \\
&+ k_\mu\eta\big\|\mathbf{J}^{(0)}\big\|_2\big\|\mathbf{f}^{(j)} - \mathbf{J}^{(0)\top}(\boldsymbol{\theta}^{(j)} - \boldsymbol{\theta}^{(0)})\big\|_2
\end{aligned}
$$

where the inequality holds due to triangle inequality, matrix spectral norm inequality, and the Lipschitz continuity of the logistic function. We now bound the three terms in the RHS separately.

$$\big\|\mathbf{I} - \eta(m\lambda\mathbf{I} + k_\mu\mathbf{H}^{(0)})\big\|_2\big\|\boldsymbol{\theta}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(j)}\big\|_2 \leq (1-\eta m\lambda)\big\|\boldsymbol{\theta}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(j)}\big\|_2,$$

where the inequality holds since, $\eta(m\lambda\mathbf{I} - k_\mu[\mathbf{J}^{(0)\top}\mathbf{J}^{(0)}]) \preceq \eta(m\lambda\mathbf{I} + C_1 n_t m L\mathbf{I}) \preceq \mathbf{I}$, for some $C_1 > 0$, where the inequality holds due to the choice of η. For the second term, we have,

$$\eta\big\|\mathbf{J}^{(j)} - \mathbf{J}^{(0)}\big\|_2\big\|\mathbf{p}^{(j)} - \mathbf{y}\big\|_2 \leq C_2\eta n_t^P\tau^{1/3}L^{7/2}\sqrt{m\log m},$$

for some $C_2 > 0$, where the inequality holds due to Eq (D.8) and Lemma D.3. For the third term,

$$k_\mu\eta\big\|\mathbf{J}^{(0)}\big\|_2\big\|\mathbf{f}^{(j)} - \mathbf{J}^{(0)\top}(\boldsymbol{\theta}^{(j)} - \boldsymbol{\theta}^{(0)})\big\|_2 \leq C_3\eta n_t^P m\tau^{4/3}L^{7/2}\sqrt{\log m}$$

for some $C_3 > 0$, where the inequality holds due to Eq (D.7) and Eq (D.9). By chaining all the inequalities, we have,

$$\big\|\boldsymbol{\theta}^{j+1} - \widetilde{\boldsymbol{\theta}}^{j+1}\big\|_2 \leq (1-\eta m\lambda)\big\|\boldsymbol{\theta}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(j)}\big\|_2 + C_2\eta n_t^P\tau^{1/3}L^{7/2}\sqrt{m\log m} + C_3\eta n_t^P m\tau^{4/3}L^{7/2}\sqrt{\log m},$$

where $C_4 > 0$ is a constant. By recursively applying the above inequality from 0 to j, we have,

$$\big\|\boldsymbol{\theta}^{j+1} - \widetilde{\boldsymbol{\theta}}^{j+1}\big\|_2 \leq (C_2\eta n_t^P\tau^{1/3}L^{7/2}\sqrt{m\log m} + C_3\eta n_t^P m\tau^{4/3}L^{7/2}\sqrt{\log m})/(\eta m\lambda) \leq \tau/2,$$

where $C_5 > 0$ is a constant, the equality holds by the definition of τ. The last inequality holds due to the choice of m, where $m^{1/6} \geq C_4 L^{7/2}(n_t^P)^{2/3}\lambda^{-2/3}\sqrt{\log m}(C_2 + C_3\sqrt{n_t^P/\lambda})$. Therefore, for any $j \in [J]$, we have

$$\big\|\boldsymbol{\theta}^{(j)} - \boldsymbol{\theta}^{(0)}\big\|_2 \leq \big\|\widetilde{\boldsymbol{\theta}}^{(j)} - \boldsymbol{\theta}^{(0)}\big\|_2 + \big\|\boldsymbol{\theta}^{(j)} - \widetilde{\boldsymbol{\theta}}^{(j)}\big\|_2 \leq \sqrt{n_t^P/(2m\lambda)} + \tau/2 = \tau,$$

where the first inequality holds due to triangle inequality, the second inequality holds due to Lemma D.4. This inequality also shows the assumption $\big\|\boldsymbol{\theta}^{(j)} - \boldsymbol{\theta}^{(0)}\big\|_2 \leq \tau$ holds for any j. Hence, according to Lemma D.4, we have

$$\big\|\boldsymbol{\theta}^{(j)} - \boldsymbol{\theta}^{(0)} - \widehat{\boldsymbol{\gamma}}_t\big\|_2 \leq (1-\eta m\lambda)^{j/2}\sqrt{t/(m\lambda)} + C_5 m^{-2/3}L^{7/2}(n_t^P)^{7/6}\lambda^{-7/6}\sqrt{\log m}(1 + \sqrt{n_t^P/\lambda}).$$

