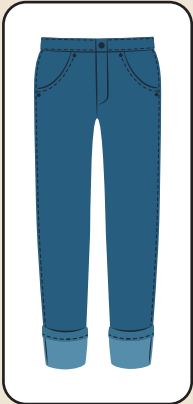




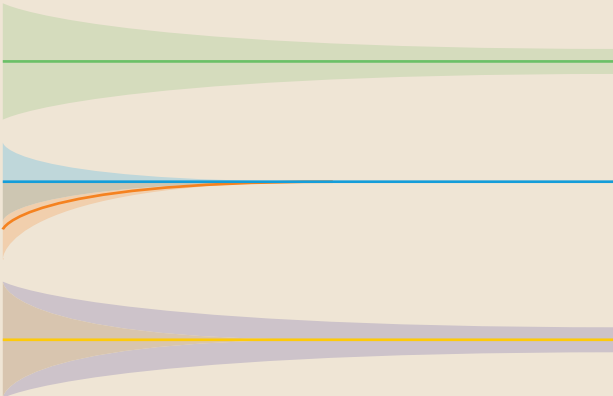
Estimation Methods for Ranking and Recommendation



Addressing Bias, Variance,
Consistency, and Efficiency



Norman Knyazev



Estimation Methods for Ranking and Recommendation: Addressing Bias, Variance, Consistency, and Efficiency

Norman Knyazev

Estimation Methods for Ranking and Recommendation. Addressing Bias, Variance, Consistency, and Efficiency

Norman Knyazev

Radboud Dissertation Series

ISSN: 2950-2772 (Online); 2950-2780 (Print)

Published by RADBOUD UNIVERSITY PRESS

Postbus 9100, 6500 HA Nijmegen, The Netherlands

www.radbouduniversitypress.nl

Layout: Crius Group, Hulshout

Cover image: Norman Knyazev

Printing: DPN Rikken/Pumbo

ISBN: 9789465152707

DOI: 10.54195/9789465152707

Free download at: <https://doi.org/10.54195/9789465152707>

© 2026 Norman Knyazev

**RADBOUD
UNIVERSITY
PRESS**

This is an Open Access book published under the terms of Creative Commons Attribution-Noncommercial-NoDerivatives International license (CC BY-NC-ND 4.0). This license allows reusers to copy and distribute the material in any medium or format in unadapted form only, for noncommercial purposes only, and only so long as attribution is given to the creator, see <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

Estimation Methods for Ranking and Recommendation: Addressing Bias, Variance, Consistency, and Efficiency

Proefschrift

ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen
op gezag van de rector magnificus prof. dr. J.M. Sanders,
volgens besluit van het college voor promoties
in het openbaar te verdedigen

op maandag 21 september 2026

om 16:30 uur precies

door

Norman Knyazev

Promotor:

Prof. dr. ir. D. Hiemstra

Copromotor:

Dr. H.R. Oosterhuis (Universiteit van Amsterdam)

Manuscriptcommissie:

Prof. dr. ir. A.P. de Vries

Prof. dr. M.A. Larson

Prof. dr. A. Hanjalic (Universiteit Utrecht)

Prof. dr. P. Castells (Amazon, Verenigde Staten)

Dr. J. Huang (University of Cambridge, Verenigd Koninkrijk)

Acknowledgements

I guess I always knew I wanted to become a doctor. I remember, when I was little, I first really wanted to become a dentist. Then, when it was time to go to the university, I almost ended up studying medicine. But for a variety of reasons, neither of these happened—it just didn't feel right. It was only when I joined Radboud University did it click with me—now this was the kind of doctor I really *did* want to become. What was different this time? In part, it was likely the choice of the subject. Yet I am convinced that—even more importantly—it was the people throughout this journey, and their support and encouragement, that really made it happen. And for this, I would like to thank the following individuals.

First and foremost, I would like to thank my supervisors Djoerd Hiemstra and Harrie Oosterhuis. For so many things. For first giving me this opportunity. For showing me what it means to be a good researcher. For pushing me forward when I needed it, and for pulling me back when I had gone too far. Harrie, I want to sincerely thank you for being my science role model. Not only did you pass me your passion for all things ranking and bias. You taught me everything I know about how to do research: how approach problems, how to look for solutions, how to clearly communicate my findings, and so much more. You encouraged my curiosity and you were always there when I needed your thoughts or advice, be it about research or anything else. I envy your students that get to continue working with you and hope our paths keep crossing in the future.

Djoerd, thank you for all your advice throughout these years, both when I was up and when I was down. Thank you for helping me understand that good meaningful research is so much more than just achieving high NDCG. And thank you for helping me navigate my journey both at the university and outside of it.

Second, I want to thank my committee: Pablo Castells, Alan Hanjalic, Jin Huang, Martha Larson and Arjen de Vries—both for your invaluable feedback on this manuscript, as well as your advice and guidance at different points throughout my research career. I am honoured to have you on my committee.

I also wish to thank everyone at Radboud Information Retrieval group, past and present—Chris, Daria, Emma, Faegheh, Gijs, Heydar, Hideaki, Koen, Mohanna, Negin, Oscar, Thorsten and Tim—for welcoming me to the group, for supporting me when I needed it, teaching me about IR, and

indulging me in my incoherent ramblings as I tried to process what I had learned about IR myself.

Furthermore, I would like to say thank you to Alex, Christos, Moabi and Zhuoran for your frequent words of advice; Bernadette, Kasper, Nicole and Tom for all the help navigating my journey at Radboud; as well as everyone else at the Data Science department for your support throughout the years.

I would also like to thank Azqa, Dmitrii, Mateo, Philipp and Shashank for welcoming me across the Netherlands, sharing great moments, along with my various passions, and generally helping me stay sane. Big thanks also to Catherine, Daan, Jaehun, Jingwei, Manel, Mariya, Olivier, Roger, Sasha for your help and company throughout these years.

I would then like to thank everyone that made my time outside of the university into such unforgettable experiences: Abdullah, Andrea, Domagoj, Eric, Esther, Hugo, Linlin, Matt, Mattia, Matteo, Omar, Natalia, Raúl, Roi, and Ronald; Alex, Alper, Bhuvesh, Clark, Dave, Dimos, Dongting, Dwip, Ivan, Lynn, Neil, Ron, Sarah, Tong, Yozen, and Leo; Adrian, Alexandre, Anand, Anastasiia, Anton, Ben, Chelsea, Daniel, Daria, Denis, Diana, Emre, Giuseppe, GJ, Huseyin, Jakob, Javier, Joos, Linglu, Luna, Matteo, Matteo, Mazen, Mihal, Mona, Moritz, Praveen, Sam, Sean, Sergei, Simon, Thomas, Thoralf, Vano, Yannik, Zsuzsanna, and Vito. I learned so much from you all!

To my family and those around us, thank you for your help, support and encouragement. Without you I wouldn't have had a chance to experience even a tenth of what I have.

Finally, the biggest thank you is to the love of my life, Francesca. Thank you for being by my side throughout this long journey, for supporting me at every moment unconditionally and making every one of my days that much brighter.

Contents

1	Introduction	1
1.1	Statistical Properties of Estimators for Learning from User Interactions	3
1.2	Statistical Properties of Estimators for Sampling	6
1.3	Research Outline and Questions	8
1.4	Main Contributions	11
1.5	Thesis Overview	13
1.6	Origins	13
2	The Bandwagon Effect: Not Just Another Bias	15
2.1	Introduction	16
2.2	Related Work	17
2.3	Background	20
2.4	Theoretical Analysis: The Bandwagon Effect	24
2.5	Bandwagon Effect Mitigation	26
2.6	Experiments and Results	28
2.7	Discussion	32
2.8	Conclusion	33
2.A	Additional Proofs	35
3	Sample-Free Almost-Exact Estimation of Plackett-Luce Propensities for Off-Policy Ranking	39
3.1	Introduction	40
3.2	Related work	41
3.3	Background: Propensity Estimation with PL Rankings . . .	42
3.4	Method: Placement Propensity Estimation	44
3.5	Experimental Setup	47
3.6	Results	49
3.7	Conclusion	54

4	Learning to Rank with Variable Result Presentation Lengths	55
4.1	Introduction	56
4.2	Related Work	58
4.3	Background: The Standard Ranking Setting	59
4.4	Variable Presentation Length Problem Setting	60
4.5	Method: Variable Length Plackett-Luce	63
4.6	Experimental Setup	70
4.7	Results	73
4.8	Conclusion and Future Work	79
5	A Lightweight Method for Modeling Confidence in Recommendations with Learned Beta Distributions	81
5.1	Introduction	82
5.2	Related Work	84
5.3	Background: The Beta Distribution	85
5.4	Method: Learned Beta Distributions for Rating Prediction .	87
5.5	Experimental Setup	95
5.6	Results and Discussion	98
5.7	Conclusion and Future Work	106
6	Conclusion	109
6.1	Main Findings	109
6.2	Summary of Main Findings	112
6.3	Future Work	114
	Bibliography	115
	Research Data Management	143
	About the Author	144
	Summary	145
	Samenvatting	146

Chapter 1

Introduction

Search and recommendation systems help users retrieve and discover relevant documents, products, content and other types of information [164, 214]. Search systems allow users to issue queries to indicate which information they would like to access and efficiently navigate to the documents containing it [31, 214]. At the same time, recommendations allow users to passively discover new content or products – collectively referred to as items – as well as creators they may not be aware of [39, 164], or alternatively re-surface those they have already engaged with [117]. Their essential role puts search and recommendation systems at the core of many online platforms, including those focused on search [15], e-commerce [95, 185], news [159], video [44, 192] and audio streaming [24, 138], recruitment [64], and many others [50, 67, 173].

Providing high-quality search and recommendation results has proven to be a difficult task. Modern item catalogues often contain millions or even billions of items, of which only some may be relevant to each individual user at any given time [20, 54]. *Recommender systems* (RecSys) algorithms aim to identify those relevant items for each user by matching them to the users' interests and needs [164]. However, as the user tastes and requirements are generally not explicitly known, various ways of modeling user preferences have been proposed [2, 93, 148, 181, 200, 228]. For example, sequential recommender systems aim to capture user preference changes over time [200], cross- and multi-domain recommendation aims to capture and interlink user interests across multiple different domains (e.g., books and movies) [93], whilst reciprocal recommendation instead aims to match pairs of users (e.g., for jobs or dating) [148].

In a related fashion, *ranking systems* – a critical component of search – are tasked with identifying the relevant documents in response to a user's

search query and presenting them in the form of a ranked list [122]. These systems similarly address a number of problems, including which of the millions of documents to rank [47, 59, 114], how many documents to show to the user [119], which order to show the results in [29, 30, 122, 143], and how the results may be perceived by the user [12, 96, 150, 151]. Of those, perhaps the most-studied task is *learning to rank* (LTR), concerned with optimizing models that decide the order of presented documents in ranking systems [122].

Historically, some of the earlier approaches for LTR and recommendation relied on expert relevance judgements, i.e., professional annotators or reviewers, who could indicate whether a document is relevant to a query or whether an item is likely to suit one or more users [66, 172, 194]. However, relying on such feedback is often not feasible. Firstly, large catalogue sizes commonly observed in industry would require a prohibitively high number of relevance judgements [37, 54, 156]. Furthermore, the subjective nature of relevance for each user or query means that an expert may be unable to provide an accurate relevance label [186]. Additionally, user preferences are also known to change over time, potentially necessitating constant relabeling [177]. For the above reasons, a more personalized and scalable approach to measuring relevance is generally required.

As a consequence, most modern search and recommendation systems instead rely on the widely available and personalized user feedback to infer item or document relevance, where the feedback most often comes in the form of user ratings, clicks and other interactions [82, 83, 209]. Optimizing models with user interactions allows the model to show the user more results similar to the clicked or highly rated ones, whilst reducing the incidence of the lower-rated or unclicked ones [112, 209]. User interactions can then also be used to evaluate the quality of learned models, where highly rated or frequently clicked items or documents should also be rated or ranked highly by these systems [122, 226].

Importantly, while learning from user behaviors has allowed for large-scale adoption of ranking and recommendation approaches, user interactions are also known to be affected by various forms of interaction biases [6, 12, 90, 145, 178, 193, 209]. These reflect various user behavior patterns that can prevent the system from learning the correct user preferences from their interactions [12, 90]. Interaction biases may thus cause the system to fail to identify relevant items or documents, or conversely to mislabel others as more relevant than they actually are [178]. Some of the numerous interaction biases include position bias [28, 89, 90, 145], selection bias [147, 178, 209], bandwagon effect [193, 218], anchoring bias [4] and many others [6, 12, 38].

Of these, perhaps the most extensively studied in LTR has been *position bias* [45], whereby users are more likely to observe and thus interact with the relevant results presented at the beginning of a ranked list, whilst being less likely to do so at lower ranks. Position bias can have a significant negative impact on our ability to accurately estimate document relevance, as fewer positive interactions with lower-ranked items may not reflect their low relevance but simply their low placement in the ranking [90]. Optimizing models in the presence of position bias may, in turn, lead to new models placing a document high in the new ranking not because it is the most relevant document but because previous models had ranked it high, and therefore it received many clicks [145]. Accounting for the effects of interaction biases is thus an important consideration within RecSys and LTR.

1.1 Statistical Properties of Estimators for Learning from User Interactions

To measure and optimize the performance of a system across multiple queries or users, ranking and recommendation approaches frequently aggregate either their individual predictions, or their prediction errors, into a single estimate based on some fixed rule, i.e., an estimator [112, 178, 195, 226]. A common error measure used for the evaluation and optimization of rating prediction models is the mean squared error (MSE), empirically estimated via the average of squared errors of the model's individual rating predictions of known ratings [178]. Similarly, ranking performance is often evaluated through the rank-weighted average relevance such as normalized discounted cumulative gain (NDCG), estimated through a weighted average of individual document relevances, where the weights reflect the document order selected by the model. Importantly, position bias and various other interaction biases can also be viewed in terms of their impact on the statistical properties of such error and relevance estimators [6, 28, 209, 218]. Some of the key properties here include: i) statistical bias, ii) variance, iii) consistency, and iv) efficiency. We will now describe how interaction biases affect statistical properties of relevance (and error) estimators in RecSys and LTR.

1.1.1 Statistical bias

The effects of position bias may be best understood in terms of the first property – *statistical bias* [90]. A common assumption about user behavior

is that users will only click on a relevant document if they first observe it, which is less likely to happen at lower positions [45, 210]. As a consequence, the number of clicks a document receives may not correctly reflect its true relevance, which is generally assumed to be the frequency with which the document would be clicked by the user given they already examined it [41]. This systematic difference between the true document relevance and the expectation of the click-based estimator is a form of statistical bias. Optimizing models without accounting for statistical bias, even with a large or an infinite number of user interactions, may lead the optimization process to converge to a suboptimal model and thus poor search and recommendation performance, along with worse user experience [145]. Unbiasedness, i.e., zero bias, is thus generally a desirable estimator property [212].

Off-policy evaluation (OPE) and optimization methods are often used to remove or offset statistical biases [28, 145, 140, 168, 178], where perhaps the best-known approach may be *inverse propensity scoring* (IPS) [212]. IPS corrects user interactions for the effects of position bias, effectively undoing its effect on each individual interaction. This is most frequently done by dividing the user clicks on each document by the probability of the user observing that document [145] or alternatively by the probability of the document being placed in each position [118], where the latter is also known as document propensity. If these probabilities are known or estimated precisely, IPS can be shown to be unbiased [212, 178]. The conceptual simplicity and effectiveness of IPS has resulted in a widespread adoption of IPS and related methods within both LTR [7, 28, 90, 145, 209] and RecSys [140, 168, 178, 212].

1.1.2 Variance

However, an important limitation of IPS is its effect on the second estimator property: estimator *variance*. In particular, whilst IPS is theoretically unbiased, some of the probabilities used for division may be very small and their inverse thus very high [10, 196]. Small differences in the observed user interactions due to randomness, e.g., which of the relevant documents did the user observe and whether they actually clicked on them, may thus get amplified by IPS and lead to very different estimated values [63, 190]. In statistical terms this means that applying IPS may lead to high estimator variance - the expectation of the estimator across all possible datasets and user behaviors is correct, but any given estimate may be far off.

Various approaches to reduce estimator variance have been proposed [53, 190, 196], including clipping [10, 125, 190], where the very low probabilities are set to some fixed higher value, thus reducing the impact of noise in

the data. However, as mentioned earlier, IPS is only unbiased when the probabilities are correct and unaltered – clipping is thus able to reduce variance but at the cost of introducing bias. Alternatively, if a high-variance estimator is unbiased, another way to obtain accurate predictions is to simply use more interaction data. However, due to the limitations in the amount of available interactions even at industrial scale, high-variance estimators in practice can still lead to significant misestimation of relevance [28]. Low estimator variance, in addition to unbiasedness, is thus another desirable estimator property.

1.1.3 Consistency

Bias and variance are also closely linked to estimator *consistency*, the third property. Consistent estimators become progressively more accurate as the amount of available data is increased, and effectively converge to the true value as the data size grows to infinity [215]. This may initially suggest that consistency is simply another word for unbiasedness. After all, unclipped IPS is unbiased and its error disappears as the amount of data is increased – it is thus indeed consistent. Nevertheless, not all unbiased estimators are consistent, as with IPS i) a single click on a randomly chosen document, or ii) all observed clicks on all documents, are both unbiased estimators of average document relevance. Yet adding more clicks to the former, an inconsistent estimator, would not improve its prediction accuracy, as it would still be based on a single click and thus would not make efficient use of additional interactions. Similarly, consistency does not require unbiasedness, as *self-normalized IPS* (SNIPS) [196], another common OPE approach, is biased, yet its error still disappears as the size of the data is increased towards infinity. Finally, variance is also not simply the finite-sample version of consistency, as deterministic biased estimators with zero variance clearly do not satisfy the requirement of converging to the true value. Altogether, the above examples underscore that consistency is its own separate property. And as we want estimators to become progressively more accurate and eventually correct as the size of the data is increased highlights the importance of consistency alongside other estimator properties.

1.1.4 Efficiency

An important limitation of consistency is that it cannot be used to compare estimators on any finite amount of data. Instead, both consistency and variance are closely related to the final property: *efficiency*. In the context of interaction biases, the most frequently encountered form of efficiency may

be *data-efficiency*, reflecting how accurate the predictions of an estimator are in relation to the amount of data available. More data-efficient estimators require fewer interactions to make equally accurate predictions as those made by the less data-efficient estimators [53, 146, 208]. And relatedly, for the same amount of data, data-efficient estimators may be able to make more accurate predictions than the less data-efficient ones. For instance, the doubly robust estimator [53, 167, 208], under some mild assumptions, generally provides lower-variance predictions compared to IPS - it requires fewer datapoints to achieve the same accuracy as IPS and is thus more data-efficient.

Previous works on interaction biases have suggested that data-efficiency may also be a particularly significant problem under the *bandwagon effect* [171, 193, 206, 218, 219, 234]. Under this type of interaction bias, user ratings and other feedback may also be influenced by that of the other users. For instance, users who believe that an item is liked by others may also become more likely to leave a higher rating themselves [43]. And conversely, a user's rating may become lower than it would have otherwise been if they are told that an item is disliked by others. This may occur, for instance, if the user is exposed to the average item rating, nowadays frequently shown alongside each item across various platforms [94, 58, 71]. In the above cases, individual ratings may then only partially reflect the preferences of each user, and as a consequence, more user ratings may be required to determine each item's relevance, reducing data-efficiency of existing relevance estimators [94]. As in practice user interactions are a limited resource, estimators thus also need to be data-efficient, in addition to accounting for bias, variance and consistency.

1.2 Statistical Properties of Estimators for Sampling

An important consideration is that while estimator statistical properties affect relevance estimation in the presence of interaction biases, they may also play a significant role in other settings within RecSys and LTR, particularly across various applications of sampling. Some of these include approximating the computationally costly softmax layer in recommendation [17, 99, 153], estimating document propensities for ranking models [69, 143] and capturing recommendation model uncertainty [134, 169]. Sampling-based methods also form the basis of efficient optimization of ranking models like the Plackett-Luce ranking model [26, 154, 143, 144].

Various sampling-based schemes designed to reduce the computational cost of calculating the exact softmax probability in the presence of large item catalogues have been shown to lead to statistically biased estimates [17, 99]. Furthermore, whilst many sampling approaches in other tasks may be unbiased, they are still also inherently random, and thus their estimates are affected by variance. Analogous to data-efficiency, *sample-efficiency* refers to the relative level of accuracy of an estimator for a given amount of samples [26, 65, 144, 221, 223]. More sample-efficient estimators require fewer samples to obtain the same level of accuracy as the less sample-efficient ones. Importantly, sample-efficiency is also strongly linked to *computational efficiency*, representing the amount of computational resources and time required by a program [17, 153, 223]. In the context of sampling, less sample-efficient estimators may also be less computationally efficient, as they may need to take and subsequently process more samples [144, 223]. And more generally, computationally efficient approaches are generally preferred over the less efficient ones due to reduced runtime as well as lower hardware and energy costs [98, 163, 187].

Computational efficiency, along with estimator variance, also play a particularly significant role in confidence modeling in RecSys. Confidence modeling allows models to represent which predictions they are certain of, and where they are not [205]. Among these methods, prediction uncertainty is frequently captured by the empirical variance of their relevance predictions [134, 169, 205], which can be obtained by sampling multiple predictions from the model and measuring their spread. Where this confidence measure correlates with the size or the frequency of the model’s errors, following only high-confidence predictions may, for instance, be used to guarantee some level of user experience [139]. Nevertheless, confidence modeling has not yet found wide adoption in RecSys, likely in part due to the computational efficiency problems, as many methods rely on heavy sampling routines [169, 205]. Computationally inexpensive methods are thus desirable across a wide range of settings, both within sampling, and in RecSys and LTR in general.

In this thesis we explore ranking and recommendation estimators for learning from user interactions and sampling that account for the above properties, i.e., they address bias, variance and efficiency, as well as inconsistency. We examine when these issues arise, how they affect relevance and other estimation types in RecSys and LTR and propose methods to address them. The main goal of this thesis is thus to highlight the importance of these properties and provide researchers and practitioners with guidelines for their use.

1.3 Research Outline and Questions

We begin our exploration of the impact of bias, variance and efficiency in recommendation and ranking by examining the bandwagon effect [171, 193, 206, 218, 219, 234]. Under this interaction bias, a user’s rating may be affected by the ratings of other users. For instance, a user may be more likely to give a more positive rating to an item if told that other users like it [58, 193]. Importantly, previous work [218, 219, 234] has described the average user rating under by the bandwagon effect as a biased estimate of the true item relevance. Furthermore, it also showed that the effect may increase the number of user interactions required to learn true item relevance. The above observations thus suggest that the bandwagon effect may lead to both statistical bias and low data-efficiency. Nevertheless, the exact statistical nature of the bandwagon effect has not been studied previously.

In Chapter 2, we examine the bandwagon effect and its statistical properties in the context of relevance estimation, prompting our first research question:

RQ1 *Is the bandwagon effect in user interactions a problem of selection bias or is it better described in terms of other statistical properties?*

We argue that the bandwagon effect should not be seen as a problem of statistical bias, as we demonstrate that under the simple form of the bandwagon effect, the sample average of ratings of all users is in fact unbiased. Nevertheless, we prove that the bandwagon effect can make estimators inconsistent, introducing a distinct set of problems for convergence in relevance estimation. Our theoretical analysis investigates the conditions under which the bandwagon effect poses a consistency problem and explores several approaches, including those inspired by IPS, for mitigating these issues.

Within LTR more broadly, IPS approaches have also been used to alleviate position bias in ranking models [28, 90, 118, 145]. Importantly, obtaining an unbiased estimate with IPS requires the knowledge of document-position placement probabilities (a.k.a. document propensities) for all documents for every position. Nevertheless, computing these propensities poses a challenge, as for most ranking models, including the increasingly popular Plackett-Luce (PL) ranking model [154, 128, 143], this requires iterating over all possible rankings, which is generally not computationally feasible. A common solution is thus to approximate them by sampling multiple rankings and using the observed document frequencies per position [69, 143]. Nevertheless, even when using extremely large numbers of sampled rankings,

these estimates often still contain significant estimation errors, suggesting low sample- and computational efficiency of this approach.

In Chapter 3, we investigate the possibility of accurate propensity estimation for PL models in a computationally efficient manner. In particular, we ask the following research question:

RQ2 *Can we accurately and efficiently estimate document placement probabilities for the Plackett-Luce ranking model without sampling?*

We answer this question in the affirmative by proposing the novel *Marginalized Plackett-Luce* (MPL) method to efficiently and accurately calculate document-rank placement probabilities under the PL ranking model by leveraging the connection between PL and Gumbel distributions along with convolutions. Our experiments also confirm that the propensity estimation of MPL is highly accurate and efficient.

OPE methods, including IPS, can then leverage user feedback observed from users interacting with the existing ranking model to estimate the quality of a new one. Importantly, whilst more accurate propensity estimates may lead to improved OPE performance due to reduction in bias, it is also possible that the current sampling-based estimation methods already provide sufficiently accurate propensity estimates at the same computational cost. Furthermore, when combined with variance reduction techniques, such as (propensity) clipping [10, 125, 190], existing OPE methods may not be able to take full advantage of larger amounts of available user interactions.

In Chapter 3 we thus also investigate the applicability of our proposed PL propensity estimation method to OPE, leading to our third research question:

RQ3 *Do our new propensity estimates allow us to reduce bias and improve data-efficiency in off-policy learning to rank?*

Our experiments demonstrate that the propensity estimates of our MPL method lead to OPE errors comparable with that of the asymptotically unbiased sampling-based method yet at over 100 times lower runtime. Furthermore, MPL estimates lead to significantly lower OPE errors and reduced application of clipping compared to the sampling-based estimator with comparable computational complexity, allowing MPL to take advantage of larger amounts of user interaction data. Overall, our work thus allows for the wider use of PL ranking models, with lower bias and higher data-efficiency in off-policy learning to rank.

Importantly, a standard implicit assumption made by LTR methods is that each document in a top- K ranking is also presented in an equal format. However, previous work has shown that users’ perceptions of relevance can be changed by varying presentations, i.e., allocating more vertical

space to some documents to provide additional textual or image information [106]. Furthermore, presentation length can also redirect attention, as users are more likely to notice longer presentations when scrolling through results [48, 130]. Importantly, result presentations of different lengths are already frequently used by various online platforms [176, 236]. Whilst recent works have provided sample-efficient optimization techniques for PL ranking models in the standard ranking setting [26, 128, 143, 144], it is unclear whether they can be applied to the variable presentation length setting, where simultaneously the ordering of documents and their presentation length has to be decided.

In Chapter 4 we formally introduce the variable presentation length setting and examine the applicability of PL models in this setting, leading to our fourth research question:

RQ4 *Can the Plackett-Luce ranking model be extended to simultaneously learn document ordering and presentation size in a computationally and sample-efficient way?*

To answer this question, we propose *variable length Plackett-Luce* (VLPL) – a new family of Plackett-Luce list-wise sampling-based gradient estimation methods for the joint optimization of document ordering and lengths. Our semi-synthetic experiments show that our VLPL-1 and VLPL-2 algorithms can effectively balance the expected exposure and attractiveness of all documents, achieving the best performance across different ranking conditions. Furthermore, we demonstrate that our VLPL-2 model, with its lower variance gradient estimates, is able to use fewer samples compared to VLPL-1. Our results thus suggest that PL can be successfully leveraged to achieve high quality and high sample- and computational efficiency in the variable presentation length setting.