This completes the proof. □

D.2 Proof of Lemma B.8

We first define $\mathbf{A}_t = \lambda\mathbf{I} + \sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}} \mathbf{g}_{i'j'}^{s,0}\mathbf{g}_{i'j'}^{s,0\top}/m$. By taking the gradient of Eq (B.7), we have $\widehat{\boldsymbol{\gamma}}_t$ as the solution of,

$$\sum_{s=1}^{t-1}\sum_{(i'j')\in\Omega_s^{ind}} \Big(\sigma(\langle \mathbf{g}_{i'j'}^{s,0}, \boldsymbol{\gamma}\rangle) - y_{i'j'}^s\Big)\mathbf{g}_{i'j'}^{s,0} + m\lambda\boldsymbol{\gamma} = 0$$

Define $q_t(\boldsymbol{\gamma}) = \sum_{s=1}^{t-1}\sum_{(i\prime,j\prime)\in\Omega_s^{ind}} \sigma(\langle \mathbf{g}_{i'j'}^{s,0}, \boldsymbol{\gamma}\rangle)\mathbf{g}_{i'j'}^{s,0}/m + \lambda\boldsymbol{\gamma}$ be the invertible function such that the estimated parameter $\widehat{\boldsymbol{\gamma}}_t$ satisfies $q(\widehat{\boldsymbol{\gamma}}_t) = y_{i'j'}^s\mathbf{g}_{i'j'}^{s,0}/m$.

As logistic function $\sigma(\cdot)$ is continuously differentiable, ∇q_t is continuous. Hence, according to the Fundamental Theorem of Calculus, we have $q_t(\boldsymbol{\gamma}^*) - q_t(\widehat{\boldsymbol{\gamma}}_t) = \mathbf{Q}_t(\boldsymbol{\gamma}^* - \widehat{\boldsymbol{\gamma}}_t)$, where $\mathbf{Q}_t = \int_0^1 \nabla q_t\left(l\boldsymbol{\gamma}^* + (1-l)\widehat{\boldsymbol{\gamma}}_t\right) dl$, and $\boldsymbol{\gamma}^*$ is the optimal solution of Eq (B.7), and according to Lemma B.3, $\boldsymbol{\gamma}^* = \boldsymbol{\theta}^* - \boldsymbol{\theta}_0$.

Therefore, $\nabla q_t(\boldsymbol{\gamma}) = \sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}} \dot{\sigma}(\langle \mathbf{g}_{i'j'}^{s,0}, \boldsymbol{\gamma}\rangle)\mathbf{g}_{i'j'}^{s,0}\mathbf{g}_{i'j'}^{s,0\ \top}/m + \lambda\mathbf{I}$, where $\dot{\sigma}$ is the first order derivative of $\sigma(\cdot)$. Accordingly, we have the following inequality,

$$\begin{aligned}
\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\gamma}}_t\|_{\bar{\mathbf{A}}_t} =& \|\mathbf{Q}_t^{-1}(q_t(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0) - q_t(\widehat{\boldsymbol{\gamma}}_t))\|_{\bar{\mathbf{A}}_t} \\
=& \sqrt{(q_t(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0) - q_t(\widehat{\boldsymbol{\gamma}}_t))^\top \mathbf{Q}_t^{-1}\bar{\mathbf{A}}_t\mathbf{Q}_t^{-1}(q_t(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0) - q_t(\widehat{\boldsymbol{\gamma}}_t))} \\
\leq& c_\mu^{-1}\|q_t(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0) - q_t(\widehat{\boldsymbol{\gamma}}_t)\|_{\bar{\mathbf{A}}_t^{-1}}
\end{aligned}$$

where the first equality is due to the definition of q_t and $\mathbf{Q}_t$, and the inequality is based on the definition of c_μ, which is defined as $c_\mu = \inf_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \dot{\sigma}(\boldsymbol{x}^\top\boldsymbol{\theta})$. It is easy to verify that $c_\mu \leq \frac{1}{4}$. Thus, we can conclude that $\mathbf{Q}_t \succeq c_\mu\bar{\mathbf{A}}_t$, which implies that $\mathbf{Q}_t^{-1} \preceq c_\mu^{-1}\bar{\mathbf{A}}_t^{-1}$.

Based on the definition of $\widehat{\boldsymbol{\gamma}}_t$ and the assumption on the noisy feedback that $y_{ij}^t = \sigma(h(\mathbf{x}_i) - h(\mathbf{x}_j)) + \xi_{ij}^t$, where ξ_{ij}^t is the noise in user feedback, we have

$$\begin{aligned}
&\sqrt{m}q_t(\widehat{\boldsymbol{\gamma}}_t) - \sqrt{m}q_t(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0) \\
=& \sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}} (y_{i'j'} - \langle \mathbf{g}_{i'j'}^{s,0},\ \boldsymbol{\theta}^* - \boldsymbol{\theta}_0\rangle)\mathbf{g}_{i'j'}^{s,0}/\sqrt{m} - \lambda\sqrt{m}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0) \\
=& \sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}} \xi_{i'j'}^s\mathbf{g}_{i'j'}^{s,0}/\sqrt{m} - \lambda\sqrt{m}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0).
\end{aligned}$$

As $\xi_{i'j'}^s \sim \nu$-sub-Gaussian, according to Theorem 1 in [14], with probability at least $1-\delta_1$

$$\begin{aligned}
\|\sqrt{m}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\gamma}}_t)\|_{\bar{\mathbf{A}}_t} \leq& c_\mu^{-1}\|\sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}} \xi_{i'j'}^s\mathbf{g}_{i'j'}^{s,0}/\sqrt{m} - \lambda\sqrt{m}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0)\|_{\bar{\mathbf{A}}_t^{-1}} \\
\leq& c_\mu^{-1}\Big(\|\sum_{s=1}^{t-1}\sum_{(i',j')\in\Omega_s^{ind}} \xi_{i'j'}^s\mathbf{g}_{i'j'}^{s,0}/\sqrt{m}\|_{\bar{\mathbf{A}}_t^{-1}} + \sqrt{\lambda m}\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\|\Big) \\
\leq& c_\mu^{-1}(\sqrt{\nu^2\log\det(\bar{\mathbf{A}}_t)/(\delta_1^2\det(\lambda\mathbf{I}))} + \sqrt{\lambda}S).
\end{aligned}$$

This completes the proof.

D.3 Proof of Lemma B.10

As defined before, we assume that there are in n_T possible document candidate to be evaluated during the model learning, and there are N_T^P possible document pairs to be evaluated, and $N_T^P = \sum_{s=1}^{T} V_s^2 \leq T V_{\max}^2$. Then, we define the following quantities.

$$\mathbf{G} = [\mathbf{g}(\mathbf{x}_1; \boldsymbol{\theta}_0)/\sqrt{m}, \ldots, \mathbf{g}(\mathbf{x}_{n_T}; \boldsymbol{\theta}_0)/\sqrt{m}] \in \mathbb{R}^{p \times n_T}$$
$$\widehat{\mathbf{G}} = [\mathbf{G}, (\mathbf{g}(\mathbf{x}_{1,1}; \boldsymbol{\theta}_0) - \mathbf{g}(\mathbf{x}_{1,2}; \boldsymbol{\theta}_0))/\sqrt{m}, \ldots, (\mathbf{g}(\mathbf{x}_{N,1}; \boldsymbol{\theta}_0) - \mathbf{g}(\mathbf{x}_{N_T^P,2}; \boldsymbol{\theta}_0))/\sqrt{m}] \in \mathbb{R}^{p \times (N_T^P + n_T)}$$

where $\widehat{\mathbf{G}}$ contains all the possible document features, and feature vectors of all the possible document pairs. Similarly, we can have the corresponding $\widehat{\mathbf{H}}$ for $\widehat{\mathbf{G}}$ based on the values of $\mathbf{H}$. We define the effective dimension of $\widehat{\mathbf{H}}$ as,

$$\widehat{d} = (\log\det(\mathbf{I} + \widehat{\mathbf{H}}/\lambda))/(\log(1 + 4(N_T^P + n_T)/\lambda)). \tag{D.11}$$

First, we will verify the definition of the effective dimension. Suppose there exists a mapping $\psi : \mathbb{R}^d \to \mathbb{R}^{\widetilde{d}}$ satisfying $\|\psi(x)\|_2 \leq 1$ which maps any document feature vector $\mathbf{x} \in \mathbb{R}^d$ to the Hilbert space $\mathcal{H}$ associated with the Gram matrix $\mathbf{H} \in \mathbb{R}^{n_T \times n_T}$ over the set of all the possible document features. Then $\mathbf{H} = \boldsymbol{\Psi}^\top \boldsymbol{\Psi}$ with $\boldsymbol{\Psi} = [\psi(\mathbf{x}_1), ...\psi(\mathbf{x}_{n_T})]$. And for the pairwise matrix $\widehat{\mathbf{H}}$, we have $\widehat{\mathbf{H}} = \widehat{\boldsymbol{\Psi}}^\top \widehat{\boldsymbol{\Psi}}$ with $\widehat{\boldsymbol{\Psi}} = [\psi(\mathbf{x}_1), \psi(\mathbf{x}_1) - \psi(\mathbf{x}_2), ..., \psi(\mathbf{x}_{n_T}) - \psi(\mathbf{x}_{n_T-1}), 0]$. Thus, we can bound the effective dimension $\widehat{d}$ as,

$$\widehat{d}_N = \frac{\log\det(\mathbf{I} + \widehat{\mathbf{H}}/\lambda)}{\log(1 + 4(N_T^P + n_T)/\lambda)} = \frac{\log\det(\mathbf{I} + \widehat{\boldsymbol{\Psi}}\widehat{\boldsymbol{\Psi}}^\top/\lambda)}{\log(1 + 4(N_T^P + n_T)/\lambda)} \leq \widetilde{d} \cdot \frac{\log\|\mathbf{I} + \widehat{\boldsymbol{\Psi}}\widehat{\boldsymbol{\Psi}}^\top/\lambda\|_2}{\log(1 + 4(N_T^P + n_T))}$$