Finally, whilst in most RecSys and LTR settings a common goal is to reduce variance from most sources, variance in the predicted user preferences can also reflect model’s uncertainty in its predictions. In particular, most RecSys do not provide an indication of confidence in their decisions. Therefore, they do not distinguish between recommendations of which they are certain, and those where they are not. Existing confidence methods for RecSys are either inaccurate heuristics, conceptually complex or computationally very computationally expensive [3, 14, 134, 169, 205]. Consequently, real-world RecSys applications rarely adopt these methods, and thus, provide no confidence insights in their behavior.

Chapter 5 examines the possibility of accurate and computationally efficient confidence modeling in RecSys, explored by our fifth and final research question:

RQ5 *Can variance estimation be incorporated into rating prediction algorithms as confidence modeling in a computationally efficient manner?*

For this purpose we propose *learned beta distributions* (LBD), representing user preferences in terms of the Beta distribution, as a simple and practical recommendation method with an explicit measure of confidence without introducing heavy computational costs. At the same time LBD maintains competitive accuracy with existing recommendation methods and also has a significantly stronger correlation between its accuracy and confidence, suggesting that computationally efficient confidence modeling in RecSys through estimated variance is possible.

1.4 Main Contributions

In this section we summarize the main contributions of this thesis. We begin with our algorithmic contributions, followed by our theoretical and resource contributions. Algorithmic contributions refer to the novel estimation and ranking methods introduced as a part of this thesis, theoretical contributions refer to formal proofs and derivations, along with important empirical results, whilst resource contributions refer to the publicly available implementations of each chapter’s code.

1.4.1 Algorithmic contributions

- A novel estimator to reduce the impact of the bandwagon effect on relevance estimation.
- The Marginalized Plackett-Luce (MPL) approach for computationally efficient near-exact estimation of document-position placement probabilities (propensities) under the Plackett-Luce (PL) ranking model.
- The variable length Plackett-Luce (VLPL) algorithm family for sample- and computationally efficient joint optimization of document order and presentation lengths in learning to rank.
- Learned beta distributions (LBD) method for low-complexity and computationally efficient joint modeling of user preference and prediction confidence in rating prediction.

1.4.2 Theoretical contributions

- A formal proof that whilst a simple model of the bandwagon effect may not introduce statistical bias to the sample mean estimator, it may still lead to conditional bias, along with low sample-efficiency and inconsistency.
- A formal proof of a novel necessary condition for consistency of the sample mean under our simple model of the bandwagon effect.
- Derivation of a new relevance estimator and a formal proof that it is conditionally unbiased under our simple model of the bandwagon effect.
- Empirical results confirming that the bandwagon effect can lead to prohibitively large number of user interactions being required for accurate relevance estimation.
- Derivation of the novel document-position placement probability formulation under the PL ranking model, i.e., its order statistic, along with an efficient almost-exact approximation.
- Empirical results demonstrating the accuracy, computational efficiency and its applicability for off-policy evaluation (OPE) of the proposed PL estimation method compared to the existing approach.
- A formal definition of the variable presentation length ranking task.
- A formal proof that the optimal strategy for the variable presentation length ranking task cannot be separated into independent document ordering and length selection steps.
- Empirical results demonstrating that VLPL provides significant performance improvements in the variable presentation length setting over the baseline approaches.
- Empirical results confirming that LBD model has significantly stronger correlation between its uncertainty estimates and observed errors compared to the existing confidence modeling methods.

1.4.3 Resource contributions

- Code for Chapter 2: "The Bandwagon Effect: Not Just Another Bias". Zenodo. <https://doi.org/10.5281/zenodo.17351148>
GitHub: <https://github.com/NKNY/bandwagonictir2022>

- Code for Chapter 3: "Sample-Free Almost-Exact Estimation of Plackett-Luce Propensities for Off-Policy Ranking".
Zenodo. <https://doi.org/10.5281/zenodo.17352314>
GitHub. <https://github.com/NKNY/plpropensitiesecir2026>
- Chapter for Chapter 4: "Learning to Rank with Variable Result Presentation Lengths".
Zenodo. <https://doi.org/10.5281/zenodo.17352329>
GitHub. <https://github.com/NKNY/varlenranksigir2025>
- Code for Chapter 5: "A Lightweight Method for Modeling Confidence in Recommendations with Learned Beta Distributions".
Zenodo. <https://doi.org/10.5281/zenodo.17352353>
GitHub. <https://github.com/NKNY/confidencerecsys2023>

1.5 Thesis Overview

This thesis consists of an introduction chapter, followed by four research chapters and a conclusion chapter. Whilst every research chapter can be read independently, the reader may wish to begin with Chapter 2, as it explicitly examines the theoretical aspects of bias, variance and efficiency in the context of the bandwagon effect, and discusses the connections between them. Chapter 3 addresses the problem of propensity estimation for Plackett-Luce ranking models and introduces MPL, an efficient and almost-exact propensity estimation method. Chapter 4 tackles the task of joint optimization of document order and presentation length in LTR, and proposes the VLPL method family for this problem. Chapter 5 introduces the LBD method family that simultaneously learn user preference along with model confidence for rating prediction. Lastly, in the final Chapter 6 we summarize and discuss the findings of this thesis.

1.6 Origins

This section lists the contributions that form the basis for the chapters of this thesis. The following contributions were the same across all papers. Norman Knyazev came up with the main research ideas, theoretical and computational methods, performed all the experiments and wrote the majority of the text. Harrie Oosterhuis led the discussions on the paper structure and provided significant contributions to the text. Djoerd Hiemstra provided valuable feedback on the papers.

- **Chapter 2:** N. Knyazev, H. Oosterhuis. The Bandwagon Effect: Not Just Another Bias. In *ICTIR '22: Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 243 – 253, 2022.
- **Chapter 3:** N. Knyazev, H. Oosterhuis. Sample-Free Almost-Exact Estimation of Plackett-Luce Propensities for Off-Policy Ranking. In *ECIR '26: European Conference on Information Retrieval*, 2026.
- **Chapter 4:** N. Knyazev, H. Oosterhuis. Learning to Rank with Variable Result Presentation Lengths. In *SIGIR '25: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2245 – 2254, 2025.
- **Chapter 5:** N. Knyazev, H. Oosterhuis. A Lightweight Method for Modeling Confidence in Recommendations with Learned Beta Distributions. In *RecSys '23: Proceedings of the 17th ACM Conference on Recommender Systems*, pages 306 – 317, 2023.

Chapter 2

The Bandwagon Effect: Not Just Another Bias

Optimizing recommender systems based on user interaction data is mainly seen as a problem of dealing with selection bias, where most existing work assumes that interactions from different users are independent. However, it has been shown that in reality user feedback is often influenced by earlier interactions of other users, e.g., via average ratings, number of views or sales per item, etc. This phenomenon is known as the *bandwagon effect*. Importantly, the statistical nature of the bandwagon effect has not been previously examined in-depth. To address this gap, our first thesis research question **RQ1** thus asks:

RQ1 *Is the bandwagon effect in user interactions a problem of selection bias or is it better described in terms of other statistical properties?*

In contrast with previous literature, we argue that the bandwagon effect should not be seen as a problem of statistical bias. In fact, we prove that this effect leaves both individual interactions and their sample mean unbiased. Nevertheless, we show that it can make estimators inconsistent, introducing a distinct set of problems for convergence in relevance estimation. Our theoretical analysis investigates the conditions under which the bandwagon effect poses a consistency problem and explores several approaches for mitigating these issues. This work aims to show that the bandwagon effect poses an underinvestigated open problem that is fundamentally distinct from the well-studied selection bias in recommendation.

2.1 Introduction

One of the main challenges in optimizing recommender systems (RecSys) is correctly capturing user preferences from historical interaction data [74, 80, 82, 164, 191]. A widely adopted approach is to train a model to predict known ratings and then apply it to previously unobserved user-item pairs [155, 189, 198]. However, in the last decade it has become apparent that observed interactions are often a very skewed representation of the actual user preferences [32, 155, 188]. In particular, ratings are known to be affected by various forms of *selection* bias, such as *popularity* bias [188], *positivity* bias [155] or *position* bias [145]. Popularity bias, for instance, occurs because as most interactions involve only a small number of very popular items, the patterns in observed ratings may not necessarily extrapolate to the data as a whole [32, 155, 178, 188]. It is now well understood that ignoring statistical biases during optimization is likely to lead to suboptimal recommendation performance [32, 76, 178] and numerous solutions have been proposed to address various forms of selection bias in recommendation data [76, 84, 167, 168, 175, 178, 208, 224].

In this work, we will investigate the related but fundamentally distinct *bandwagon effect*, also referred to as *herding* or *conformity bias*. Despite being observed by multiple studies, this often-overlooked effect occurs because users tend to follow the behavior and opinions of others [58, 182]. That is, if a user is told that many other users have liked an item, they themselves become more likely to give a positive rating. Conversely, when a user is told that many other users dislike an item, their own rating is likely to become lower. Several studies have observed this behavior when exposing the average rating from previous users [5, 43, 58, 193] and analogous trends have also been reported for implicit feedback [70, 157]. Consequently, the bandwagon effect can pose a serious problem when trying to infer user preferences from interaction data.

Despite its well-known existence, there have only been limited attempts at reducing the influence of the bandwagon effect [123, 207, 229] and few authors have theoretically analyzed it in statistical terms [218, 220]. Our work will build on a mathematical model proposed by Xie *et al.* [218, 220], who in line with earlier literature [207, 229] interpret the bandwagon effect as a problem of statistical bias.

In this work, we provide a rigorous theoretical analysis of the bandwagon effect and its statistical impact on rating and click-through rate (CTR) estimation. In stark contrast with previous work, we prove that a simple form of bandwagon effect of Xie *et al.* [220] does *not* introduce statistical bias, as both individual ratings and their sample mean yield unbiased

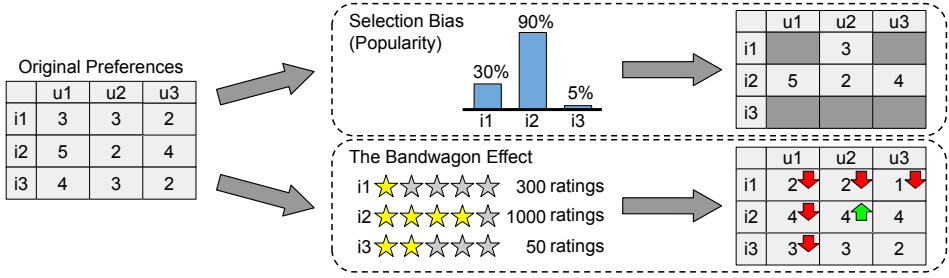


Figure 2.1: Impact of selection bias and bandwagon effect on average rating estimation. Under popularity bias, users are less likely to interact with unpopular items, leaving most of their ratings unobserved. Under the bandwagon effect, new ratings are skewed in the direction of previous users’ opinion, with larger group size eliciting a stronger effect.

estimates under the effect. Nonetheless, our analysis also reveals that the bandwagon effect can create serious problems for convergence by making the sample mean inconsistent. Therefore, despite not being a bias issue, the bandwagon effect can still introduce significant errors during optimization. To understand when this may pose a problem, we theoretically prove a condition under which inconsistency is guaranteed to arise for the sample mean. Furthermore, our theoretical and simulated results show that even when an estimator is consistent, the convergence can still be so slow that finding the true user preferences becomes practically infeasible. We therefore investigate whether existing debiasing methods can be adapted to handle these convergence issues. However, as none of our methods are able to combine theoretical guarantees with robust empirical performance, we conclude that the bandwagon effect currently remains an open problem that needs more attention in future work.

2.2 Related Work

2.2.1 Debiasing interactions for recommendation

Learning from user interactions has a long tradition in the field of recommender systems [73, 112, 132, 155, 178]. A common approach is to cast the recommendation problem as a rating prediction task where the goal is to predict the ratings a user would give to items [73, 112, 132]. However, one should not straightforwardly base the optimization on the ratings observed in interaction logs as they are often not representative of actual user preferences due to selection bias [73, 132]. The most well-known and

well-studied form of bias in rating data is popularity bias: a small selection of popular items receives the large majority of user ratings and is thus grossly overrepresented [32, 155, 188]. Positivity bias is also prevalent: users may be more likely to leave a rating on items they like; consequently, low ratings become underrepresented in the resulting interaction data [155]. As seen in Figure 2.1 (top), if popular items that attract the majority of ratings also happen to be more liked by users (i.e. receive higher ratings), the average rating estimate from observed ratings becomes skewed towards a higher value. Schnabel et al. [178] showed that not accounting for selection bias makes the optimization procedure focus on statistically overrepresented user-item combinations. As a solution, they propose to use inverse propensity scoring (IPS) to unbiasedly estimate and optimize the performance of a system [121]. Debiasing for recommendation has become a very popular topic in recent years: for example, Wang et al. [208] propose a doubly robust estimator that combines IPS with regression estimates; Saito et al. [168] adapted the IPS matrix factorization approach to debias click data instead of ratings; and Jeunen and Goethals [84] incorporate a prior distribution to avoid overestimation during prediction.

Selection bias is also widespread in clicks. Its most well-studied form is position bias, which occurs when the position at which an item is displayed heavily impacts the attention the item will receive, and thus how many users will click on it [45]. Similar to ratings, IPS is a popular method to correct for selection bias in clicks [90, 142].

The problem of selection bias in explicit feedback, i.e. ratings, and implicit feedback, e.g., clicks, is thus well recognized and has been addressed extensively by the existing literature [73, 84, 90, 112, 132, 142, 155, 168, 208, 178]. This can lead to the impression that learning from interactions is mainly a selection bias problem. Nevertheless, our current work on the bandwagon effect will show that this is definitely not the case.

2.2.2 The bandwagon effect

The *bandwagon effect*, or *herding*, describes a phenomenon whereby individuals tend to exhibit a greater affinity towards an item when they believe other individuals also like it [158, 182]. The decisions of earlier individuals can propagate through *information cascades* to individuals also taking the same decision later, leading to snowballing effects, as well documented in social psychology, economics and finance, and, more recently, web information systems [21, 34, 43, 158].

Online platforms frequently include various social cues next to items to assist users with their choice, e.g., the number and the average value

of previous user ratings. Numerous studies have shown that users actively leverage such information when making purchasing or consumption decisions [5, 51, 58, 60, 70, 75, 101, 157, 170, 171, 193]. Hanson and Putler [70] found that artificially inflating download counts for some of the products on a software download service led to those products receiving more subsequent downloads than their unaltered counterparts. Similarly, Qin et al. [157] report that Chrome Web Store users rarely click on items displayed with a low download count. Flanagin and Metzger [58] noted that users exposed to the average rating also rated movies closer to that average. Analogously, Sundar et al. [193] found that participants reported higher purchasing intent for high-rated items over low-rated items and that this effect arose from an improved perception of quality and value. Both studies also reported that the strength of the effect was positively correlated with the item’s number of ratings, consistent with earlier works associating larger group size with stronger herding [11, 75, 115]. Altogether, these observations suggest that users incorporate the experience of others into their own judgements and the strength of this effect is correlated with the size of the group.

This phenomenon can have a number of negative consequences. On an individual level, the bandwagon effect can be so strong that users may act against their own interests: Dholakia and Soltysinski [51] highlight that eBay users frequently bid on items that already have many bids, whilst ignoring comparable or superior listings with fewer bids. However, another consequence of the bandwagon effect – and the focus of this paper – is the difficulty it brings to estimating item relevance from user interactions. In the pioneering experiment of Salganik et al. [171], a large number of sequentially arriving participants browsed a randomly shuffled grid of the same 48 songs and could interact with songs of their choice. The participants were randomly assigned to one of nine groups, with users in groups 2 – 9 additionally presented with the download count from previous users *from their group* alongside each song. The authors observed that songs had a higher probability of being downloaded in a group where they already had a high number of downloads. In contrast, a song shown next to a low download count was less likely to get downloaded compared to the same song in groups where it had a high or no presented download count (group 1). Moreover, with the exception of the very best and worst songs, as determined by the control group 1, the final ranking according to download counts varied wildly across different groups. In their follow-up work, Salganik and Watts [170] further showed that the bandwagon effect may be so strong that artificially altering the presented download count can make unpopular songs popular and vice-versa, with only the best songs eventually recovering their true

relevance ranking. Taken together, the above results suggest that under the bandwagon effect even a large amount of user feedback is not guaranteed to yield an accurate estimate of true user preference. This is also demonstrated in Figure 2.1 (bottom): users’ true preferences are intertwined with the ratings of earlier users and this influence grows as the number of ratings increases. As a consequence, even when assuming no selection bias, true average rating cannot be determined via the sample mean.

Recently, there has been some interest in modelling herding and disentangling the effects of the crowd on the individual. Wang et al. [207] propose an estimator maximizing a surrogate function of the likelihood of the data and demonstrate the effectiveness of their approach for prediction of subsequent ratings. Zhang et al. [229] show that subsequent rating estimation can be further improved by considering that some users may actually choose to distance their opinion from the crowd. Liu et al. [123], Zhang et al. [231], Wei et al. [213] all propose various extensions to classical recommender models by using the rating mean and/or count as additional inputs. Finally, Xie, Zhong and colleagues formulate the bandwagon process as either a rating reweighting or hidden state prediction problem and show theoretically that the bandwagon effect slows down the convergence of the sample mean [218, 220, 234].

Notably, most of the above works treat the bandwagon effect as a problem of statistical bias. Furthermore, to the best of our knowledge, only the works of Xie et al. [218, 220] and Zhong et al. [234] attempt to establish a link between the bandwagon effect and statistical theory. Our theoretical analysis will contrast with the existing literature by showing that the bandwagon effect can affect relevance estimation without introducing statistical bias.

2.3 Background

2.3.1 Preliminaries

Before we address the bandwagon effect specifically, this section will introduce our problem setting and the statistical properties that our analysis will consider. For the sake of generality, we focus on *unpersonalized* rating estimation, as we expect the issues observed for this simple task to also apply to the more involved forms of recommendation. Furthermore, our findings are not specific to recommendation and can be readily applied to other problems involving relevance estimation such as CTR estimation.

In our generic unpersonalized recommendation setting, the goal is to estimate the average preference of all users for each item. We assume that this value $p \in (0, 1)$ exists for each item and represents the percentage of users that truly like the item. Furthermore, users interact with the items by assigning them binary ratings s.t. $r_i \in \{0, 1\}$ is the i 'th rating for an item. While we refer to r_i as a rating, it can equally well represent many other binary interactions such as *clicks*, *purchases*, *shares*, etc. As discussed in Section 2.2.2, in real systems users are often able to see how many other users have rated an item and what their average rating was, such that past ratings can affect the behaviour of the current user. We define p to be the *true preference* or the percentage of users who like an item when they have no knowledge of the preferences of any other users. Our goal is then to estimate p for a *single* item from the given set of observed ratings $\{r_1, r_2, r_3 \dots\}$.

Let $\hat{p}_n$ indicate an estimate of p based on n observed ratings. Our analysis will investigate three important beneficial properties that we generally desire from such an estimate. To start, we normally want the estimate to be unbiased. In other words, in expectation it should be equal to the true value of p :

$$\text{Unbiased}(\hat{p}_n) \longleftrightarrow \mathbb{E}[\hat{p}_n] = p. \quad (2.1)$$

Understandably, bias and unbiasedness have received a lot of attention in previous selection bias literature, as it is desirable to avoid systematic errors in rating estimation. However, unbiasedness on its own does not guarantee that an estimator is actually usable in practice.

The second important and often overlooked property that we may want from an estimator is consistency. An estimator is considered consistent if the estimate converges in probability to the true value of p as the size of available data approaches infinity:

$$\text{Consistent}(\hat{p}_n) \longleftrightarrow \lim_{n \rightarrow \infty} \hat{p}_n = p. \quad (2.2)$$

Whilst at a first glance unbiasedness and consistency may appear to refer to the same property, this is certainly *not* the case. For example, estimators can be biased yet consistent [196] or unbiased and inconsistent. As illustrated in Figure 2.2, the problem with an unbiased inconsistent estimator is that its confidence intervals do not converge to a zero error even as the size of the data increases indefinitely. This means its estimation errors will not disappear even after gathering more data, thus preventing accurate convergence.

Nevertheless, whilst consistency guarantees that eventually the estimate will be correct, the amount of data required for convergence may still be

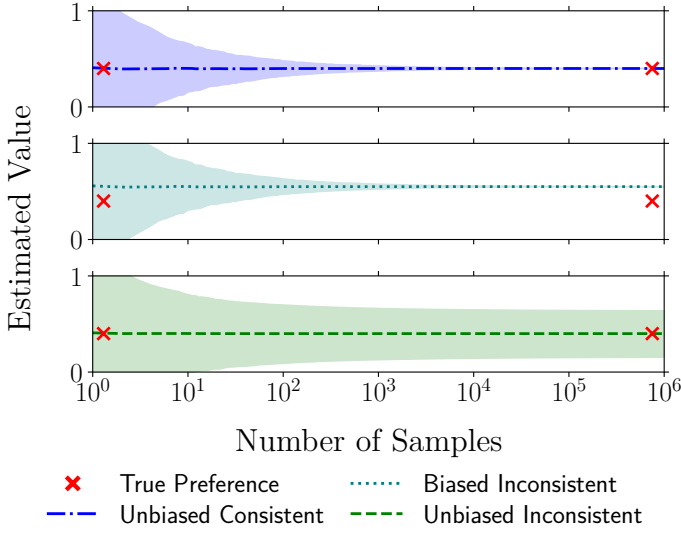


Figure 2.2: Three estimators with varying bias and consistency; bold lines show the mean estimate over 1,000 independent runs, shaded areas indicate 90% confidence intervals. Top - unbiased and consistent: its expected value and point of convergence are both correct. Middle - biased and inconsistent: its expected value and point of convergence are both incorrect. Bottom - unbiased and inconsistent: its expected value is correct but its confidence interval will never converge.

unattainable in practice. Therefore, the final property we will consider is (data) efficiency, measured by the expected squared error (equivalent to variance for unbiased estimators):

$$\text{Efficiency}(\hat{p}_n) = \mathbb{E}[(\hat{p}_n - p)^2]. \quad (2.3)$$

Overall, efficiency represents a more practical side of an estimator: the error we can expect it to have w.r.t. the true value p after n samples. The effect the number of samples has on this measure can be seen as the speed of convergence. While most existing work on learning from user interactions focuses on correcting for bias, they often also apply propensity clipping or self-normalization which introduce a little bias to greatly reduce variance [90, 145, 174, 178, 212].

Our theoretical investigation will consider how the bandwagon effect impacts all three properties: bias, consistency and efficiency.

2.3.2 A formal model of the bandwagon effect

The mathematical model of the bandwagon effect we will use in this paper is a simplified version of that of Xie et al. [218, 220]. This model represents a sequential unpersonalized recommendation scenario where users rate the item in succession and one at a time. It assumes there are two factors that influence the probability of a positive value for the n 'th rating ($r_n = 1$): (i) p , the true proportion of users who would like the item given that they do not know anything of the other users' preferences (no bandwagon effect); and (ii) $\bar{p}_{n-1}$, the mean of the ratings that have taken place previously. The rating probability is modelled as a linear combination of these two factors:

$$P(r_n = 1) = \lambda_n p + (1 - \lambda_n) \bar{p}_{n-1}, \quad \bar{p}_{n-1} = \frac{1}{n-1} \sum_{i=1}^{n-1} r_i. \quad (2.4)$$

$\lambda_n \in [0, 1]$ captures the degree to which users adjust their behavior to match that of the crowd. For example, $\forall \lambda_n = 1$ indicates a process with no bandwagon effect and $\forall n > 1 : \lambda_n = 0$ a process where true preferences play no role except for the very first user. As previously mentioned, earlier works have shown a positive correlation with the strength of herding and the size of the group [11, 75, 115]. We capture this increase in effect by assuming that the influence of the sample mean cannot decrease as the number of ratings increases:

$$\lambda_1 = 1 \wedge (\forall n > 1 : 0 \leq \lambda_n \leq \lambda_{n-1}). \quad (2.5)$$

In addition, this assumption states that the first rating is completely unaffected by the bandwagon effect (i.e. $\lambda_1 = 1$), as the first user has no information about the preferences of others. We have chosen the above formulation over others (e.g., [207, 229]) due to its amenability to statistical analysis whilst still capturing the key aspects of the bandwagon effect.

Before we begin our theoretical analysis, we will briefly discuss two claims from earlier work by Xie and Zhong [218] so that they can later be compared with our findings in Section 2.4. First, Xie and Zhong refer to the bandwagon effect as *persuasion bias*, by which they mean that the bandwagon effect biases the rating distribution such that it no longer represents the true underlying preference. Second, they show that as long as the lower bound of λ_n is above 0, the bandwagon effect will eventually be eliminated since the sample mean $\bar{p}_n$ is guaranteed to converge to the true parameter value:

$$\inf_{n \in N^+} \lambda_n > 0 \longrightarrow \lim_{n \rightarrow \infty} \bar{p}_n = p. \quad (2.6)$$

Our theoretical analysis will take another critical look at the bias and consistency of $\bar{p}_n$. Surprisingly, our findings regarding bias heavily contrast with previous work on the bandwagon effect.

2.4 Theoretical Analysis: The Bandwagon Effect

Now that Section 2.3.2 has introduced our mathematical model of the bandwagon effect, we will begin our theoretical analysis of its effects on bias, consistency and efficiency. First, we want to again consider the claim of previous work that the bandwagon effect introduces bias to the rating distribution [218, 220]. From our bandwagon model (Equation 2.4) it appears that the probability of a positive rating is indeed a distortion of the true preference p . However, Theorem 2.A.1 proves that the expected values of all sample means $\bar{p}_n$ and all ratings r_n are actually equal to the true preference:

$$\forall n : \mathbb{E}[\bar{p}_n] = \mathbb{E}[r_n] = p. \quad (2.7)$$

In other words, under our model, both the sample means and actual ratings provide *unbiased* estimates of the true preference, corresponding to either top or bottom, but not center of Figure 2.2. This finding contrasts heavily with how previous work has approached the bandwagon effect as a bias problem [123, 207, 213, 218, 220, 229, 230, 234] and similarly, how learning from user interactions is generally also framed as a bias problem [73, 84, 90, 112, 132, 142, 155, 168, 178, 208]. The bandwagon effect *may* be seen as an increased impact of earlier interactions on the estimate of the item's relevance. However, in contrast with selection bias, there is no predetermined rating value that is disproportionally represented for the purposes of average relevance estimation or loss minimization.

Nevertheless, the lack of bias does not mean that the bandwagon effect does not introduce other problems. As discussed in Section 2.3.1, there are other statistical properties besides unbiasedness that one should consider. Theorem 2.A.4 proves the following efficiency of the sample mean, i.e. its expected (squared) error:

$$\mathbb{E}[(\bar{p}_n - p)^2] = p(1 - p) \left(\frac{1}{n^2} + \sum_{i=1}^{n-1} \frac{1}{i^2} \prod_{j=i+1}^n \frac{(j-1)(j+1-2\lambda_j)}{j^2} \right).$$

Consistent with our prior discussion, the above formulation reveals that when $\forall \lambda_n = 1$, the variance of $\bar{p}_n$ is equivalent to the variance of sampling from the

true preference p . Conversely, when $\forall n > 1 : \lambda_n = 0$, the expected error does not decrease after the first interaction i.e. $\forall n : \mathbb{E}[(\bar{p}_n - p)^2] = \mathbb{V}[\bar{p}_1] = p(1-p)$. Obviously, an error that never decreases provides a serious convergence problem.