which is based on the fact that $\det(\mathbf{I} + \mathbf{M}^\top\mathbf{M}/\lambda) = \det(\mathbf{I} + \mathbf{M}\mathbf{M}^\top/\lambda)$ holds for any $\mathbf{M}$, and we have $\det(\mathbf{M}) \leq \|\mathbf{M}\|_2^{\widetilde{d}}$ for any $\mathbf{M} \in \mathbb{R}^{\widetilde{d} \times \widetilde{d}}$. As we know that,

$$\|\mathbf{I} + \widehat{\boldsymbol{\Psi}}\widehat{\boldsymbol{\Psi}}^\top/\lambda\|_2 \leq 1 + \|\widehat{\boldsymbol{\Psi}}\widehat{\boldsymbol{\Psi}}^\top\|_2/\lambda \leq 1 + \sum_{s=1}^{n_T^P} \|(\psi(\mathbf{x}_{i,1}) - \psi(\mathbf{x}_{i,2}))(\psi(\mathbf{x}_{i,1}) - \psi(\mathbf{x}_{i,2}))^\top\|_2/\lambda \leq 1 + 4(n_T^P + n_T)$$

we have $\widehat{d} \leq \widetilde{d}$.

Proof of Lemma B.10. According to Lemma 11 in [14]], we have the following inequality:

$$\sum_{t=1}^{T} \sum_{(i',j') \in \Omega_t} \min\left\{\|\mathbf{g}_{i'j'}^{t,0}/\sqrt{m}\|_{\mathbf{A}_t^{-1}}^2, 1\right\} \leq 2\log\frac{\det\mathbf{A}_T}{\det\lambda\mathbf{I}}.$$

Based on the definition of $\mathbf{G}$ and $\widehat{\mathbf{G}}$, we have,

$$\begin{aligned}\log\frac{\det\mathbf{A}_T}{\det\lambda\mathbf{I}} &= \log\det\left(\mathbf{I} + \sum_{t=1}^{T} \sum_{(i',j') \in \Omega_t^{ind}} \mathbf{g}_{i'j'}^{t,0}\mathbf{g}_{i'j'}^{t,0\top}/(m\lambda)\right) \\ &\leq \log\det\left(\mathbf{I} + \sum_{i=1}^{N} \mathbf{g}_i\mathbf{g}_i^\top/(m\lambda)\right) = \log\det\left(\mathbf{I} + \widehat{\mathbf{G}}\widehat{\mathbf{G}}^\top/\lambda\right) = \log\det\left(\mathbf{I} + \widehat{\mathbf{G}}^\top\widehat{\mathbf{G}}/\lambda\right),\end{aligned}$$

where the inequality holds naively, the third equality holds since for any matrix $\mathbf{M} \in \mathbb{R}^{p\times N}$, we have $\det(\mathbf{I}+\mathbf{M}\mathbf{M}^\top) = \det(\mathbf{I}+\mathbf{M}^\top\mathbf{M})$. Therefore, we have

$$\begin{aligned}
\log\det\left(\mathbf{I}+\widehat{\mathbf{G}}^\top\widehat{\mathbf{G}}/\lambda\right) &= \log\det\left(\mathbf{I}+\widehat{\mathbf{H}}/\lambda+(\widehat{\mathbf{G}}^\top\widehat{\mathbf{G}}-\mathbf{H})/\lambda\right)\\
&\le \log\det\left(\mathbf{I}+\widehat{\mathbf{H}}/\lambda\right)+\langle(\mathbf{I}+\widehat{\mathbf{H}}/\lambda)^{-1},(\widehat{\mathbf{G}}^\top\widehat{\mathbf{G}}-\widehat{\mathbf{H}})/\lambda\rangle\\
&\le \log\det\left(\mathbf{I}+\widehat{\mathbf{H}}/\lambda\right)+\|(\mathbf{I}+\widehat{\mathbf{H}}/\lambda)^{-1}\|_F\|(\widehat{\mathbf{G}}^\top\widehat{\mathbf{G}}-\widehat{\mathbf{H}})/\lambda\|_F\\
&\le \log\det\left(\mathbf{I}+\widehat{\mathbf{H}}/\lambda\right)+\sqrt{N_T^P+n_T}\|\widehat{\mathbf{G}}^\top\widehat{\mathbf{G}}-\widehat{\mathbf{H}}\|_2
\end{aligned}$$

where the first equality stands trivially, the second inequality is due to the convexity of $\log\det(\cdot)$, the third inequality holds due to the fact that $\langle\mathbf{M},\mathbf{B}\rangle \le \|\mathbf{M}\|_F\|\mathbf{B}\|_F$, the third inequality holds due to the facts that $\mathbf{I}+\widehat{\mathbf{H}}/\lambda \succeq \mathbf{I}$, $\lambda \ge 1$ and $\|\mathbf{M}\|_F \le \sqrt{N}\|\mathbf{M}\|_2$ for any $\mathbf{M}\in\mathbb{R}^{N\times N}$. According to Lemma D.1, we know that with properly chosen m, $\|\mathbf{G}^\top\mathbf{G}-\mathbf{H}\|_F \le n\epsilon$. For any $(i,j)\in[N]^2$ in $\widehat{\mathbf{G}}^\top\widehat{\mathbf{G}}-\widehat{\mathbf{H}}$, we have

$$\begin{aligned}
&|\langle\mathbf{g}(\mathbf{x}_{i,1};\boldsymbol{\theta}_0)-\mathbf{g}(\mathbf{x}_{i,2};\boldsymbol{\theta}_0),\mathbf{g}(\mathbf{x}_{j,1};\boldsymbol{\theta}_0)-\mathbf{g}(\mathbf{x}_{j,2};\boldsymbol{\theta}_0)\rangle/m-\widehat{\mathbf{H}}_{i,j}|\\
\le&|\langle\mathbf{g}(\mathbf{x}_{i,1};\boldsymbol{\theta}_0),\mathbf{g}(\mathbf{x}_{j,1};\boldsymbol{\theta}_0)\rangle-\mathbf{H}_{(i,1),(j,1)}|+|\langle\mathbf{g}(\mathbf{x}_{i,1};\boldsymbol{\theta}_0),\mathbf{g}(\mathbf{x}_{j,2};\boldsymbol{\theta}_0)\rangle-\mathbf{H}_{(i,1),(j,2)}|\\
&+|\langle\mathbf{g}(\mathbf{x}_{i,2};\boldsymbol{\theta}_0),\mathbf{g}(\mathbf{x}_{j,1};\boldsymbol{\theta}_0)\rangle-\mathbf{H}_{(i,2),(j,1)}|+|\langle\mathbf{g}(\mathbf{x}_{i,2};\boldsymbol{\theta}_0),\mathbf{g}(\mathbf{x}_{j,2};\boldsymbol{\theta}_0)\rangle-\mathbf{H}_{(i,2),(j,2)}|.
\end{aligned}$$

Therefore, we have $\|\widehat{\mathbf{G}}^\top\widehat{\mathbf{G}}-\widehat{\mathbf{H}}\|_F \le 4N_T^P/n_T\|\mathbf{G}^\top\mathbf{G}-\mathbf{H}\|_F$, and by choosing m, we have

$$\log\det\left(\mathbf{I}+\widehat{\mathbf{H}}/\lambda\right)+\sqrt{N}\|\widehat{\mathbf{G}}^\top\widehat{\mathbf{G}}-\widehat{\mathbf{H}}\|_2 \le \widehat{d}\log(1+4(n_T+N_T^P)/\lambda)+1$$

This completes the proof. □

E Proofs of lemmas in Appendix D

E.1 Proof of Lemma D.2

Proof of Lemma D.2. With τ satisfying the condition of Lemmas B.5,B.6, and B.7, for any $j\in[J]$ at round t, we have,

$$\|\mathbf{J}^{(j)}\|_F \le \sqrt{n_t^P}\max_{i\in[n_t^P]}\|\mathbf{g}(\mathbf{x}_{i,1};\boldsymbol{\theta}^{(j)})-\mathbf{g}(\mathbf{x}_{i,2};\boldsymbol{\theta}^{(j)})\|_2 \le \widehat{2}C_3^z\sqrt{n_t^P mL} \tag{E.1}$$

where the first inequality holds due to $\|\mathbf{J}^{(j)}\|_F \le \sqrt{n_t^P}\|\mathbf{J}^{(j)}\|_{2,\infty}$, the second inequality holds due to the triangle inequality and Lemma B.7. Accordingly, we have,

$$\begin{aligned}
\|\mathbf{J}^{(j)}-\mathbf{J}^{(0)}\|_F \le&\sqrt{n_t^P}\max_{i\in[n_t^P]}\|\mathbf{g}(\mathbf{x}_{i,1};\boldsymbol{\theta}^{(j)})-\mathbf{g}(\mathbf{x}_{i,2};\boldsymbol{\theta}^{(j)})-(\mathbf{g}(\mathbf{x}_{i,1};\boldsymbol{\theta}^{(0)})-\mathbf{g}(\mathbf{x}_{i,2};\boldsymbol{\theta}^{(0)}))\|_2\\
\le&\widehat{2}C_3^wC_3^z\sqrt{n_t^P m\log m}\tau^{1/3}L^{7/2},
\end{aligned} \tag{E.2}$$

where the second inequality holds due to triangle inequality and lemma B.7. Similarly, we have,

$$\|\mathbf{f}^{(s)}-\mathbf{f}^{(j)}-[\mathbf{J}^{(j)}]^\top(\boldsymbol{\theta}^{(s)}-\boldsymbol{\theta}^{(j)})\|_F \le \widehat{C}_6\tau^{4/3}L^3\sqrt{n_t m\log m},$$

where $C_4>0$ and $C_5>0$ are constants, the second inequality is based on Lemma B.5 with the assumption that $\|\boldsymbol{\theta}^{(j)}-\boldsymbol{\theta}^{(0)}\|_2\le\tau$ and $\|\boldsymbol{\theta}^{(s)}-\boldsymbol{\theta}^{(0)}\|_2\le\tau$.