To investigate whether the sample mean is consistent, we look at the value of the expected error in the limit of infinite samples. Theorem 2.A.5 proves the following expected error in the limit:

$$\lim_{n \rightarrow \infty} \mathbb{E}[(\bar{p}_n - p)^2] = \lim_{n \rightarrow \infty} p(1-p) \left(\sum_{i=1}^{n-1} \frac{1}{i(i+1)} \prod_{j=i+1}^{n-1} \left(1 - \frac{2\lambda_j}{j+1} \right) \right).$$

Importantly, if the expected error is not zero in the limit as in Figure 2.2 (bottom), the estimator is not consistent and vice-versa. Based on this fact, we prove a novel necessary condition for the consistency of $\bar{p}_n$ in Theorem 2.A.6:

$$\forall \epsilon : \lim_{n \rightarrow \infty} \Pr(|\bar{p}_n - p| > \epsilon) = 0 \longrightarrow \lim_{n \rightarrow \infty} \sum_{i=1}^n \lambda_i^2 = \infty. \quad (2.8)$$

In other words, if the sample mean is consistent (corresponds to Figure 2.2, top), the sum of λ_i^2 terms is unbounded. Equivalently, convergence of $\lim_{n \rightarrow \infty} \sum_{i=1}^n \lambda_i^2$ leads to inconsistency. We note that whilst the sufficient condition of Xie and Zhong [218] (Equation 2.6) highlights the situations where we can guarantee consistency, it cannot be used to guarantee inconsistency. For instance, one cannot use it to show whether $\lambda_n = n^{-1}$ leads to issues with convergence as $\inf_{n \in \mathbb{N}^+} \lambda_n = 0$. On the other hand, as $\lim_{n \rightarrow \infty} \sum_{i=1}^n i^{-2}$ is finite, Theorem 2.A.6 shows that $\bar{p}_n$ is inconsistent. Our condition can thus be seen as complementary to the sufficient condition of earlier work, qualifying the set of λ sequences where inconsistency is certain.

We can also leverage Theorem 2.A.5 to quantify the asymptotic expected absolute error for any specific λ values. For instance, Proposition 2.A.8 proves that for $\lambda_i = c^{i-1}$ with $c \in (0, 1)$:

$$\begin{aligned} \lim_{n \rightarrow \infty} \mathbb{E}[|\bar{p}_n - p|] &\geq 2p(1-p) \exp(-c\phi(c, 1, 2) - c^2\phi(c^2, 2, 2)) \\ \text{s.t. } \phi(z, s, a) &\equiv \sum_{k=0}^{\infty} \frac{z^k}{(a+k)^s}. \end{aligned} \quad (2.9)$$

Similar bounds may also be derived for other λ formulations.

Our analysis proved that neither the sample mean $\bar{p}_n$ nor the individual ratings r_n are biased, a finding that appears to heavily contrast with earlier

work that casts the bandwagon effect as a problem of bias [123, 207, 213, 218, 220, 229, 230, 234]. We speculate that the bias framing may at least in part stem from the dependency between ratings. More specifically, conditioning the expectation of $\bar{p}_n$ and r_n on an earlier $\bar{p}_m$ by Theorem 2.A.1 and Proposition 2.A.7 leads to:

$$\forall n > m > 1 : \mathbb{E}[\bar{p}_{n-1} \mid \bar{p}_m] - p = (\bar{p}_m - p) \prod_{i=m+1}^{n-1} \left(1 - \frac{\lambda_i}{i}\right), \quad (2.10)$$

$$\forall n > m > 1 : \mathbb{E}[r_n \mid \bar{p}_m] - p = (1 - \lambda_n) \mathbb{E}[\bar{p}_{n-1} - p \mid \bar{p}_m]. \quad (2.11)$$

This can be seen as a *conditional* bias, i.e. conditioned on any $\bar{p}_m$ being incorrect, all subsequent ratings and sample means are biased:

$$\forall n > m \geq 1 : \mathbb{E}[\bar{p}_n \mid \bar{p}_m \neq p] \neq p. \quad (2.12)$$

We speculate that previous work has interpreted this conditional bias as bias in the rating distribution and sample mean. Nonetheless, as discussed above, our bandwagon effect model does not lead to systematic errors in relevance estimation. In Section 2.7, we will argue that therefore the bandwagon effect is best interpreted as a problem of consistency and not statistical bias.

To summarize the main findings of our theoretical analysis: we have proven that the bandwagon effect does not introduce bias to the rating distribution but can make the sample mean inconsistent. We have also derived a necessary condition for consistency, expanding on prior work. Lastly, our formulation of the expected error can quantify the asymptotic error for any specific λ values.

2.5 Bandwagon Effect Mitigation

2.5.1 Correcting for the bandwagon effect

We have established that while the bandwagon effect does not make ratings biased, it can lead to convergence problems. As possible solutions, inspired by analogous approaches in click debiasing [41, 204], we will propose an affine and a maximum likelihood estimator.

Firstly, similar to earlier unbiased Learning to Rank work [204], we notice that the model in Equation 2.4 is invertible. We therefore propose an intermediate affine estimate based on a single rating:

$$\hat{r}_i = \frac{r_i + (1 - \lambda_i)\bar{p}_{i-1}}{\lambda_i}. \quad (2.13)$$

Theorems 2.A.9 and 2.A.10 prove that our intermediate estimator is both unbiased and conditionally unbiased w.r.t. previous ratings:

$$\forall i > j \geq 1 : \mathbb{E}[\hat{r}_i \mid \hat{r}_j] = \mathbb{E}[\hat{r}_i \mid \bar{p}_j] = \mathbb{E}[\hat{r}_i] = p. \quad (2.14)$$

Therefore, using $\hat{r}_i$ instead of r_i in an averaging estimator may allow us to speed up convergence by incorporating only the information about the user's true preference without reinforcing the previous $i - 1$ ratings. With this in mind, we propose our affine estimator that uses all of the observed ratings via a weighted mean:

$$\hat{p}_n = \frac{1}{\sum_{i=1}^n \omega_i} \sum_{i=1}^n \omega_i \hat{r}_i, \quad \text{s.t. } \forall 1 \leq i \leq n : \omega_i > 0. \quad (2.15)$$

Clearly $\hat{p}_n$ is also unbiased as it is simply a weighted average over the unbiased estimates $\hat{r}_i$. Additionally, Theorem 2.A.14 proves it is consistent if $\sum_{i=1}^n (\omega_i / \lambda_i)^2$ grows slower in n than $(\sum_{i=1}^n \omega_i)^2$, with Corollary 2.A.15 providing specific conditions $\forall \omega_i = 1$ and $\forall \omega_i = \lambda_i$. Furthermore, Theorem 2.A.13 proves $\hat{p}_n$ has the following variance:

$$\mathbb{V}[\hat{p}_n] = \frac{1}{(\sum_{i=1}^n \omega_i)^2} \sum_{i=1}^n \frac{\omega_i^2}{\lambda_i^2} \left(p(1-p) - (1-\lambda_i)^2 \mathbb{V}[\bar{p}_{i-1}] \right). \quad (2.16)$$

Unfortunately, interpreting $\mathbb{V}[\hat{p}_n]$ is not straightforward as in addition to the choice of ω_i it depends on both the rating values as well as their order. Analogous to its use in selection bias [90, 190], one can also apply clipping to the bandwagon effect, e.g., $\hat{\lambda}_i = \max(\lambda_i, \tau)$ s.t. $\tau > 0$. However, in contrast with IPS, $\hat{r}_n$ and $\hat{p}_n$ would remain unbiased, though not conditionally unbiased (Corollary 2.A.11): clipping or misestimation s.t. $\hat{\lambda}_i \neq \lambda_i$ may still affect the rate of convergence.

Secondly, inspired by the approach of click models [41], we also propose the approximate maximum likelihood estimator p_n^* , obtained via iterative application of Newton's method [62] to maximize the log-likelihood of the data L :

$$L = \sum_{i=1}^n r_i \ln(\lambda_i p_n^* + (1-\lambda_i) \bar{p}_{i-1}) + (1-r_i) \ln(1 - (\lambda_i p_n^* + (1-\lambda_i) \bar{p}_{i-1})). \quad (2.17)$$

Because both the likelihood and the solution found by Newton's method appear intractable, we are unable to prove this estimator is unbiased. Nonetheless, maximum likelihood estimators have been very effective when applied to user interactions in the past [41, 235] and can be consistent under certain feasible conditions [46].

2.5.2 Bandwagon effect estimation

Both our proposed estimators require accurate λ values to capture the bandwagon effect well. Inspired by the experiments of Salganik et al. [171], we propose the following parameter estimation scheme that can be applied when dealing with one or more items affected by the same bandwagon effect.

We will assume that λ values are shared across a subset of $K \geq 1$ items and that all users are randomly assigned to one of M bins s.t. users give n ratings per item and are only presented with rating information regarding users in the same bin. Let p^k indicate the true preference of all users w.r.t. item k (analogous to p) and $\hat{\lambda}_i$ an estimate of λ_i . The log-likelihood of this data is then:

$$L = \sum_{k=1}^K \sum_{m=1}^M \sum_{i=1}^n r_{kmi} \ln (\hat{\lambda}_i p^k + (1 - \hat{\lambda}_i) \bar{p}_{i-1}^{km}) + (1 - r_{kmi}) \ln (1 - \hat{\lambda}_i p^k - (1 - \hat{\lambda}_i) \bar{p}_{i-1}^{km}). \quad (2.18)$$

The maximum likelihood estimate can be found by maximizing L , for instance through the iterative application of Newton's method. In order to get $\hat{\lambda}_i$ values for $i > n$, we propose fitting them to a curve, e.g. $\hat{\lambda}_i = a + (1 - a) \cdot b^{i-1}$ where $a \in [0, 1]$ and $b \in (0, 1)$. Note that as p^k may be unknown, one may instead estimate λ_i values by using its unbiased estimate $\bar{p}^k = \frac{1}{Mn} \sum_{m=1}^M \sum_{i=1}^n r_{kmi}$.

2.6 Experiments and Results

2.6.1 Experimental setup

To investigate how the bandwagon can affect convergence and whether our proposed estimators can mitigate negative effects, we performed several experiments with simulated user interactions.

For an item with true relevance $p = 0.4$, we generated user interactions under three conditions: a *no bandwagon* setting where $\forall \lambda_i = 1$, a *weak bandwagon* setting where $\lambda_i = 0.6 + 0.4 \cdot 0.9^{i-1}$ and a *strong bandwagon* setting where $\lambda_i = 0.1 + 0.9 \cdot 0.95^{i-1}$. Our comparison includes the following four estimators: (i) the sample mean $\bar{p}_n$ (Equation 2.4); (ii) affine mean $\hat{p}_n$, which is the affine estimator (Equation 2.15) with uniform weights $\forall i : \omega_i = 1$; (iii) affine weighted $\tilde{p}_n$, which is the affine estimator but with the weights $\forall i : \omega_i = \lambda_i$; and (iv) the maximum likelihood estimate p_n^* (Equation 2.17). For statistical significance, the entire sampling process was repeated 1,000 times independently, all our graphs include 90% confidence bounds over these independent runs.

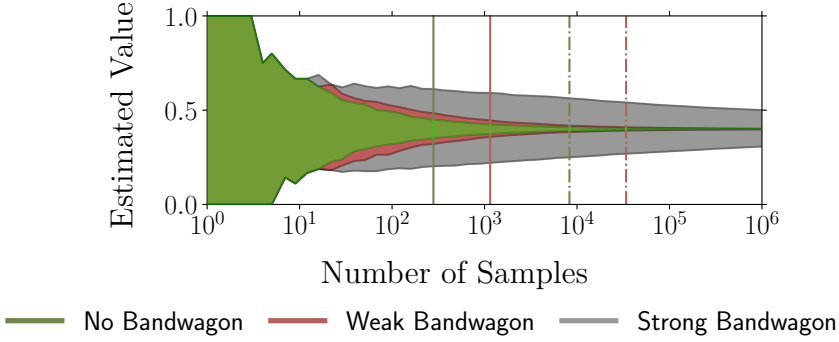


Figure 2.3: Convergence of $\bar{p}_n$ under various levels of bandwagon. Solid and dashed lines indicate points where a 90% confidence interval over 1,000 independent runs is within 0.05 and 0.01 of p respectively, $p = 0.4$.

Furthermore, to investigate how sensitive our estimators are to misestimated $\hat{\lambda}$ values, we computed the estimator values with both the actual λ values and estimated $\hat{\lambda}$ values. We followed the estimation scheme of Section 2.5.2, generating additional ratings with $K = 1$, $p_1 = 0.4$, $M = 20$ and $n = 100$ to represent a plausible production-level scenario [171]. The result obtained from fitting a curve: $\hat{\lambda}_i = a + (1 - a) \cdot b^{i-1}$ where $a \in [0, 1]$ and $b \in (0, 1)$ to the data was: $\hat{\lambda}_i = 0.33 + 0.67 \cdot 0.92^{i-1}$. We only report the results for misestimated $\hat{\lambda}$ values in the strong bandwagon setting, due to high similarity with the weak setting.

2.6.2 Estimator bias and convergence

Our results are visualized in Figure 2.3, which displays the convergence rate of the sample mean under various levels of bandwagon effect, and Figure 2.4, which shows the behavior of all estimators with known λ in the weak and strong bandwagon settings.

First, we consider the influence of the bandwagon effect on the sample mean ($\bar{p}_n$): from Figure 2.3 it is immediately clear that the effect's strength has a significant impact on the rate of convergence. Whilst under no bandwagon effect the confidence intervals are within 0.05 of p after fewer than 300 samples and within 0.01 after roughly 5,000 samples, under the weak bandwagon effect these numbers exceed 1,000 and 25,000 respectively. Under the strong bandwagon effect the confidence intervals increase even further, exceeding 0.05 after 10^6 samples. However, in all cases the confidence bounds still appear to decrease steadily, suggesting that $\bar{p}_n$ will converge

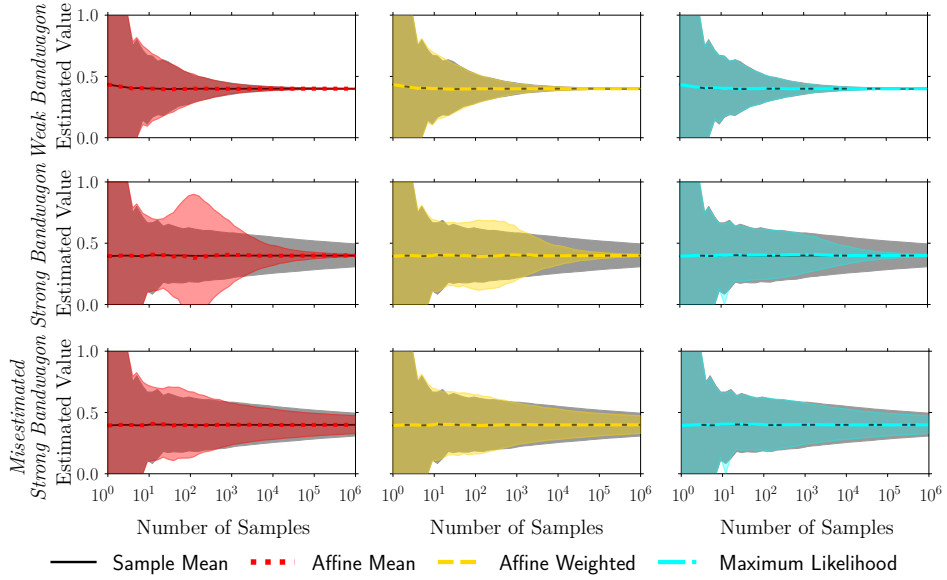


Figure 2.4: Convergence of $\hat{p}_n$ (left), $\tilde{p}_n$ (center) and p_n^* (right) compared to $\bar{p}_n$ with $p = 0.4$ and $\lambda_i = 0.6 + 0.4 \cdot 0.9^{i-1}$ (top), $\lambda_i = 0.1 + 0.9 \cdot 0.95^{i-1}$ (center and bottom) and misestimated $\hat{\lambda}_i = 0.33 + 0.67 \cdot 0.92^{i-1}$ (bottom); bold lines show the mean estimate over 1,000 independent runs, shaded areas indicate 90% confidence intervals.

eventually, as also predicted by the sufficient condition for consistency (Equation 2.6). Nevertheless, it is also clear that the bandwagon effect introduces a significant loss in data efficiency. In practice, this could mean that an infeasibly large amount of data is required to get a decent estimate under a strong bandwagon effect.

Next, we examine the impact on the bias of the sample mean and other estimators. As shown in the first two rows of Figure 2.4, even prior to convergence, all estimators including $\bar{p}_n$ make negligible errors in expectation, strongly suggesting that they are all unbiased. Whilst the unbiasedness was already theoretically proven for $\bar{p}_n$ and the affine estimators $\hat{p}_n$ and $\tilde{p}_n$, it is surprising to see that even at low numbers of samples the expected error of the maximum likelihood estimate p_n^* appears negligible. Our results thus seem to indicate that all the estimators are unbiased or have an extremely small bias, further confirming that the bandwagon effect is not a bias problem.

We then focus on the weak bandwagon condition (Figure 2.4 top). Here, all estimators converge to p , at a rate similar to that of $\bar{p}_n$. That the sample

mean and affine estimators converge is not surprising, as the setting meets the sufficient condition for consistency. However, in this scenario none of the estimators actually provide a clear improvement over the sample mean, indicating that they are unable to apply significant beneficial corrections to the bandwagon effect. The lack of improvement over $\bar{p}_n$ is particularly concerning as this indicates that our estimators are similarly affected by the loss of data efficiency induced by the bandwagon effect.

Nonetheless, one may expect the convergence to slow down even further in the strong bandwagon setting shown in the middle row of Figure 2.4. Here, both the affine and maximum likelihood estimators are actually able to converge on the true value much sooner than the sample mean. However, confidence bounds of the former also become wider after roughly 10 sampled ratings and stay wider until roughly 1,000 samples, with the weighting of $\tilde{p}_n$ seemingly reducing this effect when compared with $\hat{p}_n$. Thus during this period, the sample mean actually appears to be a more reliable choice in terms of expected error. This observation illustrates that choosing between estimators is not always straightforward, as in this case the optimal choice between $\bar{p}_n$ and $\tilde{p}_n$ depends on the number of samples that are available. Importantly, the maximum likelihood estimator does not have this drawback: its confidence bounds never appear wider than the sample mean while still converging quicker.

Across both settings, we have thus observed no noticeable bias problems for any estimator. Importantly, the sample mean has serious convergence issues under the strong bandwagon, where the other estimators are able to converge much sooner. Nonetheless, despite its high efficiency, the affine estimator was also found to initially have the highest variance. It appears that when λ values are known, the maximum likelihood estimator is the most reliable in spite of the lack of proven theoretical guarantees for this estimator. However, even the maximum likelihood estimator is unable to fully undo the slowdown introduced by either level of the bandwagon effect, highlighting the substantial impact this effect still has on effective sample size and relevance estimation.

2.6.3 Bandwagon misestimation

To investigate the effect of misestimated $\hat{\lambda}$ values, we look at the bottom row in Figure 2.4. In particular, it shows what happens when we underestimate a strong bandwagon effect. Clearly, we see that no noticeable bias has been introduced to any estimator. Interestingly, the affine estimators actually suffer less from initial variance: however, we speculate that overestimating the bandwagon effect could instead enlarge the variance. Nevertheless, all

our estimators converge substantially slower than when λ was known (cf. middle row), with their confidence bounds now only moderately smaller than those of $\bar{p}_n$. We thus conclude that the ability of our estimators to accelerate convergence is very sensitive to correct estimation of λ , where we do not observe an advantage for any of the estimators.

The results with misestimated $\hat{\lambda}$ values demonstrate that λ estimation is a crucial part of effective bandwagon correction. Researching bandwagon estimation methods is therefore an important task for future work: since without robust estimation, correcting for the bandwagon effect appears to remain an open problem.

2.7 Discussion

So far, we have investigated the bandwagon effect in both theoretical and empirical terms. In contrast with the well-studied selection bias [84, 175, 178, 224], our experimental results indicate that the bandwagon effect mainly poses a problem for convergence without introducing noticeable statistical bias. Yet the effect is generally referred to as a bias by previous work [207, 218, 220, 234]. This raises a higher-level conceptual question: *should the bandwagon effect be interpreted as a bias?* The remainder of this section will contrast three possible perspectives on this question that we deem valuable for the understanding of the field.

On one hand, both our theoretical and empirical results clearly indicate that statistical bias is not the issue. The expected value of both the sample mean and individual ratings is equal to the true relevance, with no noticeable error in the mean value of each estimator in our experiments. Therefore, in statistical terms, it seems that we are purely dealing with convergence and consistency issues and not with bias. This distinction is important as solutions to inconsistency may be very different than those for bias: i.e. our experimental results suggest that methods based on debiasing techniques are not particularly effective at solving convergence issues.

On the other hand, our theoretical analysis also revealed that the bandwagon effect introduces conditional bias: given that there is an error in the sample mean after m ratings, the expected value of any subsequent rating or sample mean will be incorrect. In other words, under the bandwagon effect any errors in the sample mean, which are unavoidable in practice, will subsequently introduce (conditional) bias. A proponent of the first perspective could argue that this view is mistaking convergence issues for bias. Nevertheless, we argue that as long as both parties understand the

difference between *bias* and *conditional bias*, their disagreement is mainly about semantics and not on the underlying nature of the issue.

Alternatively, one could consider the bandwagon effect as actually changing the *true* preferences of users. This view poses that the observed ratings are not a distortion of the true preferences but that the actual preferences of users have changed. One could argue that this is in line with the users' experience as they do not make a conscious choice to alter their ratings in response to others' feedback but aim to assign ratings according to their satisfaction. Instead, their enjoyment of a product may still be subconsciously influenced by the enjoyment of others. Thus the advantage of this view is the apparent alignment with the users' own experience. However, their experience is precisely unaware of any subconscious factors such as the bandwagon effect. The danger is that by not viewing the bandwagon effect as a misrepresentation of preferences, one may be less inclined to correct for it. However, when training a recommender system, it surely seems critical to know whether high ratings were given because a user was affected by previous users, or whether they made this decision solely based on the item itself.

Apart from these conceptual perspectives, we strongly argue that in statistical terms the bandwagon effect is a problem of convergence and consistency and not of statistical bias. Nevertheless, outside of pure statistical terms, we think that each of the three views brings a valuable perspective to the field. Most importantly, we think the field should be aware of the difference between the bandwagon effect and statistical bias. As long as this distinction is understood, we think it is acceptable if one chooses to conceptually categorize the bandwagon effect as another form of bias, although we would advise against it. Overall, our work aims to promote a more complete statistical understanding of the issues surrounding user interactions that goes beyond bias to include variance, consistency and convergence.

2.8 Conclusion

In this paper we examined the bandwagon effect: an underexplored phenomenon where users' interactions are affected by interactions of earlier users. While previous work has approached the bandwagon effect as problem of bias [218, 220], our theoretical analysis revealed that it does not add bias to the rating distribution nor to the sample mean. Instead, we have shown that it is better viewed as a problem of convergence and consistency. We derived a necessary condition for consistency under the effect, thereby also adding to earlier consistency claims [218]. We then proposed several novel

estimators to mitigate the convergence issues. While our empirical analysis revealed that our novel estimators can substantially increase the convergence rate under a strong bandwagon effect, all estimators also carry their own limitations. Moreover, even when using the empirically best-performing estimator, our purposely simple model of the bandwagon effect still significantly increases the number of samples required for accurate relevance estimation.

Overall, our results thus suggest the following answer to our thesis research question **RQ1**: the bandwagon effect is best viewed as a problem consistency and data-efficiency, not statistical bias.

Future work may wish to examine the impact of more complex bandwagon models (e.g., [229, 234]) as well as parameter and rating scale choices on bias and convergence, both in unpersonalized as well as *personalized* settings. For the latter case in particular, user preference may be further associated with the time of arrival (e.g., item’s fans are both more likely to like it and to rate it early) or λ values for one timestep may vary across users. Nevertheless, even under our simple model, our findings lead us to conclude that there is no clear and well-understood solution for the bandwagon effect.

2.A Additional Proofs

Theorem 2.A.1 (Bias of sample mean and single rating). $\bar{p}_n$ and r_n (Equation 2.4) are unbiased estimators of parameter p for all n :

$$\forall n : \mathbb{E}[\bar{p}_n] = \mathbb{E}[r_n] = p. \quad (2.19)$$

Proof. $\mathbb{E}[\bar{p}_1] = \mathbb{E}[r_1] = p$ by Equations 2.4-2.5. Then by induction:

$$\mathbb{E}[\bar{p}_n] = \frac{(n-1)\mathbb{E}[\bar{p}_{n-1}] + \mathbb{E}[r_n]}{n} = \frac{\lambda_n p + (n - \lambda_n)\mathbb{E}[\bar{p}_{n-1}]}{n} = p.$$

r_n is unbiased as it is a weighted sum of p and unbiased $\mathbb{E}[\bar{p}_{n-1}]$:

$$\mathbb{E}[r_n] = \mathbb{E}[\lambda_n p + (1 - \lambda_n) \bar{p}_{n-1}] = \lambda_n p + (1 - \lambda_n) \mathbb{E}[\bar{p}_{n-1}] = p. \quad \square$$

Lemma 2.A.2 (Variance of r_i). Under the bandwagon effect the variance of any rating r_i is $p(1 - p)$.

Proof. By Theorem 2.A.1:

$$\forall i : \mathbb{V}[r_i] = \mathbb{E}[r_i^2] - \mathbb{E}^2[r_i] = \mathbb{E}[r_i] - \mathbb{E}^2[r_i] = p - p^2. \quad \square$$

Lemma 2.A.3. Covariance of $\bar{p}_{n-1}$ and r_n is $(1 - \lambda_n)^2 \mathbb{V}[\bar{p}_{n-1}]$.