Last, it is easy to have that $\|\mathbf{y}\|_2 \le \sqrt{n_t}\max_{i\in[n_t]}|y_i| \le \sqrt{n_t}$. This completes the proof. □

E.2 Proof of Lemma D.3

Proof of Lemma D.3. The proof is based on Lemma C.3 in [251], where the convergence of squared loss is analyzed. In our case, we adopt the cross-entropy loss. In the following, we provide the key difference between our analysis concerning the cross-entropy function.

With τ satisfying the conditions in Lemmas B.7, B.5, B.6, and the loss function we have for the neural network as,

$$\begin{aligned}
&\mathcal{L}_t(\boldsymbol{\theta}) \\
=&\sum_{s=1}^{t}\sum_{(i,j)\in\Omega_s^{ind}} -(1-y_{ij}^s)\log(1-\sigma(f_{ij})) - y_{ij}^s\log(\sigma(f_{ij})) + \frac{m\lambda}{2}\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|^2 \\
=&\sum_{i=1}^{n_t} -y_i\log(\sigma(f(\mathbf{x}_{i,1};\boldsymbol{\theta})-f(\mathbf{x}_{i,2};\boldsymbol{\theta})) - (1-y_i)\log(1-\sigma(f(\mathbf{x}_{i,1};\boldsymbol{\theta})-f(\mathbf{x}_{i,2};\boldsymbol{\theta})) + \frac{m\lambda}{2}\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|^2
\end{aligned}$$

with the first equation the same as Eq (2.1.2), and we re-write the loss with n_t, which is defined in the proof lemma B.9.

We need the following quantities,

$$\begin{aligned}
\mathbf{J}(\boldsymbol{\theta}) &= \Big(\mathbf{g}(\mathbf{x}_{1,1};\boldsymbol{\theta})-\mathbf{g}(\mathbf{x}_{1,2};\boldsymbol{\theta}),\ldots,\mathbf{g}(\mathbf{x}_{n_t,1};\boldsymbol{\theta})-\mathbf{g}(\mathbf{x}_{n_t,2};\boldsymbol{\theta})\Big) \in \mathbb{R}^{p\times n_t} \\
\mathbf{f}(\boldsymbol{\theta}) &= \Big(f(\mathbf{x}_{1,1};\boldsymbol{\theta})-f(\mathbf{x}_{1,2};\boldsymbol{\theta}),\ldots,f(\mathbf{x}_{n_t,1};\boldsymbol{\theta})-f(\mathbf{x}_{n_t,2};\boldsymbol{\theta})\Big)^\top \in \mathbb{R}^{n_t\times 1} \\
\mathbf{p}(\boldsymbol{\theta}) &= \Big(\sigma(f(\mathbf{x}_{1,1};\boldsymbol{\theta})-f(\mathbf{x}_{1,2};\boldsymbol{\theta})),\ldots,\sigma(f(\mathbf{x}_{n_t,1};\boldsymbol{\theta})-f(\mathbf{x}_{n_t,2};\boldsymbol{\theta}))\Big)^\top \in \mathbb{R}^{n_t\times 1} \\
\mathbf{y} &= \big(y_1,\ldots y_{n_t}\big)^\top \in \mathbb{R}^{n_t\times 1}.
\end{aligned}$$

First, the cross entropy loss, $l=\sum_{k=1}^{K} -y_k\log(\sigma(s_k)) - (1-y_k)\log(1-\sigma(s_k))$ is convex and $\frac{1}{4}$-smooth. The convexity is trivial to prove. For the smoothness, we have $\frac{\partial l}{\partial s_k} = \sigma(s_k)-y_k$, and $\frac{\partial^2 l}{\partial s_k^2} = \sigma(s_k)(1-\sigma(s_k))$. As $\sigma(s_k)\in[0,1]$, $\frac{\partial^2 l}{\partial \boldsymbol{s}^2} \preceq \frac{1}{4}\mathbf{I}$.

Based on the smoothness of cross entropy loss function, we have for arbitrary $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$

$$\begin{aligned}
&\mathcal{L}_t(\boldsymbol{\theta}') - \mathcal{L}_t(\boldsymbol{\theta}) \qquad\qquad \text{(E.3)}\\
\leq&\langle \mathbf{p}(\boldsymbol{\theta})-\mathbf{y}, \mathbf{f}(\boldsymbol{\theta}'-\mathbf{f}(\boldsymbol{\theta}))\rangle + \frac{1}{8}\|\mathbf{f}(\boldsymbol{\theta}')-\mathbf{f}(\boldsymbol{\theta})\|^2 + m\lambda\langle\boldsymbol{\theta}-\boldsymbol{\theta}^0,\boldsymbol{\theta}'-\boldsymbol{\theta}\rangle + \frac{m\lambda}{2}\|\boldsymbol{\theta}'-\boldsymbol{\theta}\|^2 \\
\leq&\langle\nabla\mathcal{L}(\boldsymbol{\theta}),\boldsymbol{\theta}'-\boldsymbol{\theta}\rangle + \|\mathbf{p}(\boldsymbol{\theta})-\mathbf{y}\|_2\|\mathbf{e}\|_2 + \frac{1}{8}\|\mathbf{e}\|^2 + \frac{C_e}{8}(m\lambda+n_t mL)\|\boldsymbol{\theta}'-\boldsymbol{\theta}\|^2 \qquad \text{(E.4)}
\end{aligned}$$

where $\mathbf{e}=\mathbf{f}(\boldsymbol{\theta}')-\mathbf{f}(\boldsymbol{\theta})-\mathbf{J}(\boldsymbol{\theta})^\top(\boldsymbol{\theta}'-\boldsymbol{\theta})$, the last inequality is based on Lemma D.2. By the convexity of cross entropy loss, we have,

$$\begin{aligned}
\mathcal{L}_t(\boldsymbol{\theta}') - \mathcal{L}_t(\boldsymbol{\theta}) \geq&\langle\mathbf{p}(\boldsymbol{\theta})-\mathbf{y},\mathbf{f}(\boldsymbol{\theta}')-\mathbf{f}(\boldsymbol{\theta})\rangle + m\lambda\langle\boldsymbol{\theta}-\boldsymbol{\theta}^0,\boldsymbol{\theta}'-\boldsymbol{\theta}\rangle + \frac{m\lambda}{2}\|\boldsymbol{\theta}'-\boldsymbol{\theta}\|^2 \\
\geq&-\frac{\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2}{2m\lambda} - \|\mathbf{p}(\boldsymbol{\theta})-\mathbf{y}\|_2\|\mathbf{e}\|_2, \qquad \text{(E.5)}
\end{aligned}$$

where the third inequality is based on Cauchy-Schwarz inequality, the fourth inequality is based on the fact that $\langle\mathbf{a},\mathbf{x}+c\|\mathbf{x}\|_2^2\rangle \geq -\|\mathbf{a}\|_2^2/(4c)$ for any vectors $\mathbf{a},\mathbf{x}$ and. $c>0$.

Taking $\boldsymbol{\theta}' = \boldsymbol{\theta} - \eta\nabla\mathcal{L}(\boldsymbol{\theta})$ for Eq (E.4) and substituting Eq (E.5) into Eq (E.4), we have

$$\begin{aligned}\mathcal{L}_t(\boldsymbol{\theta} - \eta\nabla\mathcal{L}(\boldsymbol{\theta})) - \mathcal{L}_t(\boldsymbol{\theta}) \leq & -\eta(1 - \frac{C_e}{8}(m\lambda + n_t mL)\eta)\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 + \|\mathbf{p}(\boldsymbol{\theta}) - \mathbf{y}\|_2\|\mathbf{e}\|_2 + \frac{1}{8}\|\mathbf{e}\|^2 \\ \leq & m\lambda\eta(\mathcal{L}(\boldsymbol{\theta}') - \mathcal{L}(\boldsymbol{\theta})/2) + \|\mathbf{e}\|_2^2(1 + 2m\lambda\eta + 2/(m\lambda\eta)).\end{aligned}$$

Interested readers can refer to [251] for the details of the derivations. It is easy to verify that $\|\mathbf{p}(\boldsymbol{\theta}) - \mathbf{y}\|^2 \leq 2\mathcal{L}(\boldsymbol{\theta})$. Therefore, by taking $\boldsymbol{\theta} = \boldsymbol{\theta}^{(j)}$ and $\boldsymbol{\theta}' = \boldsymbol{\theta}^{(0)}$, we have

$$\mathcal{L}(\boldsymbol{\theta}^{(j+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(0)}) \leq (1 - m\lambda\eta/2)\big[\mathcal{L}(\boldsymbol{\theta}^{(j)}) - \mathcal{L}(\boldsymbol{\theta}^{(0)})\big] + m\lambda\eta/2\mathcal{L}(\boldsymbol{\theta}^{(0)}) + \|\mathbf{e}\|_2^2\big(1 + 2m\lambda\eta + 2/(m\lambda\eta)\big)$$

We have $\mathcal{L}(\boldsymbol{\theta}^{(0)}) = n_t \log 2 \leq n_t$, and $\|\mathbf{e}\|_2^2\big(1 + 2m\lambda\eta + 2/(m/\lambda\eta)\big) \leq m\lambda\eta n_t/2$ from [251], we have $\mathcal{L}(\boldsymbol{\theta}^{(j+1)}) - \mathcal{L}(\boldsymbol{\theta}^0) \leq 2n_t$, $\|\mathbf{p}^{(j+1)} - \mathbf{y}\|_2 \leq 2\sqrt{n_t}$