Proof.

$$\begin{aligned} \text{Cov}[\bar{p}_{n-1}, r_n] &= \mathbb{E}[(\bar{p}_{n-1} - p)(\lambda_n p + (1 - \lambda_n) \bar{p}_{n-1} - p)] \\ &= (1 - \lambda_n)^2 \mathbb{V}[\bar{p}_{n-1}]. \end{aligned} \quad (2.20) \quad \square$$

Theorem 2.A.4 (Efficiency of sample mean). The efficiency of the sample mean under the bandwagon effect defined in Equation 2.4 is

$$\mathbb{E}[(\bar{p}_n - p)^2] = p(1 - p) \left(\frac{1}{n^2} + \sum_{i=1}^{n-1} \frac{1}{i^2} \prod_{j=i+1}^n \frac{(j-1)(j+1-2\lambda_j)}{j^2} \right).$$

Proof. By Theorems 2.A.1 and Lemmas 2.A.2, 2.A.3:

$$\begin{aligned} \mathbb{E}[(\bar{p}_n - p)^2] &= \frac{(n-1)^2}{n^2} \mathbb{V}[\bar{p}_{n-1}] + \frac{2(n-1)}{n^2} \text{Cov}[r_n, \bar{p}_{n-1}] \\ &+ \frac{\mathbb{V}[r_n]}{n^2} = \frac{(n-1)(n+1-2\lambda_n)}{n^2} \mathbb{V}[\bar{p}_{n-1}] + \frac{p(1-p)}{n^2}. \end{aligned} \quad (2.21)$$

The proof is completed by unrolling the process to $\mathbb{V}[\bar{p}_1] = p(1 - p)$ and collecting all the terms. $\square$

Theorem 2.A.5. The asymptotic efficiency of the sample mean is:

$$\lim_{n \rightarrow \infty} \mathbb{E}[(\bar{p}_n - p)^2] = \lim_{n \rightarrow \infty} p(1-p) \left(\sum_{i=1}^{n-1} \frac{1}{i(i+1)} \prod_{j=i+1}^{n-1} \left(1 - \frac{2\lambda_j}{j+1} \right) \right).$$

Proof. By re-grouping numerators and denominators in the product of Theorem 2.A.4 and cancelling out extra terms:

$$\mathbb{E}[(\bar{p}_n - p)^2] = p(1-p) \left(\frac{1}{n^2} + \sum_{i=1}^{n-1} \frac{n+1-2\lambda_n}{i(i+1)n} \prod_{j=i+1}^{n-1} \left(1 - \frac{2\lambda_j}{j+1} \right) \right).$$

The proof is completed by taking $\lim_{n \rightarrow \infty}$ of above expression. $\square$

Theorem 2.A.6 (Sample mean consistency). $\sum \lambda_i^2$ diverging is a necessary condition for the consistency of $\bar{p}_n$ (Equation 2.4):

$$\forall \epsilon > 0 : \lim_{n \rightarrow \infty} \Pr(|\bar{p}_n - p| > \epsilon) = 0 \longrightarrow \lim_{n \rightarrow \infty} \sum_{i=1}^n \lambda_i^2 = \infty. \quad (2.22)$$

Proof. By Loève [124, p. 159], consistency of $\bar{p}$ implies its variance vanishing: $\forall \epsilon > 0 : \Pr(|\bar{p}_n - p| \geq \epsilon) \geq \mathbb{V}[\bar{p}_n] - \epsilon^2$. Then: $\forall \epsilon > 0 : \lim_{n \rightarrow \infty} \Pr(|\bar{p}_n - p| > \epsilon) = 0 \rightarrow \lim_{n \rightarrow \infty} \text{Var}(\bar{p}_n) = 0 \rightarrow \forall i : \lim_{n \rightarrow \infty} \prod_{j=i+1}^{n-1} \left(1 - \frac{2\lambda_j}{j+1} \right) = 0 \rightarrow \forall i : \lim_{n \rightarrow \infty} \sum_{j=i+1}^{n-1} \frac{\lambda_j}{j+1} = \infty \rightarrow \forall i : \lim_{n \rightarrow \infty} \sqrt{\sum_{j=i+1}^{n-1} \lambda_j^2} \sqrt{\sum_{j=i+1}^{n-1} \frac{1}{(j+1)^2}} = \infty \rightarrow \lim_{n \rightarrow \infty} \sum_{i=1}^n \lambda_i^2 = \infty$. The second step is due to partial products in Theorem 2.A.5, the third by Leonard [116, p. 6], the fourth by Cauchy-Schwarz inequality and the fifth by convergence of p-series with $p = 2$. $\square$

Proposition 2.A.7 (Conditional bias of sample mean). $\mathbb{E}[\bar{p}_n - p \mid \bar{p}_m] = (\bar{p}_m - p) \prod_{i=m+1}^n \left(1 - \frac{\lambda_i}{i} \right)$ where $m < n$.

Proof. By repeatedly applying the induction step in Theorem 2.A.1, replacing $\mathbb{E}[\bar{p}_n]$ with $\mathbb{E}[\bar{p}_n \mid \bar{p}_m]$ and noting that $\mathbb{E}[\bar{p}_m \mid \bar{p}_m] = \bar{p}_m$:

$$\mathbb{E}[\bar{p}_n - p \mid \bar{p}_m] = \mathbb{E}[\bar{p}_{n-1} - p \mid \bar{p}_m] \left(1 - \frac{\lambda_n}{n} \right) = (\bar{p}_m - p) \prod_{i=m+1}^n \left(1 - \frac{\lambda_i}{i} \right). \quad \square$$

Proposition 2.A.8 (Error bound of $\bar{p}_n$). If $\lambda_i = c^{i-1}$ s.t. $c \in (0, 1)$, the asymptotic expected absolute error of the sample mean is at least $2p(1-p) \exp(-c\phi(c, 1, 2) - c^2\phi(c^2, 2, 2))$, where $\phi(z, s, a) \equiv \sum_{k=0}^{\infty} \frac{z^k}{(a+k)^s}$ is Lerch transcendent and converges to a fixed value $\forall z < 1$ [57].

Proof.

$$\begin{aligned}
\lim_{n \rightarrow \infty} \mathbb{E}[|\bar{p}_n - p|] &\geq \lim_{n \rightarrow \infty} \mathbb{E}[|\mathbb{E}[\bar{p}_n - p | r_1]|] = 2p(1-p) \\
&\cdot \exp\left(\sum_{i=2}^{\infty} \ln\left(1 - \frac{\lambda_i}{i}\right)\right) \geq 2p(1-p) \exp\left(\sum_{i=2}^{\infty} \left(-\frac{\lambda_i}{i} - \frac{\lambda_i^2}{i^2}\right)\right) \\
&= 2p(1-p) \exp\left(-c \sum_{i=0}^{\infty} \frac{c^i}{(2+i)} - c^2 \sum_{i=0}^{\infty} \frac{c^{2i}}{(2+i)^2}\right).
\end{aligned}$$

The first step is due to the law of total expectation and Jensen's inequality for conditional expectation. The second is by Proposition 2.A.7 with $m = 1$. The third is due to a known inequality $\forall x > -0.68 : \ln(1+x) \geq x - x^2$ [113]. $\square$

Theorem 2.A.9. Each $\hat{r}_i$ (and thus also $\hat{p}_n$) is unbiased for all $\hat{\lambda}_i$.

Proof. By Theorem 2.A.1, for any choice of i and $\hat{\lambda}_i$:

$$\mathbb{E}[\hat{r}_i | \hat{\lambda}_i] = \mathbb{E}\left[\frac{r_i - (1 - \hat{\lambda}_i)\bar{p}_{i-1}}{\hat{\lambda}_i}\right] = \frac{\mathbb{E}[r_i - \bar{p}_{i-1}]}{\hat{\lambda}_i} + \mathbb{E}[\bar{p}_{i-1}] = p. \quad \square$$

Theorem 2.A.10 (Conditional bias of $\hat{r}_i | \hat{\lambda}_i = \lambda_i$). If $\hat{\lambda}_i = \lambda_i$, the affine mean estimator $\hat{r}_i$ is an unbiased estimator of p when conditioned on $\hat{r}_j$ or $\bar{p}_j$ s.t. $\forall j < i$.

Proof. For any function of previous ratings $f_i = f(r_1, \dots, r_{i-1})$:

$$\begin{aligned}
\mathbb{E}[\hat{r}_i | f_i, \hat{\lambda}_i] &= \mathbb{E}\left[\frac{\lambda_i p + (1 - \lambda_i)\bar{p}_{i-1} - (1 - \hat{\lambda}_i)\bar{p}_{i-1}}{\hat{\lambda}_i} | f_i\right] \\
&= \frac{\lambda_i}{\hat{\lambda}_i} p + \left(\frac{\hat{\lambda}_i - \lambda_i}{\hat{\lambda}_i}\right) \mathbb{E}[\bar{p}_{i-1} | f_i] = p. \tag{2.23}
\end{aligned}$$

The proof is completed by setting $f_i = \hat{r}_j$ or $f_i = \bar{p}_j$. $\square$

Corollary 2.A.11 (Conditional bias of $\hat{r}_i | \hat{\lambda}_i \neq \lambda_i$). Affine estimator $\hat{r}_i$ is not unbiased when conditioned on a function of previous interactions as the last equality of Equation 2.23 may not hold.

Lemma 2.A.12 (Covariance of affine estimators). $\hat{r}_i$ and $\hat{r}_{j < i}$ are uncorrelated as long as $\hat{\lambda}_i = \lambda_i$.

Proof. By Theorem 2.A.10 and the law of total expectation:

$$\begin{aligned}
\mathbb{E}[\hat{r}_i \hat{r}_j] &= \mathbb{E}[\mathbb{E}[\hat{r}_i \hat{r}_j | \hat{r}_j]] = \mathbb{E}[\hat{r}_j \mathbb{E}[\hat{r}_i | \hat{r}_j]] \\
&= \mathbb{E}[\hat{r}_j \mathbb{E}[\hat{r}_i]] = \mathbb{E}[\hat{r}_j] \mathbb{E}[\hat{r}_i]. \quad \square
\end{aligned}$$

Theorem 2.A.13. The variance of affine estimator in Equation 2.15 is:

$$\mathbb{V}[\hat{p}_n] = \frac{1}{(\sum_{i=1}^n \omega_i)^2} \sum_{i=1}^n \frac{\omega_i^2}{\lambda_i^2} \left(p(1-p) - (1-\lambda_i)^2 \mathbb{V}[\bar{p}_{i-1}] \right).$$

Proof. By Lemmas 2.A.12, 2.A.3 and 2.A.2:

$$\begin{aligned} \mathbb{V}[\hat{p}_n] &= \frac{1}{(\sum_{i=1}^n \omega_i)^2} \left(\sum_{i=1}^n \omega_i^2 \mathbb{V}[\hat{r}_i] + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n \omega_i \omega_j \text{Cov}[\hat{r}_i, \hat{r}_j] \right) \\ &= \frac{1}{(\sum_{i=1}^n \omega_i)^2} \sum_{i=1}^n \frac{\omega_i^2}{\lambda_i^2} (\mathbb{V}[r_i] + (1-\lambda_i)^2 \mathbb{V}[\bar{p}_{i-1}] - 2(1-\lambda_i) \\ &\quad \cdot \text{Cov}[r_i, \bar{p}_{i-1}]) = \left(\sum_{i=1}^n \omega_i \right)^{-2} \sum_{i=1}^n \frac{\omega_i^2}{\lambda_i^2} \left(p - p^2 - (1-\lambda_i)^2 \mathbb{V}[\bar{p}_{i-1}] \right). \quad \square \end{aligned}$$

Theorem 2.A.14 (Convergence of affine estimator). When $\forall \hat{\lambda}_i = \lambda_i$, $\hat{p}_n = \frac{\sum_{i=1}^n \omega_i \hat{r}_i}{\sum_{i=1}^n \omega_i}$ converges within ϵ of the true value p with probability of at least $1 - \alpha$ if $(\sum_{i=1}^n \omega_i)^2 \geq \frac{1}{2\epsilon^2} \log \frac{2}{\alpha} \sum_{i=1}^n \left(\frac{\omega_i}{\lambda_i} \right)^2$.

Proof. Define X_n as $\sum_{i=1}^n \omega_i (\hat{r}_i - p)$, $X_0 = 0$. Then X_n is a martingale as $\mathbb{E}[X_n | X_0, X_1, \dots, X_{n-1}] = X_{n-1} + \mathbb{E}[\omega_n (\hat{r}_n - p) | \bar{p}_{n-1}] = X_{n-1}$. For each n : $X_n - X_{n-1} = \omega_n (\hat{r}_n - p)$, whose values for $r_n = 0$ and $r_n = 1$ are separated by $d_n = \frac{\omega_n}{\lambda_n}$. By Azuma's inequality:

$$\begin{aligned} \Pr \left(\left| \frac{\sum_{i=1}^n \omega_i (\hat{r}_i - p)}{\sum_{i=1}^n \omega_i} \right| > \epsilon \right) &= \Pr \left(|X_n - X_0| > \epsilon \sum_{i=1}^n \omega_i \right) \\ &\leq 2 \exp \left(\frac{-2\epsilon^2 (\sum_{i=1}^n \omega_i)^2}{\sum_{i=1}^n d_i^2} \right). \end{aligned} \quad (2.24)$$

The proof is completed by taking the logarithm of both sides and redistributing the terms. $\square$

Corollary 2.A.15 (Convergence of $\bar{p}_n$ for specific ω_i). By Theorem 2.A.14, when $\forall i : \hat{\lambda}_i = \lambda_i$, $\omega_i = 1$, $\hat{p}_n$ is a consistent estimator of p if $\sum_{i=1}^n \frac{1}{\lambda_i^2}$ grows slower than n^2 . Similarly, when $\forall \hat{\lambda}_i = \lambda_i = \omega_i$, $\hat{p}_n$ is a consistent estimator of p if $(\sum_{i=1}^n \lambda_i)^2$ grows quicker than n .

Chapter 3

Sample-Free Almost-Exact Estimation of Plackett-Luce Propensities for Off-Policy Ranking

Off-policy evaluation (OPE) and optimization for learning to rank (LTR) leverage document placement probabilities to correct for the effects of various statistical biases, e.g., position bias. However, computing these propensities poses a challenge, as for most ranking models this requires iterating over all possible rankings. A common solution is to approximate them by sampling multiple rankings and using the observed document frequencies per position. Nevertheless, even when using extremely large numbers of sampled rankings, these estimates often still contain significant estimation errors.

Plackett-Luce (PL) ranking model has recently seen wide adoption owing to its simplicity and the development of computationally efficient optimization methods. Nevertheless, there is currently no accurate and sample-efficient method for the estimation of PL propensities, limiting its use in OPE. This observation prompts our second thesis research question:

RQ2 *Can we accurately and efficiently estimate document placement probabilities for the Plackett-Luce ranking model without sampling?*

In this work, we propose the novel *Marginalized Plackett-Luce* (MPL) method to efficiently and accurately calculate document-rank placement probabilities under the PL ranking model. In particular, we establish MPL by first showing that this probability is the expected value of a Poisson binomial distribution over the document scores; subsequently, we leverage a known

connection between the Poisson binomial distribution, convolutional operations and numerical integration, to achieve efficient and accurate propensity estimation. Furthermore, we argue and later empirically demonstrate that MPL provides near-exact estimation when computing the function over a practical number of evaluation points.

Current propensity estimation methods may in practice, however, already provide sufficiently accurate propensity estimates for OPE, particularly when used along with commonly used variance reduction techniques like propensity clipping that reduce estimator variance at the cost of introducing bias. This leads to our third thesis research question:

RQ3 *Do our new propensity estimates allow us to reduce bias and improve data-efficiency in off-policy learning to rank?*

To that end, our experiments also confirm that the propensity estimation of MPL also leads to substantial improvements over the sampling-based method in OPE. Taken together, our results suggest that MPL’s accuracy, efficiency and applicability to downstream tasks may thus open the door to a wider use of PL policies in off-policy learning to rank.

3.1 Introduction

Modern learning to rank (LTR) approaches often rely on historical user interaction data such as clicks, due to their relative abundance compared to other types of interactions or labels [87, 89, 90, 142]. However, user click data is also known to be affected by various types of statistical biases; most prominently by position bias [145, 210] which occurs because documents that were ranked higher by the production (logging) policy are more likely to be examined and thus interacted with [145]. As such, off-policy learning and evaluation methods rely on the information about the logging policy to correct for such forms of bias [28, 81, 145].

Importantly, in most cases, including under the widely used position-based model (PBM) and item-position model (IPM) [27, 28], these methods require the values of the placement probabilities of each document across every position. Yet, for most ranking policies, the only known method for the exact calculation of placement probabilities iterates over all possible rankings and tracks the frequency of each document at each position. However, due to the enormous number of possible rankings, this approach incurs a heavy computational cost that generally makes it infeasible in practice, even for low document numbers [143, 144].

To our knowledge, the only commonly used alternative is to estimate these probabilities by sampling a high number of rankings and counting the occurrences of every document at each position [69, 144]. However, due to variance, many sampled rankings are required for precise estimates, and thus can still lead to computational infeasibility for high precision – especially when these estimates cannot be pre-computed or need frequent re-computation [25, 52, 143].

To address this issue, we propose Marginalized Plackett-Luce (MPL), a new efficient way to calculate these placement probabilities for the popular Plackett-Luce (PL) ranking model [26, 52, 143, 144, 154]. We show that under this model, due to its connection to the Gumbel distribution, the (marginal) placement probabilities can also be seen as the expectation of a Poisson binomial random variable over the document scores. By leveraging this interpretation and the known connection of Poisson binomial to convolutions [16, 56] along with numerical integration, we propose a novel approximation approach for estimating the probability of a document being placed at a ranking position. Importantly, we argue that despite being an approximation, our approach is effectively exact when evaluating it over a practically feasible number of points. Our experiments also confirm the significant gains in accuracy and efficiency of MPL compared to the standard sampling approach. Furthermore, we show that MPL’s more accurate predictions can be leveraged for downstream use, such as for improved off-policy evaluation (OPE). Altogether, our method opens the door for a new direction of more exact propensity estimation; and thereby, we hope that it motivates for a wider use of PL policies for off-policy ranking evaluation and optimization applications.

3.2 Related work

Estimating document-placement probabilities (a.k.a. propensities) is a common problem in ranking and recommendation. For instance, in ranking fairness it is often used to model document exposure [143, 183, 184]. In off-policy learning (OPL) and estimation, propensities are also used to correct for the interaction biases introduced by the logging policy, e.g., through the application of inverse propensity scoring (IPS) using PBM [90] or IPM [118]. Whilst Monte-Carlo (MC) sampling may be used to estimate these document placement probabilities for arbitrary policies [69], Swaminathan, Joachims and others [90, 196] have shown that even small estimation errors, particularly for low propensities, can have a significant impact on OPE and OPL. Further works consider the impact of clipping [10] and

self-normalization [196] on off-policy estimation: both are found to reduce estimator variance but at the cost of introducing bias. Taken together, the above results suggest that propensity estimation may be an important source of error.

In this work, we focus on PL ranking models [154]; which are popular due to their conceptual simplicity, as well as the recent development of highly efficient PL gradient estimation methods [26, 143, 144]. Kool et al. [108] along with Ma et al. [128] also provide closed-form expressions for computing the probability of partial document orderings under the PL model. In spite of these advancements, MC sampling remains the only existing method to estimate PL placement probabilities of individual documents [143, 184].

3.3 Background: Propensity Estimation with PL Rankings

Plackett-Luce Ranking Models.

One of the main tasks in LTR is to produce a ranking $y = [y_1, \dots, y_K]$ of K documents from candidate set D for a given query or context [122]. Under the PL ranking model, previously unplaced documents are iteratively added to the ranking according to their softmax probabilities, equivalent to sampling K documents from softmax without replacement [107]. In particular, for documents with (logit) scores m_j and where $\mathbb{1}[j \notin y_{1:i-1}]$ denotes j not being placed in a partial ranking spanning positions 1 to $i-1$, the probability of full ranking y is:

$$\pi_{\text{PL}}(y) = \prod_{i=1}^K \pi_{\text{PL}}(y_i \mid y_{1:i-1}) = \prod_{i=1}^K \frac{e^{m_{y_i}} \mathbb{1}[y_i \notin y_{1:i-1}]}{\sum_{d'=1}^{|D|} e^{m_{d'}} \mathbb{1}[d' \notin y_{1:i-1}]}. \quad (3.1)$$

Notably, previous works on the optimization of various PL objectives often leverage the connection of the PL distribution to the Gumbel distribution [25, 52, 143]. In particular, a $\text{Gumbel}(m, \beta)$ random variable with location m and spread β is defined between $[-\infty, \infty]$ with the probability density function f and cumulative distribution function F of $\text{Gumbel}(m_j, 1)$ respectively being:

$$f_j(x) = f(x; m_j, 1) = e^{m_j - x - e^{m_j - x}}, \quad F_j(x) = F(x; m_j, 1) = e^{-e^{m_j - x}}. \quad (3.2)$$

The well-known Gumbel-max trick [78, 107], used for efficient sampling from softmax and Plackett-Luce distributions, leverages the fact that these

distributions can also be represented in terms of $|D|$ variables $\text{Gumbel}(m_j, 1)$, one for every document $j \in D$ and centered on their scores m_j . The softmax sampling probability $\Pr(d) = \Pr(y_1 = d)$ of document d is then equal to the probability of taking a sample from each $G_j \sim \text{Gumbel}(m_j, 1)$ and G_d , associated with document d , then being the highest [78]. Following similar intuition, Kool et al. [108] show that under the PL model, the probability $\Pr(Y \prec Y^C)$ that all documents from set Y are placed before all documents from a disjoint complementary set Y^C s.t. $Y \cup Y^C = D$ is:

$$\Pr(Y \prec Y^C) = \int_{-\infty}^{\infty} \prod_{i \in Y} (1 - F_i(x)) f(x; \log(\sum_{j \in Y^C} e^{m_j}), 1) dx. \quad (3.3)$$

Intuitively, $f(x; \log(\sum_{j \in Y^C} e^{m_j}), 1)$ represents the distribution of the highest Gumbel sample $G_j \in Y^C$ (and is itself a Gumbel distribution), whilst $\prod_{i \in Y} (1 - F_i(x))$ represents the probability that *all* $G_i \in Y$ exceed it. Finally, Ma et al. [128] also propose an extension of the above approach to multiple sets.

Propensity Estimation

Importantly, under the PL model, the probability of placing document d in position k $\Pr(d = y_k)$, i.e., d 's propensity for k , depends on the previously placed documents – obtaining it naively thus requires enumerating over all possible rankings $y_{1:k-1}$. In practice, the MC estimator can be used to approximate the propensity by sampling M rankings y^i from π_{PL} [69, 143]:

$$\Pr(d = y_k) = \sum_{y_{1:k-1}} \pi(y_{1:k-1}) \pi(d \mid y_{1:k-1}) \quad (3.4)$$

$$\approx \frac{1}{M} \sum_{i=1}^M \mathbb{1}[y_k^i = d] = \Pr_{\text{MC}}(d = y_k). \quad (3.5)$$

These propensity estimates can then be used by OPE estimators to correct the observed clicks for position bias. For example, IPS [90] reweights the logged clicks using $\frac{1}{\Pr(d=y_k)}$ under the IPM [118] and $\frac{1}{\sum_{i=1}^K \lambda_i \Pr(d=y_i)}$ under the PBM [145] click models, where λ_i are position-bias weights. Clipping $h(p) = \max(p, z)$ with hyperparameter z can be further applied to the denominators of the above weights to reduce estimator variance [10, 125, 190]. Nevertheless, poor propensity estimates are known to lead to significant OPE errors [10]. To address this issue, we will leverage the Gumbel-based intuition to derive our novel highly accurate estimator of $\Pr(d = y_k)$.

3.4 Method: Placement Propensity Estimation

This section proposes *Marginalized Plackett-Luce* (MPL) for accurately and efficiently estimating document-position probabilities of PL ranking models.

Derivation: Plackett-Luce Order Statistic.

We start by extending Equation 3.3 to three mutually exclusive sets: $Y_{k \setminus d}$ of size k that does not include d ; its complement $Y_{k \setminus d}^C$ that also excludes d ; and $\{d\}$. Thus, the probability $\Pr(Y_{k \setminus d} \prec d \prec Y_{k \setminus d}^C)$ that d is placed after all documents in $Y_{k \setminus d}$ and before all documents in $Y_{k \setminus d}^C$, is equal to the probability that all $G_i \in Y_{k \setminus d}$ are higher than G_d and that all $G_j \in Y_{k \setminus d}^C$ are lower than G_d :

$$\Pr(Y_{k \setminus d} \prec d \prec Y_{k \setminus d}^C) = \int_{-\infty}^{\infty} f_d(x) \prod_{i \in Y_{k \setminus d}} (1 - F_i(x)) \prod_{j \in Y_{k \setminus d}^C} F_j(x) dx. \quad (3.6)$$

This result is equivalent to that of Ma et al. [128], yet we retain the explicit dependence on densities $F(x)$, which we later leverage for our method.

A crucial observation is that $\Pr(Y_{k \setminus d} \prec d \prec Y_{k \setminus d}^C)$ represents the probability of a specific partially sorted ranking $Y_{k \setminus d} \prec d \prec Y_{k \setminus d}^C$ where d is in position $k + 1$. Therefore, the probability of observing any partial ranking with d in position $k + 1$ is simply the sum over all possible $Y_{k \setminus d}$. Importantly, that is exactly the probability of d being placed in position $k + 1$ under the PL distribution, i.e., its $|D| - k$ 'th order statistic:

$$\Pr(y_{k+1} = d) = \sum_{Y_{k \setminus d}} \Pr(Y_{k \setminus d} \prec d \prec Y_{k \setminus d}^C) \quad (3.7)$$

$$= \int_{-\infty}^{\infty} f_d(x) \sum_{Y_{k \setminus d}} \prod_{i \in Y_{k \setminus d}} (1 - F_i(x)) \prod_{j \in Y_{k \setminus d}^C} F_j(x) dx. \quad (3.8)$$

Importantly, in contrast to the closed-form solution in Section 3.3, $Y_{k \setminus d}$ refers to a set of documents rather than a specific ranking, and thus the relative order of its elements does not matter. In practice, this means that we do not need to consider all document permutations and only their combinations.

Plackett-Luce as Poisson Binomial Expectation.

Nevertheless, explicitly iterating over all possible sets of size $k < K$ still seems prohibitively expensive for most common ranking sizes K . To avoid this issue, we note that for a given value of x we can treat each $G_j > G_d$ as a success in a Bernoulli trial with its own probability $q_j(x) = 1 - F_j(x)$ and where all trials are independent (conditional on x). The summation in Equation 3.7 then reflects the probability of the observed number of successes S in $|D| - 1$ such trials being k :

$$\Pr(S = k \mid d, x) = \sum_{Y_{k \setminus d}} \prod_{i \in Y_{k \setminus d}} q_i(x) \prod_{j \in Y_{k \setminus d}^C} (1 - q_j(x)). \quad (3.9)$$

S is thus a Poisson binomial variable whose probabilities q may be different for each x . Combining Equations 3.8 and 3.9, integrating over all $x = G_d$ then yields:

$$\Pr(y_{k+1} = d) = \int_{-\infty}^{\infty} f_d(x) \sum_{Y_{k \setminus d}} \prod_{i \in Y_{k \setminus d}} q_i(x) \prod_{j \in Y_{k \setminus d}^C} (1 - q_j(x)) dx \quad (3.10)$$

$$= \int_{-\infty}^{\infty} f_d(x) \Pr(S = k \mid d, x) dx \quad (3.11)$$

$$= \mathbb{E}_{G_d} [\Pr(S = k \mid d, G_d)]. \quad (3.12)$$

As such, $\Pr(y_{k+1} = d)$ is the expected value of the Poisson binomial probability of k documents scoring above d . This interpretation is significant, as for a given value of x the Poisson binomial distribution can be efficiently obtained without explicitly iterating over all possible sets.

In particular, as shown by Fernandez et al. [56], a Poisson binomial distribution with trial success probabilities $q_1, \dots, q_{|D|}$ can be seen as a convolution $c^{1:|D|} = \ast_{i=1}^{|D|} [1 - q_i, q_i]$ over probability vectors $\forall [1 - q_i, q_i]$. Accordingly, we use the *direct convolution* (DC) algorithm of Biscarri et al. [16], who showed that the convolution of the first $k - 1$ documents $c^{1:k-1}$ with $[1 - q_k, q_k]$ is $[(1 - q_k) \cdot c_1^{1:k-1}] \oplus ((1 - q_k) \cdot c_{2:k}^{1:k-1} + q_k \cdot c_{1:k-1}^{1:k-1}) \oplus [q_k \cdot c_k^{1:k-1}]$. Here $\oplus$ denotes the concatenation of two lists and $c_{m:n}^{1:k-1}$ denotes the entries of $c^{1:k-1}$ between indices m and n . Leveraging this recursive relation, noting that at each step we only need to keep up to K first entries of each convolution, we obtain the Poisson binomial probabilities for the first K ranks $c_{1:K}^{1:|D|}$ in $O(|D|K)$ time.

However, an important consideration is that when simultaneously calculating $\Pr(S = k \mid d, x)$ for all $|D|$ documents of a query, each document d has to be excluded from its own convolution, as d cannot appear in either

$Y_{k \setminus d}$ nor $Y_{k \setminus d}^C$. To avoid explicitly computing a size $|D| - 1$ convolution $|D|$ times, we take inspiration from the forward-backward algorithm [13] and propose the following scheme: We start by calculating the convolution in both directions, i.e., $c^{1:|D|}$ and $c^{|D|:1}$, whilst caching all intermediate results, e.g., $c^{1:3}$ and $c^{|D|:5}$. The convolution $c^{1:|D| \setminus d}$ that excludes d is then exactly $c^{1:d-1} * c^{|D|:d+1}$. The entry associated with k successes can then be obtained by summing over the entries of the $k + 1$ 'th anti-diagonal of $(c^{1:d-1} \otimes c^{|D|:d+1})$, where $\otimes$ denotes the outer product. We can use the above approach to simultaneously calculate $\Pr(S = k \mid d, x)$ for all $d \in D$ and for all $k < K$, resulting in the computational complexity of $O(|D|K^2)$. We note that it may also be possible to leverage the relation $c_k^{1:|D| \setminus d} = (c_k^{1:|D|} - q_d \cdot c_{k-1}^{1:|D| \setminus d}) / (1 - q_d)$ to further improve the computational complexity of our approach to $O(|D|K)$. However, our approach avoids division which leads to high numerical stability that could be preferable across most K .

Numerical Integration.

With the ability to efficiently calculate $\Pr(S = k \mid d, x)$, we then return to the other aspect of Equation 3.11. In particular, $\Pr(d = y_{k+1})$ requires integrating over all possible values x that G_d can take. However, in practice almost all of the probability density of all f_d is concentrated in some $[a, b] = [\min_{m_j} + c_1, \max_{m_j} + c_2]$, where a reasonable choice of c_1 and c_2 (s.t. $c_1 < 0 < c_2$) may be some quantiles l and $1 - l$ of a Gumbel(0, 1) distribution:

$$\Pr(y_{k+1} = d) \approx \int_{a=\min_{m_j}+c_1}^{b=\max_{m_j}+c_2} f_d(x) \Pr(S = k \mid d, x) dx. \quad (3.13)$$

The form $\int_a^b g(x)$ is crucial, as such integrals can be efficiently approximated with Gauss-Legendre quadrature using the rule: $\int_{-1}^1 g(x) dx \approx \sum_{i=1}^N w_i g(x_i)$. The integration points x_i and associated weights w_i are the first N roots of the N 'th Legendre polynomial and can be pre-computed once and shared across the whole dataset [19]. Using this form to represent Equation 3.13 and applying a change of variables, switching the integration limits from $[-1, 1]$ to $[a, b]$ then allows us to obtain our final estimator, which we refer to as MPL:

$$\Pr(y_{k+1} = d) \approx \alpha \sum_{i=1}^N w_i f_d(\alpha x_i + \beta) \Pr(S = k \mid d, \alpha x_i + \beta) \quad (3.14)$$

$$= \Pr_{\text{MPL}}(d = y_{k+1}), \quad (3.15)$$

where $\alpha = (b-a)/2$ and $\beta = (a+b)/2$. Importantly, Gauss-Legendre quadrature is highly accurate and the approximation is exact where the approximated function $g(x)$ is a polynomial of degree $2N - 1$ or lower [201]. Whilst Equation 3.13 itself is not a polynomial; our observations indicate that the region spanning its values that are meaningfully above 0 can usually be well approximated using polynomials of degree < 20 , suggesting that for a sufficiently large N Equation 3.14 is sufficient to closely approximate $P(y_{k+1} = d)$. In practice, we observe that our approximation closely resembles the true value $\Pr(y_{k+1} = d)$ and in our experiments does not significantly change beyond $N = 200$ points used to simultaneously calculate $\forall d, k : \Pr(y_{k+1} = d)$. We note that the optimal N may differ when m_j are spaced far apart, as more points may be needed to correctly approximate a wider interval.

The final computational complexity of our new approach is $O(N|D|K^2)$, a considerable improvement over the $O(D!)$ of the naive approach in Section 3.3. Finally, we note that our estimator is fully differentiable w.r.t. the document scores m_j and can thus be used to optimize objectives such as normalized discounted cumulative gain (NDCG) or exposure without the need for manual derivation of gradient functions [143]. However, the scope of this paper is limited to evaluating the accuracy of the estimated propensities and their downstream use.

3.5 Experimental Setup

The first goal of our experiments is to evaluate whether MPL is successful at the propensity estimation task. In particular, we are interested in the trade-off between MPL’s accuracy and computation time, and how those compare to the sampling-based MC approach, leading to our first (chapter) research question:

RQ3.1 *Does MPL more accurately estimate PL propensities compared to MC at the same computational cost?*

Furthermore, as discussed in Section 3.4, the accuracy of MPL may depend on both the choice of hyperparameters, as well as various aspects of the query, e.g., the distribution of document scores. As such, our second research question is:

RQ3.2 *Does the estimation accuracy of MPL substantially vary under different choices of hyperparameters and under various conditions?*

Lastly, we are also interested in whether the improved propensity estimation can be leveraged for downstream tasks, for example OPE. Our third and final research question is therefore:

RQ3.3 *Do MPL propensity estimates allow for better performance in OPE?*

Our research questions thus address the accuracy, efficiency and applicability of MPL to propensity estimation, along with its practical applications.

We conduct our experiments on *Yahoo! Webscope* [37] and *MSLR-WEB30k* [156] datasets. Yahoo consists of 29,921 queries with an average of 24 candidate documents per query, while MSLR contains 30,000 queries with an average of 125 documents per query, where we remap the original relevance labels $r \in [0, 4]$ to $\rho \in [0, 1]$ as $\rho = (2^r - 1)/(2^4 - 1)$. For the propensity estimation task, we train a PL ranking model with 5 hidden layers of size 128 using the discounted cumulative gain (DCG) loss with PL-Rank-3 gradient estimation with 1,000 sampled rankings per update step [144]. We limit training to one epoch and constrain the document scores to the range $[0, 40]$ to reduce the prevalence of effectively deterministic rankings. Model training and propensity estimation is carried out in PyTorch [149] on a single NVIDIA A100 GPU using single precision.¹

To evaluate the accuracy of the estimated propensities, we compare them with the ground truth values in terms of mean squared error (MSE) and symmetric mean absolute percentage error (sMAPE) on MSLR. For true propensities p and predicted values $\hat{p}$, $\text{sMAPE}(p, \hat{p})$ is defined as $\frac{1}{n} \sum_{i=1}^n |p_i - \hat{p}_i| / (|p_i| + |\hat{p}_i|)$. Whilst sMAPE values close to 0 reflect a small difference in the ratio of $\hat{p}_i$ and p_i , values close to 1 indicate that $\hat{p}_i + p_i \approx \max(\hat{p}_i, p_i)$, and sMAPE of 0.5 represents $\hat{p}_i$ being on average 3 times higher or lower than p_i . Thus, while MSE measures absolute differences, sMAPE reflects the relative (bi-directional) prediction error, which may better reflect the use of propensities as reweighting factors in OPE.

We estimate propensities up to rank $K = 10$ with MPL using $N \in \{50, 100, 200, 500, 1000\}$ integration points with integration limit parameter set to $l = 10^{-5}$, and sample $M \in \{10^3, 10^4, \dots, 10^7\}$ rankings for MC, measuring the average per-query execution time. As the true propensity values are not actually known, we approximate *oracle* values using the sampling-based method with 10^7 samples per query. However, as a consequence, the lowest possible non-zero oracle propensity is then $p = 10^{-7}$, and in preliminary experiments 70% of propensities were left as $p = 0$ due to them not having a single observation. Importantly, probability estimates below 10^{-7}

¹We observed no difference in our results when using double precision.

may then lead to a non-zero MSE even if they are actually correct, whilst estimates $\hat{p} = 0$ may instead get zero MSE where they should not, whilst also potentially leading to zero-division in sMAPE. As such, we restrict our analysis only to the document-position pairs with at least one oracle observation, and set the remaining predicted probabilities $\hat{p} \leftarrow \max(\hat{p}, 10^{-7})$. We group propensities based on their query’s number of documents $|D|$ into equally sized groups, or on their query’s difference Δ in the highest and lowest document scores m_i , allocating 5%, 45% and 50% of the queries to groups with progressively increasing Δ values. Furthermore, we also partition the errors for individual propensities into equal groups based on their true values p , as well as evaluate predictions for different ranking sizes $K \in \{5, 10, 20, 30\}$ and MPL integration limits $l \in \{10^{-1}, 10^{-2}, \dots, 10^{-6}\}$.

For OPE, we further train two separate PL policies with access to non-overlapping sets of features, selected based on the magnitude of each feature’s rank-wise correlation (Kendall’s τ) with the training set labels of each fold. The strong target policy π_{tgt} uses the third of the features with the highest correlation magnitude, whilst π_{log} is a weaker logging policy that uses the next best third of features. We simulate user clicks c by sampling $\{10^2, 10^3, \dots, 10^7\}$ rankings per query from π_{log} – for a document d with relevance ρ_d in position k , the clicks are sampled from Bernoulli distribution with probability $\Pr(c \mid d, k) = \frac{\rho_d}{\log_2(k+1)}$. We then reweigh the clicks with the predicted logging and target policy propensities using the IPS estimator with PBM and IPM click models [28] and evaluate the estimates using MSE. Note that we compute a separate squared error term for each query as to penalize larger individual errors. The clipping parameter $z \in \{10^{-1}, 10^{-2}, \dots, 10^{-9}\}$ is chosen independently for each run to minimize the MSE of the validation set. For MPL we only report the results for $l = 10^{-5}$, $N \in \{50, 1000\}$ due to the high similarity of OPE results for all MPL models.

3.6 Results

3.6.1 Propensity estimation: accuracy and efficiency

Matching our first research question, **RQ3.1**, we begin with investigating whether MPL allows for more accurate propensity estimation compared to MC at the same computational cost. Figure 3.1 shows the relationship between MSE, sMAPE and the average propensity estimation time for a

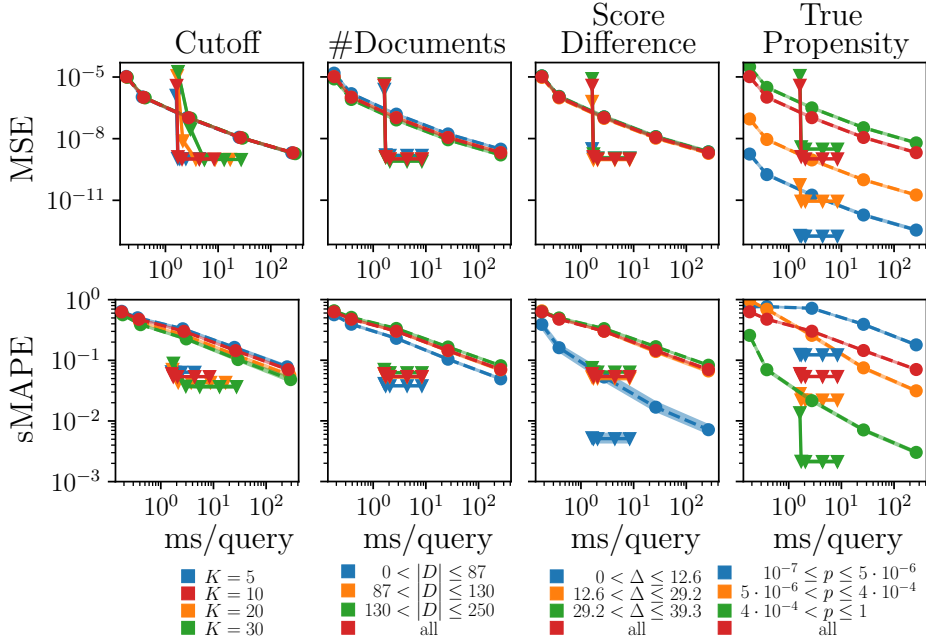


Figure 3.1: MSE and sMAPE of MPL (triangles) and MC (circles) with different per-query compute times in milliseconds. Predictions grouped by ranking size (left), query document number (center left), the difference between the highest and the lowest query document scores (center right) and the true propensity (right). Results are averages over 5 independent folds, the difference between the best and worst fold denoted by shading. Red lines denote performance on all predictions and are the same in all subfigures.

single query across different settings. Our discussion only concerns the left column of Figure 3.1.

We can see that both MPL and the sampling-based MC can be used to estimate document propensities across different ranking sizes (Figure 3.1, left), with both approaches consistently improving in both MSE and sMAPE as the number of samples or points is increased, along with compute time. Nevertheless, for $K = 10$ (red lines) MPL is able to achieve almost two orders of magnitude lower MSE and over a five-fold reduction in sMAPE at ~ 1.5 milliseconds (ms) per query compared to MC at the same computational cost. In fact, our method achieves its optimal performance at just ~ 2 ms per query ($N = 200$) and thus does not require more integration points. Moreover, MPL outperforms the sampling-based approach even when the latter runs for over 250 ms per query, i.e., 125 times longer, highlighting MPL’s high accuracy and sample-efficiency.

Interestingly, when looking at the performance of MPL across different K values, MPL attains a larger MSE in longer rankings when an insufficient number of points $N \leq 100$ is used. Furthermore, whilst the runtime of MC is overall proportional to the ranking size K , the runtime of MPL appears to scale quadratically, consistent with our discussion in Section 3.4. Nevertheless, in all cases MPL recovers its optimal performance at $N = 200$, achieving lower MSE and sMAPE than MC across all ranking sizes at the same or lower computational cost.

Based on our observation that MPL with $N = 200$ consistently outperforms the sampling-based method at the same or higher runtime, we answer **RQ3.1** affirmatively: *MPL provides highly accurate propensity estimates at a lower computational cost compared to MC.*

3.6.2 Propensity estimation: sensitivity

We turn to our second research question (**RQ3.2**) to analyze whether other aspects of queries, individual propensities, or hyperparameter choices affect the performance of MPL. Our discussion for **RQ3.2** concerns all columns except the left column of Figure 3.1; note that ranking size $K = 10$ is used for these results.

The second column of Figure 3.1 shows prediction errors for queries with different numbers of candidate documents. We see that both MPL and MC attain marginally lower MSE yet higher sMAPE on queries with more documents (green). We hypothesize that having more documents under the same scale of logits may likely lead to lower per-document probabilities, where minor absolute prediction errors can cause high relative errors. The above interpretation is also consistent with our observations for query grouping based on the spread of their logits, shown in the third column of Figure 3.1. There we can see that queries with less dispersed document scores (blue) have comparable MSE to that of other queries, whilst also having substantially lower relative errors. Importantly, whilst for MC its sMAPE on such queries is substantially lower than on other predictions, MPL achieves close to optimal performance at just $N = 50$, suggesting particularly high computational efficiency for less saturated policies.

The separation between groups is particularly pronounced when instead splitting individual predictions by their target value, as shown in the last column of Figure 3.1. Both MPL and MC demonstrate substantially lower MSE yet high sMAPE for lower propensity values, and vice-versa for the highest propensities. That this holds for the sample mean MC is not surprising [42, page 53] – however the behavior is less clear for MPL, and may instead arise due to the approximation error of the oracle estimator.

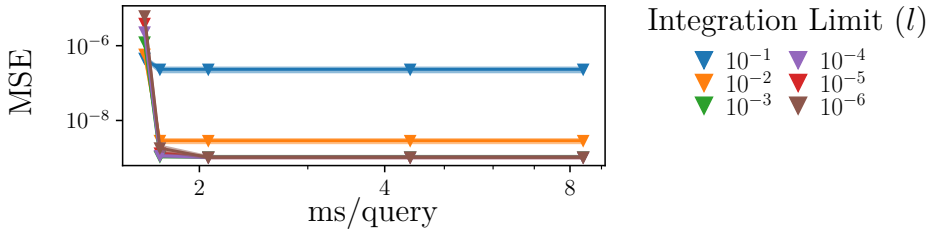


Figure 3.2: MSE of MPL with different integration limits under varied per-query compute time on an A100 GPU. Low l values represent wider integration ranges.

Importantly, however, compared to MC with sMAPE of 0.7 for the lowest propensities (i.e., it has on average a 6-fold prediction error), sMAPE of MPL is still below 0.1 (i.e., p and $\hat{p}$ are within 25% of each other). Whilst this is still higher than its sMAPE of larger propensities, the above results overall suggest high applicability of MPL to inverse propensity estimation due to its low relative error.

Finally, Figure 3.2 shows the performance of MPL across different values of l defining the integration boundaries of Equation 3.13. We see that whilst excessively large l may lead to high prediction error, likely due to the underestimation of the true propensity p , too low l leads to the need for more integration points N . Nevertheless, MPL does not appear to be highly sensitive to the exact value of l , as for most l values MPL is able to recover its high prediction accuracy.

Altogether, we observe some variation in the performance of MPL. To answer **RQ3.2**: *in addition to the choice of N and K on the computational time, MPL is particularly well suited for queries with narrow document score ranges, and its performance strongly depends on the target propensity value.*

3.6.3 Off-policy evaluation

We then arrive at our final research question (**RQ3.3**): *Do MPL propensity estimates allow for better performance in OPE?* The two left columns of Figure 3.3 show the per-query MSE for the IPS estimator under IPM and PBM for a varied number of rankings used for gathering clicks with MPL or MC propensity estimates for policies $\pi_{\log}$ and π_{tgt} .

We can see that relevance estimates with more clicks allow for lower errors under both PBM (left) and IPM (center left). However, whilst MC estimates using more sampled rankings M are able to take advantage of more accurate relevance estimates (more clicks), the improvement of MC quickly tapers off for lower M values, potentially due to the bias introduced by clipping. Figure 3.3 (right, center right) shows the clipping values selected

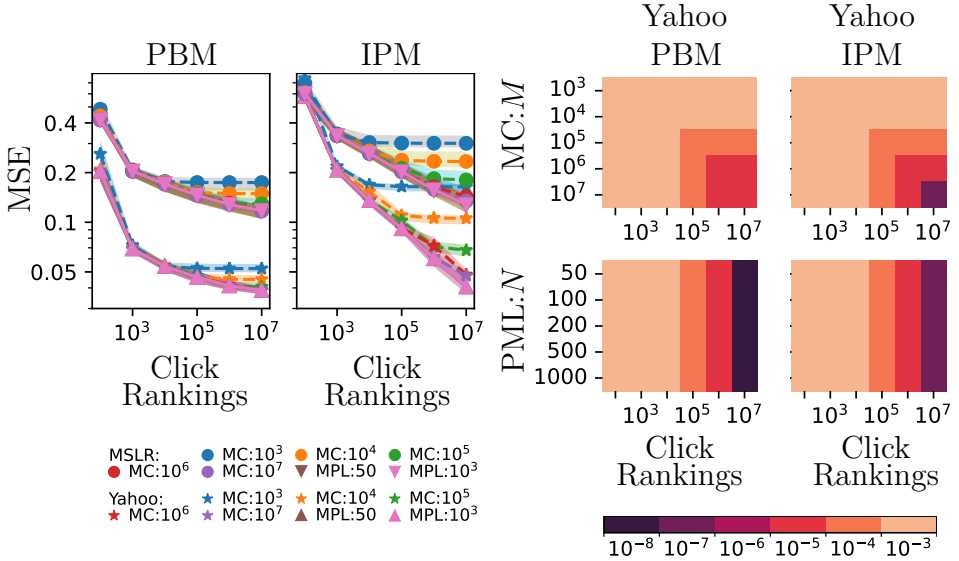


Figure 3.3: Query-level MSE on MSLR and Yahoo (left) and the chosen Yahoo clipping values z (right) with MPL and MC propensities of various numbers of integration points N or sampled rankings M under a varied number of clicks. MSE averages over 5 independent folds, the difference between the best and worst fold are denoted by shading.

for different values of N for MPL (top) or M for MC (bottom). We can see that for MC the optimal clipping value depends on both the number of rankings used to estimate document relevance as well as propensities. That the optimal clipping appears to be determined by the lower of the two, suggests that its value may be selected to minimize the variance arising from the term with the higher estimation noise.

For MPL, on the other hand, the optimal clipping parameter is fully independent of N and thus the computational cost, and can therefore be chosen to minimize the noise from the relevance estimates only. Consequently, MPL attains reduced MSE as the number of click rankings is increased, achieving comparable MSE with MC with $M = 10^6$ for PBM, and even outperforming MC with $M = 10^7$ under IPM. As there is also no discernible difference between MPL with $N = 50$ and $N = 1000$ (with the former being fully covered by the latter in the Figure) suggests the high applicability of MPL to OPE even at low runtime.

Taken together, our results thus strongly indicate an affirmative answer to **RQ3.3**: *MPL can provide considerably more accurate OPE estimates at a substantially lower computational cost.*

3.7 Conclusion

Off-policy estimators in learning to rank (LTR) rely on document placement probabilities to correct for various forms of statistical bias; however, obtaining sufficiently good propensity estimates may often be computationally challenging. In this work, we proposed Marginalized Plackett-Luce (MPL) – a novel method for estimating these probabilities under the Plackett-Luce (PL) ranking model. Our experimental results show that our approach is both highly accurate and more efficient compared to the existing sampling-based approach. We can thus answer our thesis research question **RQ2** in the affirmative: propensity estimation for PL models can be done in an accurate and computationally efficient way.

Furthermore, we demonstrate that MPL is highly applicable in practice, allowing us to obtain smaller off-policy evaluation errors at a lower computational cost. In particular, off-policy evaluation (OPE) performance using propensity estimates of MPL is comparable to that of the asymptotically unbiased sampling-based propensity estimation method using a large number of samples. Moreover, we observe that compared to the sampling-based approach with a comparable computational cost, MPL is able to apply lower propensity clipping, thus reducing its bias, whilst also taking advantage of larger amounts of user interactions, resulting in a lower OPE error. Our results thus also suggest a positive answer to our thesis research question **RQ3**: our propensity estimation method is able to reduce bias and improve data-efficiency for improved off-policy learning to rank performance.

Future work may wish to consider improving MPL’s efficiency for high ranking sizes as well as leverage MPL for model training. Overall, MPL reduces the computational cost of propensity estimation and opens the door for more accurate off-policy estimates under PL policies in LTR.

Chapter 4

Learning to Rank with Variable Result Presentation Lengths

Learning to rank (LTR) methods generally assume that each document in a top- K ranking is presented in an equal format. However, previous work has shown that users' perceptions of relevance can be changed by varying presentations, i.e., allocating more vertical space to some documents to provide additional textual or image information. Furthermore, presentation length can also redirect attention, as users are more likely to notice longer presentations when scrolling through results. Deciding on the document presentation lengths in a fixed vertical space ranking is an important problem that has not been addressed by existing LTR methods.

We address this gap by introducing the variable presentation length ranking task, where simultaneously the ordering of documents and their presentation length is decided. Despite being a generalization of standard ranking, we show that this setting brings significant new challenges: Firstly, the probability ranking principle no longer applies to this setting, and secondly, the problem cannot be divided into separate ordering and length selection tasks.

Plackett-Luce (PL) ranking model has recently seen widespread use in ranking due to its simplicity and the efficiency of its optimization methods. Our fourth thesis research question thus asks:

RQ4 *Can the Plackett-Luce ranking model be extended to simultaneously learn document ordering and presentation size in a computationally and sample-efficient way?*

To answer this question we propose *variable length Plackett-Luce* (VLPL) – a new family of Plackett-Luce list-wise gradient estimation methods for the joint optimization of document ordering and lengths. Our semi-synthetic experiments show that VLPL can effectively balance the expected exposure and attractiveness of all documents, achieving the best performance across different ranking settings. Furthermore, we observe that even simple length-aware methods can achieve significant performance improvements over fixed-length models. Altogether, our theoretical and empirical results highlight the importance and difficulties of combining document presentation with LTR.

4.1 Introduction

Ranking models generally aim to identify the most relevant documents from a set of candidates and present them to the user in the order of decreasing relevance on a search engine result page (SERP) [122]. However, the presentation of individual documents within a SERP is rarely considered when choosing the document ordering, with the documents usually simply modeled as fixed units of information or relevance. Nevertheless, in many cases the platforms can choose how each document is presented to the user, which in turn affects how the user perceives and interacts with those documents. We call this setting where both the order of the documents as well as their presentation lengths are chosen as the variable document presentation length setting – see Figure 4.1 for an example. This setting is characterized by the trade-off between presenting more information for displayed documents – potentially making them more attractive – or displaying more documents on the SERP – potentially including more relevant results.

Earlier research has shown that increasing the size of individual SERP results to include additional textual information may make the user more likely to click through to the full document [133], whilst shorter results may discourage the users from clicking [106]. Similarly, in e-commerce, adding images and product details has also been found to increase the engagement with those items [18, 222].

Making a search result larger may reduce the probability that a user will miss it when browsing [48], and thereby, alleviate some position bias on that result [130]. However, as noted by Pathak et al. [150, 151], increasing the vertical size of some results generally means that fewer results fit on the SERP [106]. In Figure 4.1, the black hat result is smaller in SERP 2 than in SERP 1 due to the larger t-shirt result. In general, increasing the

SERP 1	SERP 2	SERP 3
 Gray Jeans \$74.99 ★★★★★ Delivery \$5 or free from \$75!	 Gray Jeans \$74.99 ★★★★★ Delivery \$5 or free from \$75!	 Gray Jeans \$74.99 ★★★★★ Delivery \$5 or free from \$75!
 Blue T-shirt \$24.99 ★★★★★ Delivery \$5 or free from \$75!	Our Pick!  Blue T-shirt \$24.99 ★★★★★ Delivery \$5 or free from \$75! Other colors:   	 Blue T-shirt \$24.99 + \$5
 Black Hat \$24.99 ★★★★★ Delivery \$5 or free from \$75!	 Black Hat \$24.99 + \$5	 Black Hat \$24.99 + \$5
		 Green Socks \$19.99 ★★★★★ Delivery \$5 or free from \$75!

Figure 4.1: Rankings with variable presentation length. Compared to SERP 1, SERP 2 presents the second item with more vertical space (length), allowing more information to be displayed. In SERP 3, the second and third item are both presented with less vertical space, resulting in less information for each, but allowing an additional item to be displayed.

presentation size of an individual retrieved document also means that the documents below it are pushed down by the same amount. This, in turn, may make them less likely to be observed by the user and lead to an increase in the effects of position bias [45, 96, 100]. Hence this setting requires a balancing of document ordering and their presentation lengths. To the best of our knowledge, no existing method can simultaneously optimize a ranking and variable presentation lengths.