This completes the proof. □

E.3 Proof of Lemma D.4

Proof of Lemma D.4. It can be verified that τ satisfies the conditions of Lemma D.2, thus Lemma D.2 holds. We know that $\widetilde{\boldsymbol{\theta}}^{(j)}$ is the sequence generated by applying gradient descent on the following problem:

$$\begin{aligned}\min_{\boldsymbol{\theta}} \widetilde{\mathcal{L}}(\boldsymbol{\theta}) = & \sum_{i=1}^{n_t} -(1 - y_i)\log(1 - \sigma((\mathbf{g}(\mathbf{x}_{i,1};\boldsymbol{\theta}^{(0)}) - \mathbf{g}(\mathbf{x}_{i,2};\boldsymbol{\theta}^{(0)}))^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)}))) \\ & - y_i\log(\sigma((\mathbf{g}(\mathbf{x}_{i,1};\boldsymbol{\theta}^{(0)}) - \mathbf{g}(\mathbf{x}_{i,2};\boldsymbol{\theta}^{(0)}))^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)}))) + \frac{m\lambda}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2\end{aligned}$$

Therefore $\|\boldsymbol{\theta}^{(0)} - \widetilde{\boldsymbol{\theta}}^{(j)}\|_2$ can be bounded as

$$\begin{aligned}\frac{m\lambda}{2}\|\boldsymbol{\theta}^{(0)} - \widetilde{\boldsymbol{\theta}}^{(j)}\|_2^2 \leq & \sum_{i=1}^{n_t} -(1 - y_i)\log(1 - \sigma((\mathbf{g}(\mathbf{x}_{i,1};\boldsymbol{\theta}^{(0)}) - \mathbf{g}(\mathbf{x}_{i,2};\boldsymbol{\theta}^{(0)}))^\top(\widetilde{\boldsymbol{\theta}}^{(j)} - \boldsymbol{\theta}^{(0)}))) \\ & - y_i\log(\sigma((\mathbf{g}(\mathbf{x}_{i,1};\boldsymbol{\theta}^{(0)}) - \mathbf{g}(\mathbf{x}_{i,2};\boldsymbol{\theta}^{(0)}))^\top(\widetilde{\boldsymbol{\theta}}^{(j)} - \boldsymbol{\theta}^{(0)}))) + \frac{m\lambda}{2}\|\widetilde{\boldsymbol{\theta}}^{(j)} - \boldsymbol{\theta}_0\|^2 \\ \leq & \sum_{i=1}^{n_t} -(1 - y_i)\log(1 - \sigma((\mathbf{g}(\mathbf{x}_{i,1};\boldsymbol{\theta}^{(0)}) - \mathbf{g}(\mathbf{x}_{i,2};\boldsymbol{\theta}^{(0)}))^\top(\widetilde{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^{(0)}))) \\ & - y_i\log(\sigma((\mathbf{g}(\mathbf{x}_{i,1};\boldsymbol{\theta}^{(0)}) - \mathbf{g}(\mathbf{x}_{i,2};\boldsymbol{\theta}^{(0)}))^\top(\widetilde{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^{(0)}))) + \frac{m\lambda}{2}\|\widetilde{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}_0\|^2.\end{aligned}$$

It is easy to verify that $\widetilde{\mathcal{L}}$ is a $m\lambda$-strongly convex function and $C_1(n_t mL + m\lambda)$-smooth function for some positive constant C_1, since

$$\nabla^2\widetilde{\mathcal{L}} \preceq \big(\frac{1}{4}\big\|\mathbf{J}^{(0)}\big\|_2^2 + m\lambda\big)\mathbf{I} \preceq C_1(n_t mL + m\lambda),$$

where the first inequality holds due to the definition of $\widetilde{\mathcal{L}}$, the second inequality holds due to Lemma D.2. Since we choose $\eta \leq C_1(n_t mL + m\lambda)^{-1}$, then by standard results of gradient descent on ridge linear regression, $\widetilde{\boldsymbol{\theta}}^{(j)}$ converges to $\boldsymbol{\theta}^{(0)} + \widehat{\boldsymbol{\gamma}}_t$ with a convergence rate specified as follows,

$$\begin{aligned}\big\|\widetilde{\boldsymbol{\theta}}^{(j)} - \boldsymbol{\theta}^{(0)} - \widehat{\boldsymbol{\gamma}}_t\big\|_2^2 \leq & (1 - \eta m\lambda)^j \cdot \frac{2}{m\lambda}(\mathcal{L}(\boldsymbol{\theta}^{(0)}) - \mathcal{L}\big(\boldsymbol{\theta}^{(0)} + \widehat{\boldsymbol{\gamma}}_t\big)) \\ \leq & \frac{2(1 - \eta m\lambda)^j}{m\lambda}\mathcal{L}(\boldsymbol{\theta}^{(0)}) \leq (1 - \eta m\lambda)^j n_t,\end{aligned}$$

where the first inequality holds due to the convergence result for gradient descent and the fact that $\boldsymbol{\theta}^{(0)} + \widehat{\boldsymbol{\gamma}}_t$ is the minimal solution to $\mathcal{L}$, the second inequality holds since $\mathcal{L} \geq 0$, the last inequality holds due to Lemma D.2. □