In this work, we formally introduce and study the variable document presentation length setting. Our analysis reveals challenges that are not present in standard ranking: most importantly, the probability ranking principle (PRP) does not apply, and as a result, the optimal placement strategy cannot be separated into independent ranking and length-selection strategies. In order to tackle this problem, we extend the Plackett-Luce (PL) ranking approach to optimize both the order and presentation lengths of documents. Our novel model, the variable length Plackett-Luce (VLPL) distribution π_{VLPL} , represents each choice as a document placement paired with a specific length. We build on the PL-Rank family of gradient estimators [143, 144] to introduce the novel VLPL-1 and VLPL-2 estimators. These efficiently estimate the gradient of π_{VLPL} , with VLPL-2 having a higher sample-efficiency. Moreover, as we show that the gradient estimation can be leveraged during model training as either in-processing or post-processing [36], in total we propose four distinct methods for jointly

optimizing ordering and length. Our semi-synthetic experiments demonstrate that our approaches achieve a large and consistent improvement over baseline methods. Overall, our findings underscore the need for ranking optimization methods with a more holistic view of result presentation that goes beyond the “*ten blue links*“, as some links may look very different from others, and perhaps there should not be ten.

4.2 Related Work

Learning to rank (LTR) models aim to identify the most relevant documents and present them to the user, most commonly in the form of a ranked list. This is usually done by scoring each document and then sorting the documents based on their scores. According to the PRP [165], a scoring function should reflect the probability of each document being relevant to the user to produce optimal rankings. In recent years, the Plackett-Luce family of list-wise methods [154] have gained popularity, owing to their simplicity, robustness and interpretability [26, 143, 144]. Importantly, however, these and other widely used ranking models [26, 30, 33, 61, 87, 143, 144, 211, 217] only view ranking from the lens of document ordering, not concerning themselves with any further choices about document presentation.

A large body of work has shown that result presentation may significantly affect how users perceive and interact with each document. Maxwell et al. [133] observed that users are more likely to interact with a result on a SERP if it is presented with a longer snippet. This is also consistent with the observations of Kohavi [106], who found that allocating the same amount of screen space to fewer but larger and more interactive ads led to significant increases in user engagement and conversions. Marcos et al. [130] reported that lower-ranking results with larger and more interactive snippets are noticed earlier and for longer compared to other snippets, with snippets containing images being particularly salient. You et al. [222] also found that the image of a SERP result may have a drastic impact on how likely the user is to find the document attractive [8, 41, 225].

On the other hand, the number of documents visible to the user at a given time also significantly affects user interactions [45, 55, 87, 96, 100, 145, 178, 197]. Kelly and Azzopardi [96] showed that having more results on each SERP leads to users interacting with more documents, whilst Kim et al. [100] found that the need to scroll, in contrast with pagination, significantly reduces the speed at which the users are able to browse the result list. Taken together, the above findings suggest that decreasing snippet length, which allows to include more results on a single SERP, may lead to a

higher number of interactions with the full ranking. Furthermore, the above results are also consistent with the findings of Joachims [87] on position bias [45, 48, 126], who saw that the probability that the user observes a document decreases with the position of that document, with particularly significant drops after the end of each page.

Our work is most related to that of Mao et al. [129] and Pathak et al. [150, 151]. Mao et al. aim to learn the causal graph based on the features of the SERP, which also includes document rank and document height. This causal graph is then used to correct for the impact of those variables on the click data in order to learn the intrinsic document relevance. However, the authors assume a fixed length for each document and do not learn how to the document lengths interact across different documents. Pathak et al. [151], on the other hand, focused on the setting where snippets can either contain the headline, summary, image, or any combination of the above, with each element taking up additional space on the screen. They find that changing the presentation of a document may also change where it should be placed in the ranking, particularly due to the increase in time the users spend examining the longer presentations. Nevertheless, the authors focus on the setting where the document lengths are chosen prior to determining the document order, whilst in Section 4.4.2 we prove that this can lead to suboptimal solutions. Finally, their approach assumes that user’s examination behavior can be measured separately for each document, and does not provide a ranking function for previously unseen queries.

4.3 Background: The Standard Ranking Setting

The core LTR task is to find a ranking $y = [d_1, d_2, \dots, d_K]$ of length K that orders documents by decreasing relevance ρ_i for a given query q and document set $d \in D$. To assess the quality of a ranking, per-position discounting factors $\theta(i)$ are commonly used, e.g., one-over-rank and DCG weighting. Stochastic ranking models [143, 203] can optimize a policy π that provides a probability distribution over rankings to maximize an objective of the form:

$$\mathcal{R}_{\text{LTR}}(q) = \mathbb{E}_{\pi(y)} \left[\sum_{i=1}^K \theta(i) \rho(d_i) \right] = \sum_y \pi(y) \sum_{i=1}^K \theta(i) \rho(d_i). \quad (4.1)$$

In click modeling [41], the reward ρ may often represent the perceived relevance or *attractiveness* of the document, i.e., the probability the user will click on the document when exposed to it.

The policy of a Plackett-Luce LTR model [143, 154] is defined as:

$$\pi_{\text{PL}}(y) = \prod_{i=1}^K \pi_{\text{PL}}(y_i \mid y_{1:i-1}) = \prod_{i=1}^K \frac{e^{m(d_i)} \mathbb{1}[d_i \notin y_{1:i-1}]}{\sum_{d'=1}^{|D|} e^{m(d')} \mathbb{1}[d' \notin y_{1:i-1}]}, \quad (4.2)$$

where $m(d)$ is the score for document d under some scoring function m . In the Plackett-Luce distribution, documents are sequentially sampled from a softmax distribution without replacement. Furthermore, the sampling probabilities in each position only depend on the scores of unplaced documents. Whilst the optimization of π_{PL} policies can be done with standard policy gradient methods (i.e., via REINFORCE [216]), Oosterhuis [143, 144] has proposed the PL-Rank family of methods, specialized to efficiently estimate gradients of Plackett-Luce distributions for LTR optimization.

Nevertheless, the standard ranking setting is not applicable if documents can be presented with different lengths. In particular, this is mainly due to two limitations of its objective (Equation 4.1): 1. the rank weights $\theta(i)$ only consider the ranks of documents and not their presentation lengths; and 2. a ranking always consists of K placed documents. However, as discussed in Section 4.2, the length of a document can affect the attractiveness and the amount of attention it receives from users [130, 133]. Therefore, $\theta(i)$ should adapt according to the attention that is expected from users [85] and, similarly, attractiveness $\rho(d_i)$ should also change with length. Furthermore, changing document presentation length should affect the number of documents that can be presented on a single SERP [151]. None of these aspects are captured by the standard ranking setting.

4.4 Variable Presentation Length Problem Setting

This section introduces a formalization of the variable presentation length ranking setting. Subsequently, we also discuss some of its theoretical implications.

4.4.1 Slot based ranking

To capture the unique characteristics of this setting, we represent each placement as a choice of document d and its length l , with a full ranking

being $y = [(d_1, l_1), (d_2, l_2), \dots, (d_{|y|}, l_{|y|})]$, where each d can only be placed once. Crucially, we assume that each ranking consists of K positions or slots: $\text{len}(y) = \sum_{j=1}^{|y|} l_j = K$. However, unlike the standard setting, each position now refers to a physical space in the ranking, not a moment of placement. The lengths l thus represent the number of slots each document occupies in the ranking. We also denote the first slot d_i occupies as $s_i = 1 + \sum_{j=1}^{i-1} l_j$. The above changes then allow us to define a new optimization objective called the expected attractiveness (EA) of a ranking:

$$\mathcal{R}(q) = \mathbb{E}_y \left[\sum_{i=1}^{|y|} \theta(s_i, l_i) \rho(d_i, l_i) \right] = \sum_y \pi(y) \sum_{i=1}^{|y|} \theta(s_i, l_i) \rho(d_i, l_i). \quad (4.3)$$

An important consequence of the above differences with standard ranking is that the number of unique documents $|y|$ in a ranking can be lower than K . Our task is then to choose the document-length pairs that maximize the weighted attractiveness in Equation 4.3 without exceeding the slot budget K or duplicate document placements. In contrast with standard ranking, both the position weight θ and the reward ρ depend on the presentation length l . For this work, we assume that ρ is fixed for each (d, l) , i.e., for document d each presentation length is associated with a single relevance value, which is the expected click-through rate (CTR) of d when exposed to the user. Finally, to make our analysis and approach tractable, we assume a maximum document length of L slots. Naturally, document lengths may in reality be adjustable even up to a single pixel or may instead be fixed (but distinct) for each document. We consider both of these important next steps to be addressed by future work.

Finally, we note that our proposed setting has some overlap with that of Pathak et al. [151]: they also capture that varying document presentation lengths can lead to different number of items appearing above the fold. See Section 4.2 for a further discussion.

4.4.2 Theoretical analysis on optimal rankings

A key aspect of our problem setting is the trade-off between extending document lengths and increasing the number of documents on the SERP. Importantly, this trade-off is present even when the attractiveness of a document is independent of its length. As illustrated in Table 4.1, in a simple three-slot ranking with documents A , B and C , both rankings $[(A, 2), (B, 1)]$ and $[(B, 1), (A, 2)]$ are optimal in terms of EA under different discount weights θ . Notably, in this example, under θ_2 extending the second item into the third slot makes the user slightly more likely to observe it

Table 4.1: The impact of discount weights on optimal ordering. For three documents A, B, C with respective relevances 1.0, 0.6 and 0 (top left, same for all lengths l), different choices of discount weights θ_1 (top center) and θ_2 (top right) with $K = L = 3$ lead to different expected rewards for different rankings (bottom). BAA represents ranking $y = [(B, 1), (A, 2)]$, whilst $AB(C)$ is equivalent to both AB and ABC . Best ranking per θ is underlined.

d	$\rho(d)$	θ_1	$l=1$	$l=2$	$l=3$	θ_2	$l=1$	$l=2$	$l=3$
A	1.0	s=1	0.500	0.667	0.750	s=1	0.631	0.815	0.895
B	0.6	s=2	0.333	0.500	-	s=2	0.500	0.715	-
C	0.0	s=3	0.250	-	-	s=3	0.431	-	-

	AAA	AAB	ABB	BAA	BBA	BBB	AA	AB(C)	BA
θ_1	0.750	<u>0.817</u>	0.800	0.800	0.650	0.450	0.667	0.700	0.633
θ_2	0.895	1.074	1.060	<u>1.094</u>	0.920	0.537	0.815	0.931	0.879

compared to the first item, and thus, placing the more relevant item second leads to a higher value of the EA metric. However, the observation that under θ_2 the most attractive document A should be placed second, contradicts the PRP [165], which states that placements should be in decreasing order of relevance. We thus conclude that PRP does not hold in the variable document presentation length setting.

Moreover, first choosing document lengths to maximize the exposure can similarly lead to suboptimal results. Whilst in the above example the optimal length for A is 2 under both values of θ , had document B not been available (i.e., only options are A and C), the optimal strategy in both cases would have instead been to present document A at length 3. Therefore, the above example proves that the task of choosing document length cannot be separated into separate ranking and document-specific length selection tasks:

Theorem 4.4.1. Let $\pi^*(l, r \mid d, D)$ be the optimal policy for jointly choosing r the rank of document d and l its presentation length. The optimal policy cannot be separated into a separate ranking policy and length selection policy:

$$\forall D, \exists (\pi_r, \pi_l), \forall (d, l, r) : \pi^*(l, r \mid d, D) = \pi_r(r \mid d, D) \pi_l(l \mid d). \quad (4.4)$$

Proof. Assume documents and position weights as in Table 4.1. The optimal length policy for document A is then $\pi_l(l = 2 \mid A) = 1$. If we only change the relevance of B to 0, the optimal length policy for A is $\pi_l(l = 3 \mid A) = 1$, thus there is no universal optimal π_l . $\square$

At first glance, it may seem feasible to solve the task with dynamic programming, but a further difficulty is that the optimal ranking for K slots is not necessarily an optimal sub-ranking for $K' > K$ slots. This can be also seen in Table 4.1: B should be placed ahead of A under θ_2 and $K = 3$, but the order is reversed under $K = 2$ (i.e., $[(A, 1), (B, 1)]$). Nevertheless, optimal rankings should vary document lengths due to the changes in exposure (and attractiveness) arising in this setting. For instance, we can see that under $K = 3$ the optimal single-length solutions $[(A, 1), (B, 1), (C, 1)]$, $[(A, 2)]$ and $[(A, 3)]$ are worse than the best mixed document length solution. Finally, we note that, whilst not shown in the above example, the optimal orderings may again be different for other values of ρ that maintain the relative order $A \succ B \succ C$ but not the ratio of their ρ values, e.g., if $\rho(B) = 0.01$ then $(A, 3)$ would always be preferred.

Altogether, whilst it appears crucial to present documents at different lengths and choose their ordering accordingly, the above examples reveal that there appears no straightforward optimal solution – moreover, we proved that this task cannot be broken down into separate ranking and length-selection policies. Accordingly, the remainder of this work proposes a framework for jointly optimizing document ordering and lengths.

4.5 Method: Variable Length Plackett-Luce

4.5.1 A distribution over rankings and lengths

We propose to build a novel model on top of the existing PL model for ranking (Equation 4.2); whereas the original only models a distribution over rankings, our novel model captures a distribution over both rankings and presentation lengths.

Our novel model also follows sequential sampling of document placements, but instead of sampling single documents d , document-length pairs (d, l) are sampled, where l is the presentation length of d . To avoid a document from appearing multiple times in the same ranking, the sampling of a (d, l) pair is followed by the removal of all other pairs that involve d . For our notation, we use the binary indicator variable $\mathbb{1}[d_i \notin y_{1:i-1}] = \mathbb{1}[\exists l : (d_i, l) \in y_{1:i-1}]$ to indicate earlier placements of a document. Another difference with standard ranking is that the lengths of placed documents affect how many can be displayed in the K slots: specifically, a pair (d, l) should not be sampled if fewer than l slots are unfilled

at the end of a partial ranking. Thus, we also introduce the indicator $\tau(s_i + l_i) = \mathbb{1}[s_i + l_i \leq K + 1] = \mathbb{1}[\text{len}(y_{1:i-1}) + l_i \leq K]$ to indicate if there are enough slots left for the placement of a document-length pair.

This notation then lets us define our novel *VLPL distribution*:

$$\pi(y) = \prod_{i=1}^{|y|} \pi(y_i \mid y_{1:i-1}) = \prod_{i=1}^{|y|} \pi(d_i, l_i \mid y_{1:i-1}), \quad (4.5)$$

$$\pi(y_i \mid y_{1:i-1}) = \frac{e^{m(d_i, l_i)} \mathbb{1}[d_i \notin y_{1:i-1}] \tau(s_i + l_i)}{\sum_{d'=1}^{|D|} \sum_{l'=1}^L e^{m(d', l')} \mathbb{1}[d' \notin y_{1:i-1}] \tau(s_i + l')}. \quad (4.6)$$

Analogous to the standard PL ranking model (Equation 4.2), the placement probability $\pi(d, l \mid y_{1:i-1})$ only depends on the score $m(d, l)$ and the scores of other eligible unplaced (d', l') pairs. However, our distribution differs in what pairs are considered eligible, as placing one pair can make several other pairs ineligible to avoid duplicate placements and space constraints. This goes beyond the *sampling without replacement* of the standard PL model, which can also be seen a special case $L = 1$ of our novel variable length model.

4.5.2 Variable length Plackett-Luce optimization

Previous work on Plackett-Luce optimization has found efficient techniques for the standard Plackett-Luce model [143, 144]. Fortunately, this existing approach can be re-used for our novel variable length model. In particular, we can apply the derivation steps corresponding to Equations 11-23 of Oosterhuis [143], with only minor changes to match our model and objective. For brevity and due to its high similarity, we do not repeat the full derivation here, but report the following resulting gradient:

$$\frac{\delta}{\delta m} \mathcal{R}(q) = \sum_{d=1}^{|D|} \sum_{l=1}^L \left[\frac{\delta}{\delta m} m(d, l) \right] \mathbb{E}_y \left[\underbrace{\sum_{i=r(d, l)+1}^{|y|} \theta(s_i, l_i) \rho(d_i, l_i)}_{\text{future reward after placement}} \right] \quad (4.7)$$

$$+ \underbrace{\sum_{i=1}^{r(d)} \pi(d, l \mid y_{1:i-1}) \left[\theta(s_i, l) \rho(d, l) - \sum_{x=i}^{|y|} \theta(s_x, l_x) \rho(d_x, l_x) \right]}_{\text{expected reward minus the risk of placement}} \quad (4.8)$$

Importantly, analogous to Oosterhuis [143], the gradient of $\mathcal{R}(q)$ w.r.t. the score of a single document-length pair can be broken down into three terms:

(i) the future reward observed following the placement, (ii) plus the expected direct reward of the placement, (iii) minus the expected risk imposed by the pair before it is placed.

Unsurprisingly, compared to PL-Rank [143], the gradient in Equation 4.7 is calculated for each (d, l) pair, with the reward ρ and weights θ now depending on the observed document lengths. However, a subtle but crucial difference is that we also differentiate between $r(d, l)$, which is the rank at which d is observed at length l , and $r(d)$, which is the rank of d at any length (the rank function is set to $|y|$ if its input is not observed). The distinction between $r(d)$ and $r(d, l)$ is important as, due to the ranking size limit K , placing the document d at distinct lengths may change which (d', l') can follow it. Therefore, the future reward term for each length l of d is thus included only when that particular pair (d, l) is observed. The risk term and the expected reward, however, accumulate up until d has been shown at any length, proportional to $\pi(d, l \mid y_{1:i-1})$, after which all lengths of d become ineligible for placement. If a presentation length l is too long, these terms are 0, as $\tau(s_i + l) = 0 \rightarrow \pi(d, l \mid y_{1:i-1}) = 0$.

The gradient formulation in Equation 4.7 can be turned into an estimator by replacing the expectation with an average over sampled rankings. For brevity, we omit the full equation but name this approach VLPL-1. To our knowledge, VLPL-1 is the first method to jointly optimize rankings and lengths in the variable document presentation length setting.

The remainder of this section describes how to efficiently sample rankings from the VLPL distribution (Section 4.5.3). Subsequently, in Section 4.5.4, we propose a more sample-efficient estimator through a smarter sampling technique for re-using samples. Finally, we discuss how VLPL can be applied in an in-processing and post-processing manner (Section 4.5.5).

4.5.3 Efficiently sampling rankings

Whilst it may appear that variable length rankings are substantially more difficult to sample than standard rankings [143], this can actually be done with a comparable computational complexity. The trick is to first sample an invalid ranking y' and then transform it with a function f such that $f(y')$ is a valid variable length ranking. We start by defining y' , which, in contrast with the variable length rankings as defined before, places d once at every possible length l , whilst ignoring the maximum ranking length K . Still, y' does not allow the placement of duplicate documents of the same

length. This gives us the following probability for document placements:

$$\pi'(d, l \mid y'_{1:i-1}) = \frac{e^{m(d,l)} \mathbb{1}[(d, l) \notin y'_{1:i-1}]}{\sum_{d'=1}^{|D|} \sum_{l'=1}^L e^{m(d',l')} \mathbb{1}[(d', l') \notin y'_{1:i-1}]}. \quad (4.9)$$

Let d'_i and l'_i indicate the document at position i in y' and its length respectively. The probability of the ranking y' is then simply the product of each placement probability:

$$\pi'(y') = \prod_{i=1}^{|D|L} p(d'_i, l'_i \mid y'_{1:i-1}). \quad (4.10)$$

We note that every document-length pair is placed in y' and thus $|y'| = DL$. Obviously, this makes y' an invalid variable length ranking as each document will appear L times in the ranking, and thus it is guaranteed to exceed the maximum length of K .

Our next step is to use a function f^K that turns y' into a valid ranking of length K , such that we end up with the same ranking distribution: $\pi(f(y') = y) = \pi(y)$. Our choice of f is a function that sequentially takes the first document in y' that does not violate either of the variable ranking constraints, until length K is reached:

$$\begin{aligned} f(y')_i &= y'_j \text{ where } j = \arg \min_{j'} j' \in \{1, 2, \dots, DL\} \\ &\text{s.t. } d'_{j'} \notin f(y')_{1:i-1} \wedge \text{len}(f(y')_{1:i-1}) + l'_{j'} \leq K. \end{aligned} \quad (4.11)$$

In other words, $f(y')_1 = y'_1$, then $f(y')_2$ is the next document that would not violate the validity of the ranking when placed, and so forth. To show that this yields the same distribution as π , we first note that the probability of a document placement: $\pi(f(y')_i = d \mid f(y')_{1:i-1})$ is the probability of it being a valid option and being ranked in front of all other valid options that are still available after $f(y')_{1:i-1}$. Following Luce's choice axiom [127], the ordering probabilities of the existing options in a PL model are not changed under the introduction of additional alternatives, i.e., $\Pr(a \succ b \mid \{a, b\}) = \Pr(a \succ b \mid \{a, b, c\})$. Thus, the availability of invalid options in y' does not change the relative ranking of valid options, and therefore $\pi(f(y')_i = (d, l) \mid f(y')_{1:i-1}) = \pi(y_i = (d, l) \mid y_{1:i-1})$ when $f(y')_{1:i-1} = y_{1:i-1}$. By extension, $\pi(f(y') = y) = \pi(y)$.

4.5.4 Extending VLPL for sample-efficiency

As previously suggested, VLPL-1 may have suboptimal sample-efficiency when calculating the future reward. In order to address this, we first start

by defining the total placement reward as:

$$g(y, i) = \sum_{x=i}^{|y|} \theta(s_x, l_x) \rho(d_x, l_x). \quad (4.12)$$

This allows us to re-formulate Equation 4.7 as:

$$g(y, d, l) = g(y, r(d, l) + 1), \quad (4.13)$$

$$h(y, d, l) = \sum_{i=1}^{r(d)} \pi(d, l \mid y_{1:i-1}) [\theta(s_i, l) \rho(d, l) - g(y, i)]. \quad (4.14)$$

The expected future reward with our sampling scheme is then:

$$\mathbb{E} \left[\sum_{i=r(d, l)+1}^{|y|} \theta(s_i, l_i) \rho(d_i, l_i) \right] = \mathbb{E}[g(y, d, l)] = \mathbb{E}[g(f(y'), d, l)]. \quad (4.15)$$

To increase the sample-efficiency, we want to reuse each y' . For this purpose, we introduce the following function: $f(y' \leftarrow d, l)$, which is the same transformation as f , except when d is being selected its length is set to l (instead of the actually sampled length). Furthermore, we introduce the variable $l(d, y')$, denoting the length of the first instance of d that is included in $f(y')$. Our aim is then to use samples arising from the sampling distribution of $l(d, y')$ to estimate the expected value under $l(d, y') = l$. We therefore introduce the following importance weight:

$$p(d, l, y') = \frac{p(l(d, y') = l)}{\sum_{l'=1}^L p(l(d, y') = l')} = \frac{e^{m(d, l)} \tau'(d, l, y')}{\sum_{l'=1}^L e^{m(d, l')} \tau'(d, l', y')}, \quad (4.16)$$

where $\tau'(d, l, y') = \mathbb{1}[\text{len}(f(y')_{1:r(d|f(y'))-1}) + l \leq K]$ is true only for the lengths at which d can be placed when it is observed in $f(y')$. This then enables the following reformulation:

$$\mathbb{E}_{y'} [g(f(y'), d, l)] = \mathbb{E}_{y'} [p(d, l, y') g(f(y' \leftarrow d, l), d, l)], \quad (4.17)$$

That is, we can use rankings y' with any observed length $l(d, y')$, process them for the target length l with $f(y' \leftarrow d, l)$ and reweigh the resulting future reward with $p(d, l, y')$. Repeating this for all valid lengths at each position yields the gradient estimate:

$$\begin{aligned} \frac{\delta}{\delta m} \mathcal{R}(q) &= \sum_{d=1}^{|D|} \sum_{l=1}^L \left[\frac{\delta}{\delta m} m(d, l) \right. \\ &\quad \cdot \mathbb{E}_{y'} [p(d, l, y') g(f(y' \leftarrow d, l), d, l) + h(f(y'), d, l)]. \end{aligned} \quad (4.18)$$

Furthermore, whilst calculating a separate $f(y' \leftarrow d, l)$ for each (d, l) pair may seem time-consuming, in practice it is actually not needed. We first define $\Delta \in [1 - L, 2 - L, \dots, L - 1]$ as the difference between the target length l and observed length $l(d, y')$. Importantly, for a source ranking y' which after applying $f(y')$ includes (d_i, l_i) and (d_j, l_j) , the adjusted rankings $f(y' \leftarrow d_i, l_i + \Delta)$ and $f(y' \leftarrow d_j, l_j + \Delta)$ in fact share the same tail, i.e., $z = \max(i, j) \rightarrow f(y' \leftarrow d_i, l_i + \Delta)_{z+1:} = f(y' \leftarrow d_j, l_j + \Delta)_{z+1:}$. We can thus use the same ranking to calculate the future reward for multiple such shifts, respectively reweighting each with $p(d_1, l_1 + \Delta, y'), p(d_2, l_2 + \Delta, y'), \dots, p(d_{|f(y')|}, l_{|f(y')|} + \Delta, y')$. Moreover, these future placements would also be the same if instead we still placed d_i at the original observed length l_i but then reduced the total number of available slots in the ranking by Δ , i.e., $f(y' \leftarrow d_i, l_i + \Delta)_{z+1:} = f^{K-\Delta}(y', d_i, l_i)_{z+1:} = f^{K-\Delta}(y')_{z+1:}$, where $f^{K'}(y')$ produces a ranking y of length K' instead of K .

As Δ can take on values between $1 - L$ and $L - 1$ (corresponding to replacing the longest length with the shortest and vice-versa), the future reward always arises from one of $2L - 1$ rankings $f^{K+\Delta}(y')$. However, whilst $f^K(y' \leftarrow d, l + \Delta)$ and $f^{K-\Delta}(y' \leftarrow d, l)$ lead to the same placements after d , the slot positions s_i in the latter would be shifted down by Δ . To maintain the consistency in the reward between the two cases and to ensure the correct range of θ 's inputs, we define the adjusted total reward as:

$$g'(y, i, \Delta) = \sum_{x=i}^{|y|} \mathbb{1}[s_x + \Delta \geq 1] \theta(s_x + \Delta, l_x) \rho(d_x, l_x). \quad (4.19)$$

We note that the indicator variable will always be one when calculating the future reward $g'(y, i + 1, \Delta)$ for $1 \leq l(d_i, y') + \Delta \leq L$.

Finally, as shown by Oosterhuis [144], the computation of a gradient estimator for the PL distribution can be further accelerated by utilizing helper variables, which are first pre-computed and re-used for all positions. Along with applying Equation 4.19, we can thus express the terms in Equation 4.18 as:

$$\text{PR}_{y,i}^\Delta = g'(f^{K-\Delta}(y), i, \Delta), \quad \text{PR}_{y,(d)}^\Delta = \text{PR}_{y,r(d)+1}^\Delta, \quad (4.20)$$

$$\text{RI}_{y,i} = \sum_{x=1}^{\min(|y|, i)} \frac{\text{PR}_{y,i}^0}{\sum_{d'=1}^{|D|} \sum_{l'=1}^L e^{m(d', l')} \mathbb{1}[d \notin y_{1:x-1}] \tau[s_x + l']}, \quad (4.21)$$

$$\text{DR}_{y,i,l} = \sum_{x=1}^{\min(|y|, i)} \frac{\theta(s_x, l)}{\sum_{d'=1}^{|D|} \sum_{l'=1}^L e^{m(d', l')} \mathbb{1}[d \notin y_{1:x-1}] \tau[s_x + l']}, \quad (4.22)$$

$$\text{RI}_{y,(d,l)} = \text{RI}_{y, \min(r(d), r(l))}, \quad \text{DR}_{y,(d,l)} = \text{DR}_{y, \min(r(d), r(l)), l}, \quad (4.23)$$

where $r(l)$ is $\sup i : \text{len}(f(y')_{1:i-1}) + l \leq K$, i.e., the last position at which a document of length l can be placed, and $r(d)$ is the rank of d in $y^0 = y$. This, in turn, allows us to reformulate Equation 4.7 as:

$$\begin{aligned} \frac{\delta}{\delta m} \mathcal{R}(q) = & \sum_{d=1}^{|D|} \sum_{l=1}^L \left[\frac{\delta}{\delta m} m(d, l) \right] \mathbb{E}_{y'} \left[p(d, l, y') \text{PR}_{y', (d)}^{l-l(d, y')} \right. \\ & \left. + e^{m(d, l)} \left(\rho(d, l) \text{DR}_{f(y'), (d, l)} - \text{RI}_{f(y'), (d, l)} \right) \right]. \end{aligned} \quad (4.24)$$

This gradient can be approximated using the sampling procedure for π' described in Subsection 4.5.3, yielding our estimator VLPL-2:

$$\begin{aligned} \frac{\delta}{\delta m} \mathcal{R}(q) = & \sum_{d=1}^{|D|} \sum_{l=1}^L \frac{\delta}{\delta m} m(d, l) \sum_{i=1}^N \frac{1}{N} \left[p(d, l, y'^{(i)}) \text{PR}_{y'^{(i)}, (d)}^{l-l(d, y'^{(i)})} \right. \\ & \left. + e^{m(d, l)} \left(\rho(d, l) \text{DR}_{f(y'^{(i)}), (d, l)} - \text{RI}_{f(y'^{(i)}), (d, l)} \right) \right]. \end{aligned} \quad (4.25)$$

This estimator has higher sample-efficiency than VLPL-1 as it shares the future reward across different lengths of the same document. The risk and the expected direct reward terms RI and DR are still calculated using the standard ranking $y = y^0$ of length K . The exact steps of our approach are also outlined in Algorithm 1.

Finally, we note that VLPL-1 can similarly be implemented with PL-Rank-3-like sample-efficiency by using the reward $\text{PR}_{y', r(d, l)+1}^0$ in place of $\text{PR}_{y', r(d)+1}^\Delta$ and setting $p(d, l, y')$ to 1.

4.5.5 Using VLPL as in- or post-processing

Existing LTR work mainly aims to optimize machine learning models for the ranking task [26, 143, 145]; we will refer to this as an *in-processing* application of LTR. However, LTR gradients can also be used to optimize a ranking directly, i.e., when m is a look-up table instead of a model. This also allows for a *post-processing* application of LTR, and by extension VLPL, where the reward values ρ are predicted, and subsequently, the ranking is optimized for these predictions. Post-processing is not very popular for standard ranking, as the optimal ranking simply sorts according to ρ , thus there is little added value, while it adds substantial costs during inference. However, in our variable document length setting the PRP does not hold and, therefore, post-processing could actually result in better rankings than is possible with a standard scoring model. For similar reasons, post-processing ranking has also been used for ranking fairness [36]. In this work, we evaluate VLPL both applied as an in-processing method and a post-processing method.

Algorithm 1 VLPL-2 Gradient Estimator

```

1: Input: Items:  $D$ ; Scores:  $m$ ; Relevances:  $\rho$ ; Position weights:  $\theta$ ; Slot
   number:  $K$ ; Item length limit:  $L$ ; Sample number:  $N$ .
2:  $\{y'_{(1)}, y'_{(2)}, \dots, y'_{(N)}\} \leftarrow \text{SampleRankings}(m, N, K, L)$ 
3:  $\text{Grad} \leftarrow \mathbf{0} \in \mathbb{R}^{|D| \times L}$ 
4:  $S_1, \dots, S_L \leftarrow \sum_{d=1}^{|D|} \exp(m(d, 1)), \dots, \sum_{d=1}^{|D|} \exp(m(d, L))$ 
5: for  $j \in [1, 2, \dots, N]$  do
6:   // Each  $y^x = [(d_1^x, l_1^x), \dots, (d_{|y^x|}^x, l_{|y^x|}^x)]$ ,  $s_i^x = 1 + \text{len}(y_{1:i-1}^x)$ 
7:    $y^{-L+1}, \dots, y^{L-1} \leftarrow f^{K+L-1}(y'_{(j)}), \dots, f^{K-L+1}(y'_{(j)})$ 
8:    $S'_1, \dots, S'_L \leftarrow S_1, \dots, S_L$ 
9:    $\text{PR}_{K+L}^{-L+1}, \dots, \text{PR}_{K+L}^{L-1} \leftarrow 0, \dots, 0$ 
10:  for  $i \in [K+L-1, K+L-2, \dots, 1]$  do
11:    for  $\Delta \in [-L+1, \dots, L-1]$  do
12:       $u = \mathbb{1}[|y^\Delta| \geq i] \mathbb{1}[s_i + \Delta \geq 1] \rho(d_i^\Delta, l_i^\Delta) \theta(s_i^\Delta + \Delta, l_i^\Delta)$ 
13:       $\text{PR}_i^\Delta \leftarrow \text{PR}_{i+1}^\Delta + u$ 
14:     $\text{RI}_0, \text{DR}_0^1, \dots, \text{DR}_0^L \leftarrow (0, 0, \dots, 0)$ 
15:    for  $i \in [1, \dots, K]$  do
16:       $S' = \sum_{l=1}^L S'_l$ 
17:       $\text{RI}_i \leftarrow \text{RI}_{i-1} + \text{PR}_i^0 / S'$ 
18:      for  $l \in [1, \dots, L]$  do
19:         $\text{DR}_i^l \leftarrow \text{DR}_{i-1}^l + \theta(s_i^0, l) / S'$ 
20:         $S'_l \leftarrow \mathbb{1}[l \leq K - (s_i^0 + l_i^0) + 1] (S'_l - \exp(m(d_i^0, l)))$ 
21:         $p_i^l \leftarrow \exp(m(d_i^0, l)) \mathbb{1}[l \leq K - s_i^0 + 1]$ 
22:      for  $d \in [1, \dots, |D|]$  do
23:        for  $l \in [1, \dots, L]$  do
24:           $r_1 \leftarrow \min(\text{rank}(d, y^0), K)$ 
25:           $r_2 \leftarrow \min(r_1, \sup i : \{l \leq K - s_i^0 + 1\})$ 
26:           $u \leftarrow \exp(m(d, l)) [\rho(d, l) \cdot \text{DR}_{r_2}^l - \text{RI}_{r_2}]$ 
27:           $\text{Grad}(d, l) \leftarrow \text{Grad}(d, l) + p_{r_1}^l / \sum_{l'=1}^L p_{r_1}^{l'} \cdot \text{PR}_{r_1+1}^{l-l_{r_1}^0} + u$ 
28: return  $\text{Grad} / N$ 

```

4.6 Experimental Setup

Our aim is to better understand the variable presentation length setting and evaluate our proposed methods for this task. To begin, as discussed in Section 4.4.2, there is no known method that can provide the optimal ranking and lengths in this setting, even when all the attractiveness values ρ for the query are available (i.e., the oracle setting). We are therefore interested in determining whether VLPL can find high reward rankings,

both when the ρ values are given or unknown. Thus our first two (chapter) research questions are:

RQ4.1 When provided with true attractiveness labels, does VLPL learn rankings with a higher EA metric than the baselines?

RQ4.2 Does VLPL also learn higher EA rankings when it does not have access to the true attractiveness?

Furthermore, as it is unclear what optimized variable length rankings look like, our third research question asks:

RQ4.3 What document lengths does VLPL place in each position for different choices of θ ?

Finally, since VLPL uses samples for its gradient estimation, it seems important to understand its sample-efficiency. Our final research question is therefore:

RQ4.4 Does VLPL have comparable performance with a lower number of sampled rankings?

Thus, our research questions cover different aspects of VLPL’s performance and the variable presentation length setting.

4.6.1 Datasets and length-specific attractiveness

To our knowledge, there are no datasets featuring different presentation lengths for the same document that can be used to answer our research questions. We thus use *Yahoo! Webscope* [37] and *MSLR-WEB30k* [156] datasets, consisting of queries and associated documents. Each query-document pair is represented by a feature vector and an expert relevance judgement label $R \in \{0, 1, \dots, 4\}$. Yahoo contains 29,921 queries with an average of 24 documents per query, whilst MSLR consists of 30,000 queries with an average of 125 documents per query. For computational efficiency we focus on the queries with 250 or fewer documents, retaining all queries in Yahoo and over 95% of queries in MSLR.

To generate L attractiveness labels for every document, we first remap the labels in Yahoo to $5L$ non-overlapping equally sized bins. For a document d with original label R , the attractiveness ρ is higher for longer document presentation lengths l , yet is still primarily determined by R : $\rho \in [(RL + l - 1)/(5L), (RL + l)/(5L)]$. On the other hand, due to the higher number of documents per query in MSLR, we emphasize the most relevant documents and double each bin’s size relative to the last, renormalizing their sum.

In order to determine $\rho(d, l)$, we then follow Fang et al. [55], generating $L + 1$ scores using query-document features. L of these are normalized with other scores in (d, l) 's bin and map to a quantile within that bin's range, yielding $\rho(d, l)$. Furthermore, as some documents may not actually benefit from a longer presentation [197], we randomly reorder the ρ values of different lengths for half of the documents determined by the $L + 1$ 'th score.

4.6.2 Models, baselines and training

For our in-processing models, we use VLPL-1 and VLPL-2 to calculate the gradient of $\frac{\delta}{\delta m} \mathcal{R}(q)$ in order to train a multilayer perceptron (MLP) that learns L scores $m(d, l)$ in Equation 4.6 for each query-document pair based on its features. In contrast, for the post-processing approach, we use the binary cross-entropy loss to first train an MLP to predict ρ (instead of m). Then for each test query, we instantiate a look-up table with a single score for each (d, l) pair, and use VLPL-1/2 to learn these scores to maximize Equation 4.3 specifically for that query, using the outputs of MLP in place of ρ .

We contrast our four VLPL models with the following baselines. The first method family, referred to as sort- l , similar to the VLPL post-processing, use an MLP to estimate ρ for a single length l to sort and place documents of only that length. In addition, we train L separate single-length PL-Rank-3 models using in-processing, which we denote as PLR-3- l . For length-adaptive baselines, the greedy model uses MLP estimates of ρ to repeatedly select an eligible (d, l) pair maximizing $\theta(s, l)\rho(d, l)$. However, the above policy may also prefer one longer document over multiple shorter yet only slightly less relevant ones. The slot-avg model thus instead ranks (d, l) pairs on $\theta(s, l)\rho(d, l)/l$, in effect choosing the document with the highest per-slot expected reward. Finally, we also evaluate the above models with full access to the true relevances (oracle setting). In this case the fixed-length sorting models are an upper bound to PL-Rank-3, which is therefore omitted.

All models contain 1-5 hidden layers of size $\{16, 32, \dots, 128\}$ and sigmoid activations, trained with Adam optimizer [102] with learning rate, $L2$ regularization in $\{0, 1^{-6}, 1^{-5}, \dots, 1\}$ and dropout in $[0, 0.8]$. We report test set results over 5 independent runs performed under identical circumstances for the best configuration of each model, selected on the validation set results of the first run. VLPL trained with $N = 10,000$ sampled rankings per update step. All models trained in PyTorch [149] on a single NVIDIA RTX A5000 GPU.

4.6.3 Evaluation and position bias

We evaluate all models using the EA objective in Equation 4.3. We fix the number of slots K at 30 and maximum document length L at 3, representing one possible production setting. Under the standard rank-based position bias model [145], users are assumed to observe the contents of each position with a probability $\theta(i)$ that only depends on the position. As in our setting a document can span multiple positions, we consider the document to be observed as long as any of its slots are observed, yielding $\theta(s, l) = 1 - \prod_{i=s}^{s+l-1} (1 - \theta(i, 1))$, where $\theta(i, 1)$ is equivalent to $\theta(i)$ in Equation 4.1 – the standard observation probability for rank i [88]. We evaluate two choices of $\theta(i)$ – the slowly decaying DCG weights $\theta_{\text{DCG}}(i) = 1/\log_2(i+1)$ and the steeper inverse rank $\theta_{\text{rank}^{-1}}(i) = 1/i$, corresponding to DCG and the sum of reciprocal ranks if $\forall l = 1$.

4.7 Results

4.7.1 Ranking with known relevance

We start by addressing the first research question (**RQ4.1**): whether VLPL is able to more efficiently order documents and their lengths to achieve higher reward when compared to the baselines, when the attractiveness of each document is known. Table 4.2 shows the performance in terms of EA in this oracle setting where the number of slots $K = 30$ and maximum document length is $L = 3$ slots.

We can see that adaptively selecting document lengths has a very large impact on the EA metric. Both variable length baselines achieve a substantial and consistent improvement over the single-length baselines, with the greedy model performing better under θ_{DCG} and the slot-avg model with $\theta_{\text{rank}^{-1}}$. Different strategies also achieve the best performance among the single-length models in different settings, highlighting the importance of tailoring the document presentation to the setting where the ranking is shown.

On the other hand, both VLPL-1 and VLPL-2 demonstrate a very large and statistically significant improvement over *all* models across every setting, with a relative performance increase in EA_{DCG} over the best baseline on Yahoo exceeding 10%. As such, VLPL is clearly able to generate highly attractive rankings in this setting. Section 4.7.4 further compares the performance of VLPL models.

Table 4.2: EA for $K = 30$ and $L = 3$ in the oracle setting with DCG and inverse rank slot weights. Results are averages over five independent runs; standard deviations in parentheses. Best result in each task denoted in bold. Significant improvement of a model over all (other) baselines denoted by $\triangle$ (separate one-sided Wilcoxon signed rank tests, $p < 0.05$).

	MSLR		Yahoo!	
Model	EA _{DCG}	EA _{rank⁻¹}	EA _{DCG}	EA _{rank⁻¹}
sort-1	1.500 (0.002)	0.908 (0.002)	2.832 (0.000)	1.695 (0.000)
sort-2	1.628 (0.005)	0.925 (0.004)	3.113 (0.000)	1.680 (0.000)
sort-3	1.733 (0.010)	0.981 (0.006)	3.137 (0.000)	1.662 (0.000)
greedy	1.917 (0.008)	1.086 (0.005)	3.326 $\triangle$ (0.000)	1.778 (0.000)
slot-avg	1.911 (0.005)	1.113 $\triangle$ (0.003)	3.282 (0.000)	1.926 $\triangle$ (0.000)
VLPL-1	2.064$\triangle$ (0.008)	1.187$\triangle$ (0.005)	3.665 $\triangle$ (0.000)	2.031$\triangle$ (0.000)
VLPL-2	2.064$\triangle$ (0.008)	1.187$\triangle$ (0.005)	3.670$\triangle$ (0.000)	2.030 $\triangle$ (0.000)

Overall, we observe that adaptively choosing document presentation length is very important, with particularly high performance shown by VLPL. We thus answer **RQ4.1** positively: when true attractiveness is known, compared to the baselines, VLPL is able to find significantly more attractive variable presentation length rankings.

4.7.2 Ranking with learned relevance

We have established that VLPL can effectively optimize variable presentation length rankings when documents’ attractivenesses are given. Nevertheless, for previously unseen queries these are unknown. Accordingly, we aim to answer our second research question (**RQ4.2**): can VLPL be used to generalize to unseen queries?

As shown in Table 4.3, the performance of all models is substantially lower compared to the oracle setting (cf. Table 4.2). Here, sort-3 now also consistently performs the best among sort- l models. Together, these observations suggest that relevance misestimation can have a significant impact on learned rankings in this setting.

Nevertheless, both in-processing and post-processing versions of VLPL still achieve statistically significant improvements over all baselines across all settings. We observe particularly high performance for the post-processing approaches, and especially VLPL-2. The high performance of VLPL strongly suggests that being able to tailor document length choices to the specific

Table 4.3: EA for $K = 30$ and $L = 3$ with learned relevance and DCG and inverse rank slot weights. Results are averages over five independent runs, with standard deviations in parentheses. Best result for each task denoted in bold. Best in-processing model in each task is underscored. Significant improvement over all baselines denoted by $\triangle$ (separate one-sided Wilcoxon signed rank tests, $p < 0.05$).

	MSLR		Yahoo!	
Model	EA _{DCG}	EA _{rank⁻¹}	EA _{DCG}	EA _{rank⁻¹}
In-processing				
PLR-3-1	0.777 (0.008)	0.421 (0.003)	2.466 (0.001)	1.385 (0.000)
PLR-3-2	0.839 (0.006)	0.458 (0.003)	2.699 (0.001)	1.394 (0.002)
PLR-3-3	0.931 (0.016)	0.522 (0.008)	2.761 (0.001)	1.423 (0.001)
VLPL-1	0.953 $\triangle$ (0.019)	<u>0.533</u> $\triangle$ (0.008)	<u>2.982</u> $\triangle$ (0.021)	<u>1.603</u> $\triangle$ (0.001)
VLPL-2	<u>0.959</u> $\triangle$ (0.018)	0.527 $\triangle$ (0.005)	2.976 $\triangle$ (0.002)	1.590 $\triangle$ (0.003)
Post-processing				
sort-1	0.781 (0.005)	0.419 (0.002)	2.469 (0.001)	1.385 (0.001)
sort-2	0.843 (0.006)	0.452 (0.003)	2.698 (0.001)	1.404 (0.001)
sort-3	0.929 (0.015)	0.513 (0.010)	2.757 (0.001)	1.424 (0.001)
greedy	0.935 (0.018)	0.492 (0.011)	2.747 (0.001)	1.428 (0.005)
slot-avg	0.896 (0.020)	0.516 (0.007)	2.600 (0.010)	1.520 (0.001)
VLPL-1	0.971 $\triangle$ (0.020)	0.535 $\triangle$ (0.009)	3.042 $\triangle$ (0.001)	1.627 $\triangle$ (0.000)
VLPL-2	0.971 $\triangle$ (0.020)	0.536 $\triangle$ (0.008)	3.049 $\triangle$ (0.001)	1.628 $\triangle$ (0.001)

document pool of each query may be beneficial in the variable length ranking setting. Interestingly, VLPL-1 appears to generally perform better than VLPL-2 for in-processing. As the performance of both VLPL models, and particularly VLPL-2, is higher for post-processing, we speculate that in-processing may introduce additional noise in the future reward and which may be amplified by reward-sharing of VLPL-2.

Nevertheless, we observe that all VLPL methods perform the best across all our settings. Therefore, we can answer **RQ4.2** affirmatively: VLPL methods also achieve the highest EA on unseen queries with learned attractiveness.

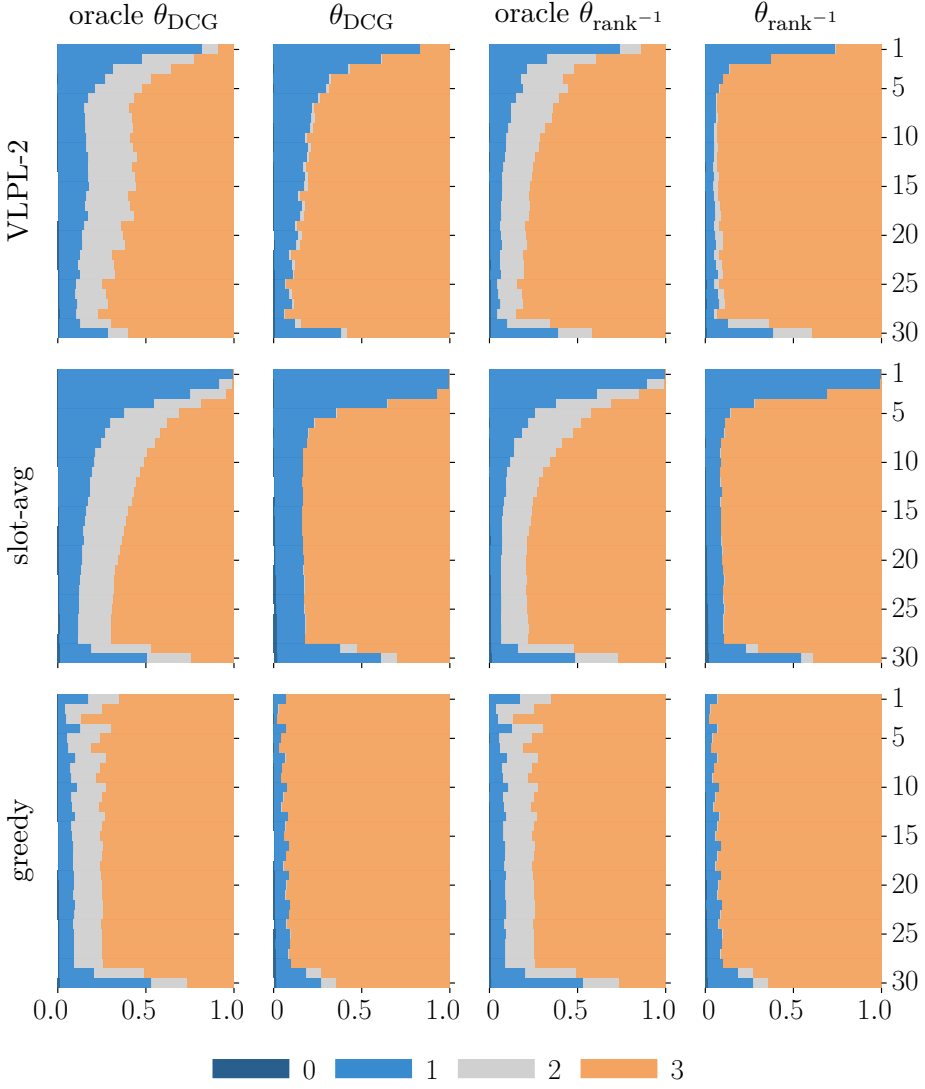


Figure 4.2: Distribution of document lengths occupying each slot in learned rankings in MSLR. y-axis of stacked bar chart denotes the slot, x-axis denotes the proportion of rankings in the first test fold where the slot is occupied by the corresponding length. Length 0 denotes padding.

4.7.3 Learned rankings

As we’ve shown that VLPL can learn high reward rankings in the variable presentation length task, we then turn to **RQ4.3**: what document lengths does VLPL place in the learned rankings?

The lengths of documents occupying each slot for MSLR are shown in Figure 4.2. We first examine the behavior of VLPL-2 in the oracle setting, which corresponds to the task of finding the best possible ranking. We see that VLPL-2 starts with a high probability of placing documents at length 1 at the top of the ranking, followed by the increase in the proportion of lengths 2 and 3. Length 3 then occupies the vast majority of the ranking in the final positions. Such behavior of showing lower-ranked documents at longer length was previously found to be beneficial by Marcos et al. [130]. The shift from shorter to longer lengths is especially pronounced for $\theta_{\text{rank}^{-1}}$, suggesting that longer lengths and the associated increase in exposure may be particularly useful under stronger position bias.

Surprisingly, when document attractiveness is not known and instead estimated, VLPL follows a pattern that is roughly similar to the one in the oracle setting, yet with almost no use of length 2. We hypothesize that this may be due to the misestimation of ρ , which, as discussed in Subsection 4.4.2 may lead to a different solution even if the relative ordering of all estimated $\rho(d, l)$ values is correct.

In contrast to VLPL, the slot-avg model places documents at a short length for longer, whilst the greedy model strongly prefers maximum length for most of the ranking, producing highly similar rankings for different θ . As the performance of VLPL-2 was previously shown to be significantly higher, the difference between models across different settings suggests that balancing document lengths is a critical factor that allows VLPL-2 to achieve strong performance, particularly in the oracle setting.

We saw that VLPL-2 strongly modulates its lengths. As such, for **RQ4.3** we conclude that: VLPL places shorter documents at the start of the ranking and progressively increases the placed document lengths, with a quicker transition for $\theta_{\text{rank}^{-1}}$.

4.7.4 Sample-efficiency

We then arrive at our final research question (**RQ4.4**): can VLPL models also achieve strong performance whilst using fewer samples? To answer this, Figure 4.3 shows VLPL performance under a varied number of samples N in the oracle and non-oracle settings.

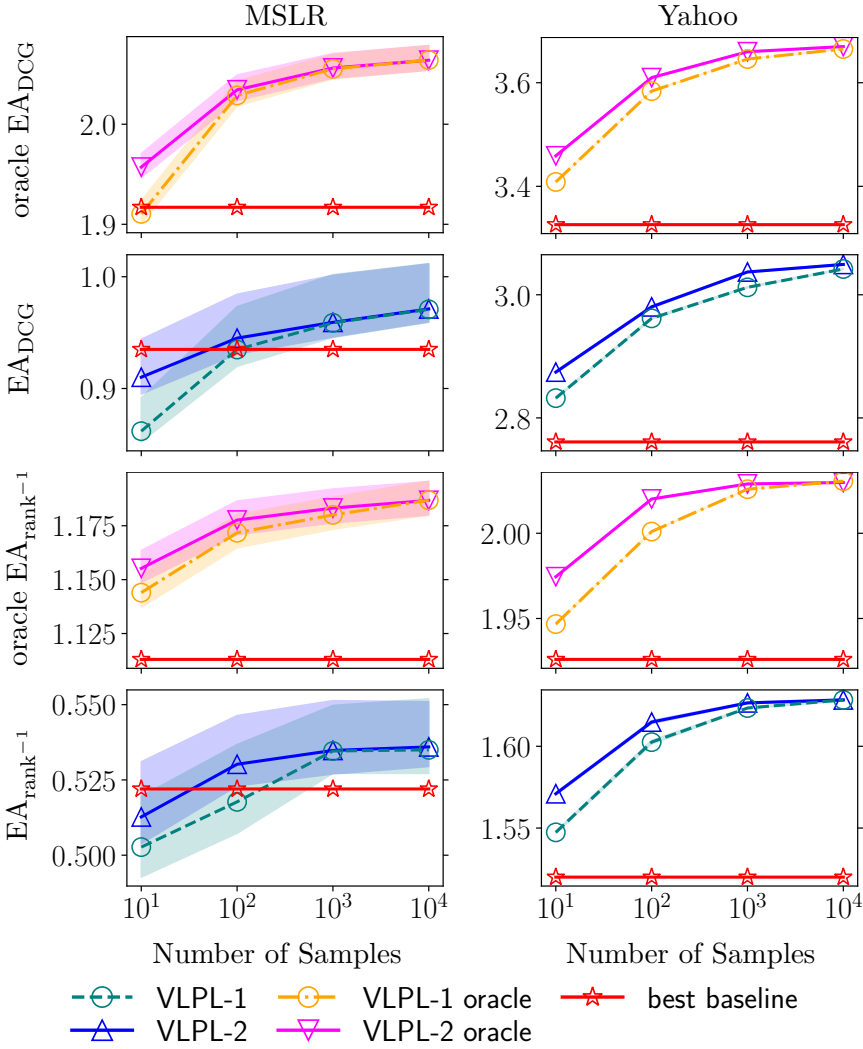


Figure 4.3: EA for post-processing VLPL models under different number of samples. Results are averages over five independent runs, the difference between the best and worst runs denoted by shading. Best baseline per setting denoted by star.

We can see that increasing the number of samples leads to improvements in EA, with smaller, but still observable gains at larger sample numbers. We also observe that whilst the performance of VLPL-1 and VLPL-2 is very similar at $N = 10,000$, VLPL-2 shows stronger performance at lower values of N , with particularly substantial improvements over VLPL-1 at $N \leq 100$. Nevertheless, for $N \geq 100$, both VLPL models generally achieve strong performance and always outperform the best baseline at $N = 1,000$, highlighting the high applicability of VLPL to the variable presentation

length ranking task. As such, in response to **RQ4.4**, we affirm that: VLPL models, and especially VLPL-2, achieve large improvements over the baselines even when using fewer samples.

4.8 Conclusion and Future Work

In this paper we introduced the variable presentation length ranking setting, where the task is to simultaneously choose the document order and the size of each document’s presentation in the ranking. We showed that this setting is substantially more complex than standard ranking – the best performance can only be achieved when document order and presentation lengths are decided jointly. To tackle this problem, we introduced the variable document length Plackett-Luce model, and four VLPL methods for optimizing it. Our experiments show that VLPL is highly suited for ranking with variable presentation lengths and underscore the importance of considering document presentation in LTR.

As such, we also answer our thesis research question **RQ4** affirmatively: PL ranking model can be successfully extended to simultaneously learn document ordering and presentation size in a computationally and sample-efficient way, especially in the form of VLPL-2.

Importantly, whilst we model document presentation lengths in terms of slots, it may also be possible to extend it to continuous values to represent pixel lengths of each document. Furthermore, as different policies may be best suited for different presentation contexts, being able to learn a policy that is guaranteed to be effective across multiple settings (e.g., on mobile and desktop) may further help the application of such models. Altogether, our work highlights the importance, the difficulty and the opportunities of LTR with variable result presentation lengths.

Chapter 5

A Lightweight Method for Modeling Confidence in Recommendations with Learned Beta Distributions

Most recommender systems (RecSys) do not provide an indication of confidence in their decisions. Therefore, they do not distinguish between recommendations of which they are certain, and those where they are not. Existing confidence methods for RecSys are either inaccurate heuristics, conceptually complex or computationally very expensive. Consequently, real-world RecSys applications rarely adopt these methods, and thus, provide no confidence insights in their behavior.

Among existing approaches, prediction variance appears to be a natural method for representing model confidence. Nevertheless, the lack of computationally efficient and accurate confidence modeling methods in RecSys leads us to ask our fifth and final thesis research question:

RQ5 *Can variance estimation be incorporated into rating prediction algorithms as confidence modeling in a computationally efficient manner?*

In this work, we propose *learned beta distributions* (LBD) as a simple and practical recommendation method with an explicit measure of confidence. Our main insight is that beta distributions predict user preferences as probability distributions that naturally model confidence on a closed interval, yet can be implemented with the minimal model-complexity. Our results show that LBD maintains competitive accuracy to existing methods while

also having a significantly stronger correlation between its accuracy and confidence. Furthermore, LBD has higher performance when applied to a high-precision targeted recommendation task.

Our work thus shows that confidence in RecSys is possible without sacrificing simplicity or accuracy, and without introducing heavy computational complexity. Thereby, we hope it enables better insight into real-world RecSys and opens the door for novel future applications.

5.1 Introduction

In order to best serve their users, recommender systems (RecSys) generally model user preferences to predict which item recommendations they would like [164, 189]. Generally, these models are constructed with collaborative filtering, where to predict the preference of one user-item combination, one considers what similar users thought of that item [131]. The most fundamental of these methods is Matrix Factorization (MF) [112] which represents users and items with embeddings, i.e., numerical vectors, where a high similarity between embeddings leads to a positive preference prediction, and vice versa. Due to its effectiveness and simplicity, MF is still one of the most competitive and widely-used RecSys methods [162].

In spite of being a heavily studied core RecSys problem, the modeling of user preferences remains a very difficult task [164]. Part of this is due to the inherent uncertainty in user preferences: human behavior is complex and stochastic, and therefore, impossible to predict with perfect accuracy [9, 104]. As a result, it is thus inevitable that even the best RecSys models will make errors in their predictions. However, despite these expected errors, the most commonly used RecSys models do not give any sense of uncertainty in their predictions [111, 205]. In other words, these models do not distinguish between predictions with high confidence, of which they are certain, or those with low confidence, of which they are unsure. Consequently, current RecSys give no insight into when recommendations are made with reliable confidence or doubtful guesswork [79, 103].

Naturally, these shortcomings make the idea of confidence modeling for RecSys very appealing. A good measure of confidence could give insight into the behavior of a system, and give an indication of the extent of the error one can expect for a particular prediction. [136, 160]. Furthermore, confidence enables functionalities such as exploration: recognizing which unexplored items have the potential of a high preference [40]; or conservative safety: guaranteeing a certain level of user experience by only following high-confidence preferences [139]. In addition, uncertainty modeling could

make RecSys more explainable [232], and enable more insightful studies of user preferences.

In response, previous work proposed RecSys with confidence by applying generative probabilistic models [169, 205]. To the best of our knowledge, all existing work assumes user ratings arise as samples from Gaussian probability distributions, where the ratings predicted by a MF model provide the means of each distribution [137]. Salakhutdinov and Mnih [169] then apply a Bayesian approach and infer a probability distribution over embeddings that captures the uncertainty in the data. But this inference has an impractically heavy computational cost, even when approximated, which seems to have prevented widespread real-world adoption. As an alternative, Wang et al. [205] use static embeddings but vary the standard deviations of the Gaussians to model different levels of uncertainty. However, this approach is very limited in the confidence patterns it can express, i.e., it only lets uncertainty vary on user and item levels separately and not over specific user-item combinations. The lack of user-item interactions for confidence modeling is similarly a problem for Koren and Sill [111], who instead capture the uncertainty by learning user-specific thresholds to assign each region of a Gaussian-like density to a single rating value. The total density of each region then represents the probability of observing that rating. While this allows confidence to vary, this solution is limited in the patterns it can express and does not appear easily interpretable. As such, it appears the current RecSys field has no method that effectively models uncertainty in an elegant and interpretable fashion, whilst also having practical computational requirements.

In this work, we introduce the learned beta distributions (LBD) as a simple and lightweight RecSys method for a natural and explicit model measure of confidence. Our method transforms a standard beta distribution into a discrete rating distribution, based on only two parameters that can be modeled with the same model-complexity as MF. By utilizing close approximations of their gradients [166, 199], LBDs are optimized without high computational costs, thus making them very practical to work with. Furthermore, LBDs can vary their confidence over different user-item combinations, with enough model expressiveness to capture complex patterns in uncertainty. Our experimental results show LBD has performance competitive with MF and the confidence models of Wang et al. [205] and Koren and Sill [111], while having a considerably stronger correlation between its confidence and its accuracy across ratings.

With the introduction of LBD for RecSys with confidence, we thus show that confidence modeling in RecSys is possible without sacrificing accuracy

or adding heavy computational complexity. While our study is limited to MF, the most foundational RecSys model [162], the approach underlying LBD is easily applicable or extendible to more complex RecSys models. For the RecSys field, we hope our practical approach enables better insight into real-world recommendations, and stimulates future work into novel applications of uncertainty models for RecSys.

5.2 Related Work

Rating prediction is one of the core recommendation tasks in environments with significant amounts of explicit feedback [3, 97, 110, 112]. However, most widely-adopted models, such as MF only focus on point predictions [112, 137, 179, 180]. As a consequence, they are also unable to differentiate between cases where the prediction task is easy and the prediction is likely correct, and those where the error is likely to be higher.

Being able to quantify confidence in individual predictions has many potential uses. Showing confidence scores alongside recommendations is known to increase user satisfaction and platform trust [72, 135, 161]. Mesas and Bellogín [136] find that it may be possible to significantly increase precision by focusing only on the highest confidence predictions. Jeunen and Goethals [84] and Adomavicius et al. [3] use lower confidence bounds instead of estimated relevance values to improve policy evaluation and recommender performance. Conversely, upper confidence bounds can be used to recommend a more diverse range of items or to explore user preferences for items where they are poorly understood [22, 40].

Existing confidence estimation methods can be broadly divided into three categories: heuristic, bootstrapping and generative approaches. The algorithms in the first group extend point predictions with some predefined measures, for instance, the number of interactions for the user or the item [3, 14, 134]. However, whilst computationally light, the use of heuristics may also lead to poor generalization [134, 227]. Mazurowski [134] therefore proposes a bootstrapping approach that trains a recommender multiple times using different slices of the training data. Uncertainty in a given rating is then represented by the variability of the prediction across different models. However, training multiple models also means that this technique has a significant computational cost.

Generative methods instead view individual ratings as samples from a distribution, whose parameters have to be learned. In practice, this distribution is generally assumed to be Gaussian with the same variance parameter for all ratings [120, 137, 169, 205, 233]. However, further steps need to be

taken to capture rating variability, as simple generative methods such as Probabilistic Matrix Factorization (PMF) have the same confidence for all ratings [137]. Bayesian Probabilistic Matrix Factorization (BPMF) [169] extends PMF by integrating over all possible user and item vectors via Gibbs sampling [35]. The empirical sampling variance of the embeddings then represents the model’s confidence in the correctness of the learned parameters. However, this method of sampling is also expensive and the model thus difficult to apply in practice. Lim and Teh [120] instead use variational methods for quicker convergence, albeit at the cost of accuracy and a worse fit to the data [169].

Our approach is conceptually most similar to the following two approaches: Confidence-Aware Matrix Factorization (CMF) of Wang et al. [205] and OrdRec of Koren and Sill [111]. The former extends PMF by relaxing its assumption of shared variance. Instead, each rating is assumed to be distributed with its own variance that is a product of learned global, user and item terms. The authors also propose an analogous extension to BPMF; however, the computational complexity arising from sampling still remains. On the other hand, OrdRec [111] assumes a fixed-variance logistic distribution to model preference, but also learns a set of varying thresholds for each user. The probability density falling between a pair of neighboring thresholds is then mapped to a single rating value and represents the probability of observing that rating. This allows variance to vary per user, but does so in an oblique and hard to interpret manner. We differ from existing work by relying on beta distributions to capture uncertainty, as we argue that this is a more appropriate choice than Gaussian or logistic distributions to reflect the variation in user ratings.

5.3 Background: The Beta Distribution

The beta distribution is a continuous probability distribution over the closed interval $[0, 1]$ [92]. Because it can easily be extended to any finite interval and can represent a great variety of distributions, the beta distribution has been widely applied to many different settings [68]. Despite this wide applicability, the distribution only has two parameters $\alpha > 0$ and $\beta > 0$. Its probability density function (PDF) is defined as:

$$f_B(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}, \quad (5.1)$$

where the normalization factor, $B(\alpha, \beta) = \int_0^1 f_B(t; \alpha, \beta) dt$, is the complete regularized beta function that ensures that the PDF integrates to one. The

mean of the distribution (μ) is determined by the ratio of α and the sum $\alpha + \beta$, where an α higher than the β means that the mean is closer to 1 than to 0, and vice versa. The variance of the distribution depends on both the sum of the parameters as well as their product:

$$\mathbb{E}[x] = \mu = \frac{\alpha}{\alpha + \beta}, \quad \mathbb{V}[x] = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}. \quad (5.2)$$

We note that generally the beta distribution is non-symmetric, except when $\alpha = \beta$. For instance, when $\alpha = \beta = 1$ it becomes equal to the (symmetric) uniform distribution.

The beta distribution models the likelihood function of the parameter underlying a Bernoulli trial, where $\alpha - 1$ is the number of successes and $\beta - 1$ the number of failures. For an intuitive understanding, imagine $\alpha + \beta - 2$ flips of a biased coin where $\alpha - 1$ heads and $\beta - 1$ tails are observed. If the coin had a probability x of heads, the probability of the above result would be proportional to $x^{\alpha-1}(1-x)^{\beta-1}$. We see that the PDF of the beta distribution is simply this probability for each x value, but normalized to produce a valid distribution.

Inspired by this intuition, one can alternatively parameterize the beta distribution by its mean $\mu \in (0, 1)$ and the sample size $\nu \in (0, \infty)$. This corresponds to $\alpha = \nu\mu$ and $\beta = \nu(1 - \mu)$ in the original parameterization, since $\mu = \alpha/(\alpha + \beta)$ and $\nu = \alpha + \beta$. Intuitively for our likelihood example, μ would correspond to the observed mean and ν reflects the number of observations of the Bernoulli trial: thereby, ν can also be seen as a measure of confidence.

The cumulative distribution function (CDF) of the beta distribution is not as straightforward:

$$F_B(x; \alpha, \beta) = I_x(\alpha, \beta) = \frac{\int_0^x f_B(t)dt}{B(\alpha, \beta)}, \quad (5.3)$$

where $I_x(\alpha, \beta)$ is the incomplete regularized beta function. Whilst no closed form for the CDF exists, it can be represented in terms of infinite power series or continued fractions [49, p. 292, 385-389]. Using these representations, close approximations of the CDF and its derivative w.r.t. α and β can be computed efficiently [166, 199] and implementations of these approximations are already available in popular software packages such as Tensorflow [1] and Jax [23].

Finally, while the beta distribution only covers the closed interval $[0, 1]$, it can easily be used to create a distribution over any closed interval. Let $R^{\min}$ and $R^{\max}$ indicate the endpoints of a closed interval for the random variable

$x' \in [R^{\min}, R^{\max}]$. We can transform the beta distribution to cover this interval by taking from it a sample $x \sim \text{beta}(\alpha, \beta)$ and transforming it: $x' = x(R^{\max} - R^{\min}) + R^{\min}$. Thus for each point x' in the interval $[R^{\min}, R^{\max}]$, there is a corresponding x in $[0, 1]$: $x = \frac{x' - R^{\min}}{R^{\max} - R^{\min}}$. As a result, the PDF of the new distribution is simply: $f'_B(x'; \alpha, \beta) = f_B(x; \alpha, \beta) / (R^{\max} - R^{\min})$, and its CDF is: $F'_B(x'; \alpha, \beta) = F_B(x; \alpha, \beta)$. The mean value undergoes the same transformation: $\mathbb{E}[x'] = \mathbb{E}[x](R^{\max} - R^{\min}) + R^{\min}$, and the variance is multiplied by the size of the interval squared: $\mathbb{V}[x'] = \mathbb{V}[x](R^{\max} - R^{\min})^2$. Thereby, the beta distribution is not only useful to model ratios or probabilities, but also for modeling other values that are bound to other closed intervals.

5.4 Method: Learned Beta Distributions for Rating Prediction

Our goal is to introduce a rating prediction model that quantifies the confidence it has in its predictions without heavy computational costs. In this section, we propose learned beta distributions as an expressive and conceptually simple method for rating prediction with an elegant measure of confidence. The remainder of this section is structured as follows: First, we detail how a beta distribution can be transformed into an optimizable distribution over discrete ratings (Section 5.4.1). Second, several approaches are proposed to model the parameters of a unique rating distribution for each user-item combination (Section 5.4.2). Third, two alternatives to modeling user and item biases into the parameters of the model are introduced (Section 5.4.3). Fourth, static and adaptive binning strategies for discretization are proposed (Section 5.4.4). Fifth, a summary of the complete approach is laid out, starting from user and item embeddings and ending in discrete rating distributions (Section 5.4.5). Finally, we compare LBD with the existing MF, CMF and OrdRec approaches in conceptual terms (Section 5.4.6).

5.4.1 From a beta distribution to an optimizable discrete rating distribution

In this subsection, we describe how a beta distribution can be transformed to model a discrete rating distribution, and subsequently optimized with a cross-entropy loss.

We assume that users can provide ratings on a discrete and linear scale: $R \in \{R_1, R_2, \dots, R_n\}$, where n is the number of possible rating values and the difference between R_1 and R_2 is equal to that between R_2 and R_3 , etc. To denote the minimum and maximum values, we use $R_1 = R^{\min}$ and $R_n = R^{\max}$, respectively.

Let α_{ij} and β_{ij} be the parameters for the distribution of rating R_{ij} , corresponding with user i and item j . We begin by transforming the associated beta distribution to the interval spanning the range of possible rating values: $[R^{\min}, R^{\max}]$. As discussed in Section 5.3, a sample from a beta distribution: $x \sim \text{beta}(\alpha_{ij}, \beta_{ij})$ can be transformed accordingly: $x' = x(R^{\max} - R^{\min}) + R^{\min}$. The resulting distribution over x' has the same shape as the beta distribution, but is transposed and stretched over the rating interval. The PDF is easily mapped back to the original distribution: $f'_B(x'; \alpha, \beta) = f_B(x; \alpha, \beta) / (R^{\max} - R^{\min})$ and the same goes for the CDF: $F'_B(x'; \alpha, \beta) = F_B(x; \alpha, \beta)$.

While our new distribution is on the correct interval, it still has to be discretized. This can be done straightforwardly by splitting the interval into bins, each representing the probability mass of one rating value. Let the $W_{ij,r} \in [0, 1]$ parameter represent the bin width for rating r s.t. $\sum_{r=1}^n W_{ij,r} = 1$. The lower edge of rating R_r can then be defined for $r > 1$ as $E_r = E_{r-1} + (R^{\max} - R^{\min})W_{ij,r-1}$, and $E_1 = R^{\min}$ for the first edge. The probability of a rating is then simply the area under the PDF and inside the corresponding bin. We note that an equivalent discretization can be achieved by binning the original beta distribution, the bin edges here are simply transformed inversely: $e_r = (E_r - R^{\min}) / (R^{\max} - R^{\min})$. Therefore, it is also straightforward to express the probabilities in terms of the CDF of the underlying beta distribution:

$$\begin{aligned} P(R_r; \alpha_{ij}, \beta_{ij}) &= F'_B(E_{r+1}; \alpha_{ij}, \beta_{ij}) - F'_B(E_r; \alpha_{ij}, \beta_{ij}) \\ &= F_B(e_{r+1}; \alpha_{ij}, \beta_{ij}) - F_B(e_r; \alpha_{ij}, \beta_{ij}). \end{aligned} \quad (5.4)$$

Importantly, our transformation from the beta CDF (F_B) to the rating probability $P(R_r; \alpha_{ij}, \beta_{ij})$ is completely differentiable. Furthermore, the gradients w.r.t. α_{ij} and β_{ij} can be computed using the close approximations of previous work [166, 199]. In other words, our discrete rating probability is fully differentiable to its input parameters. As a result, we can easily use it to optimize a cross-entropy loss:

$$\mathcal{L}(\alpha, \beta) = \frac{1}{Z} \sum_i \sum_j \log P(R_{ij}; \alpha_{ij}, \beta_{ij}). \quad (5.5)$$

Thus, we have shown that beta distributions can be transformed into discrete rating distributions in a straightforward and elegant manner. Since

the result is fully differentiable, it is easily optimizable; in particular, we propose to maximize the log-likelihood of observed rating data by minimizing a cross-entropy loss.

5.4.2 Modeling means and confidences

We have introduced a method for producing a discrete rating distribution for a rating R_{ij} from parameters α_{ij} , β_{ij} and W_{ij} . In this subsection, we propose a simple way of modeling α_{ij} and β_{ij} based on the interactions between user and item embeddings.

To start, we make use of the alternative parameterization described in Section 5.3 and introduce the mean $\mu_{ij} \in (0, 1)$ and confidence $\nu_{ij} \in (0, \infty)$. These map back to the original parameterization through: $\alpha_{ij} = \mu_{ij}\nu_{ij}$ and $\beta_{ij} = (1 - \mu_{ij})\nu_{ij}$. Furthermore, inspired by the simplicity of MF, we introduce a non-zero learned embedding for each user $U_i \in \mathbb{R}^D$ and for each item $V_j \in \mathbb{R}^D$. Our first insight is that because $\mu_{ij} \in (0, 1)$, it can be modeled elegantly with the cosine similarity:

$$\mu_{ij} = \frac{1}{2} + \frac{1}{2} \text{Cosine-similarity}(U_i, V_j) = \frac{1}{2} + \frac{1}{2} \frac{U_i^\top V_j}{\|U_i\| \|V_j\|}. \quad (5.6)$$

Thereby, a higher similarity between the directions of the embeddings results in a higher mean of the predicted rating distribution. Our second insight is that any differentiable transformation from the embeddings to a positive scalar can be used for the confidence parameter. Whilst many transformations are possible, we propose the following three:

$$\nu_{ij}^{\text{norm}} = \|U_i\| \|V_j\|, \quad \nu_{ij}^{\text{sum}} = \|U_i + V_j\|, \quad \nu_{ij}^{\text{dot}} = |U_i^\top V_j|. \quad (5.7)$$

With ν_{ij}^{norm} , the confidence parameter is equal to the product of the L_2 norms of the embeddings. Thus users or items with larger embeddings will result in more confident predictions. Alternatively, one could interpret this as each user and item having a unique confidence term, expressed in the distance of their embedding vector from the origin. The simplicity of ν_{ij}^{norm} makes it easy to understand, but it therefore also lacks expressiveness and cannot capture complex patterns that vary over specific user-item interactions.

The L_2 norm of the summation of the embeddings is used to model confidence in ν_{ij}^{sum} . Here both the direction and the distance of the embeddings determine the resulting confidence. As a result, this model can capture more complex interactions between users and items, but is also intuitively more difficult to understand.

Lastly, ν_{ij}^{dot} uses the absolute value of the dot product of the embeddings. Thus, confidence is also determined by both the direction and the distance of the embedding vectors. However, because the absolute value is used, both high similarity and dissimilarity result in high confidence predictions. The dot product highly correlates with the cosine similarity, which could also mean there is a high correlation between μ_{ij} and ν_{ij}^{dot} . It is unclear whether such a correlation is desirable in practice.

To summarize, we have introduced a novel approach to modeling a beta distribution by basing its input parameters on the interaction between user and item embeddings. In particular, we propose to model mean ratings by cosine similarity and the confidence by one of several possible transformations. Thereby, our approach has the same model-complexity as MF, since both only use user and item embeddings. But whereas MF only uses them to model pointwise ratings, LBD models predictions as probability distributions.

5.4.3 Modeling user and item biases

An important feature of standard MF models is that they can also include bias weights per user and item [91, 109, 112]. These bias weights can increase all the ratings of a user or an item, for instance, to capture that certain users tend to give higher ratings than the average user. Because adding these weights is very effective for MF, we propose adding them to our LBD model as well.

The addition of bias weights to the beta distribution is straightforward. We introduce the optimizable scalars $a \in \mathbb{R}$ and $b \in \mathbb{R}$, and denote a_0 for a global weight, a_i for a user weight and a_j for an item weight, and the analogous weights b_0 , b_i and b_j . These bias weights are then simply added to the weights predicted with the embeddings:

$$\alpha'_{ij} = \max(a_0 + a_i + a_j + \alpha_{ij}, \epsilon), \quad \beta'_{ij} = \max(b_0 + b_i + b_j + \beta_{ij}, \epsilon), \quad (5.8)$$

where ϵ is a very small positive quantity to ensure that $\alpha'_{ij} > 0$ and $\beta'_{ij} > 0$.

While the addition may appear simple, it also has a Bayesian interpretation, as the normalized product of beta distributions can also be produced by summing their parameters accordingly:

$$\begin{aligned} f_B(a_0 + 1, b_0 + 1) f_B(a_i + 1, b_i + 1) f_B(a_j + 1, b_j + 1) f_B(\alpha_{ij}, \beta_{ij}) \\ \propto f_B(a_0 + a_i + a_j + \alpha_{ij}, b_0 + b_i + b_j + \beta_{ij}). \end{aligned} \quad (5.9)$$

Therefore, we can interpret $f_B(a_0 + 1, b_0 + 1)$ as a global prior distribution, $f_B(a_i + 1, b_i + 1)$ as the likelihood function given that user i is involved,

the same for $f_B(a_j + 1, b_j + 1)$ and item j , and finally, $f_B(\alpha_{ij}, \beta_{ij})$ as the likelihood function for the interaction between i and j . While this Bayesian interpretation is valid, we note that our optimization only considers the final distribution: therefore, it does not directly optimize the accuracy of the prior and likelihood decomposition.

Additionally, we also propose bias weights in terms of the alternative parameterization. We introduce the optimizable scalars $u \in (0, 1)$ and $v \in (0, \infty)$, where u_0 and v_0 represent global weights and u_i , u_j , v_i and v_j user and item specific weights. The weights v are used as multipliers on the confidences ν : thus higher v result in more confident predictions, and vice versa. For the u weights, we create a piecewise linear mapping that places $\mu'_{ij} = 0.5$ when $\mu_{ij} = u_0 u_i u_j$: thus high u values result in a lower predicted mean, and vice versa. Formally, the bias weights are applied as follows:

$$\nu'_{ij} = v_0 v_i v_j \nu_{ij}, \quad \mu'_{ij} = \begin{cases} \frac{\mu_{ij}}{2 u_0 u_i u_j} & \text{if } \mu_{ij} \geq u_0 u_i u_j, \\ \frac{1}{2} + \frac{\mu_{ij} - u_0 u_i u_j}{2(1 - u_0 u_i u_j)} & \text{otherwise.} \end{cases} \quad (5.10)$$

Since our proposed applications of bias weights are fully differentiable, they effortlessly integrate into our LBD model.

5.4.4 Modeling static and adaptive discretization strategies

We have described how user and item embeddings and biases can model the parameters of a beta distribution. As discussed in Section 5.4.1, LBD creates a discrete rating distribution through binning according to the bin width parameters W . We propose a static binning strategy for modeling W : LBD-S; and an adaptive binning strategy: LBD-A.

The static strategy of LBD-S straightforwardly makes each bin equisized: $\forall i, j, r : W_{ij,r} = \frac{1}{n}$. In terms of model-complexity, this is the most efficient choice as none of the W parameters have to be learned. Implicitly this strategy assumes a linear relation between preference and ratings, i.e., the uniform distribution with $\alpha = 1$ and $\beta = 1$ will translate to a uniform discrete distribution. However, this assumption may not be correct, e.g., in the Movielens dataset [71], the whole ratings ($\{1, 2, 3, 4, 5\}$) are much more frequent than the half ratings ($\{0.5, 1.5, 2.5, 3.5, 4.5\}$). This difference in frequency seems to be an artifact of the rating interface, which leads users to choose whole-number ratings over the others, and is unlikely to reflect an actual clustering in the underlying preferences [86, 152]. Importantly, such

patterns are difficult to capture for LBD-S, since it assumes that probability mass is smoothly spread over all rating values.

In contrast, the adaptive binning strategy varies the bin widths per rating, user and item, and thereby, it can better capture frequency differences between certain rating values. Inspired by the discretization of Koren and Sill [111], we introduce the unnormalized bin widths $w_{ij,r} = \exp(\theta_i^r + \theta_j^r)$ with the optimizable parameters θ_i^r and θ_j^r for each user i , item j and rating value r . The bin widths are then simply obtained by normalizing: $W_{ij,r} = w_{ij,r} / \sum_{r'=1}^n w_{ij,r'}$. By increasing the bin width of certain rating values relative to others, these rating values can become more probable without changing the underlying beta distribution. Thereby, LBD-A can capture non-smooth patterns that LBD-S cannot, e.g., that half ratings are less probable than whole ratings, or whether a certain user is more likely to translate a specific level of preference to a higher rating than most other users [86]. Importantly, the increased expressiveness of LBD-A over LBD-S comes at the cost of only a few additional parameters for each user and item.

5.4.5 Overview of the LBD method

Now that all the components of the learned beta distributions (LBD) method have been introduced, we summarize how these components come together to form a single approach. The visualization in Figure 5.1 accompanies this subsection.

Similar to MF, LBD has an optimizable embedding vector for each user and item, the cosine similarity between these vectors determines the mean prediction (μ), and a positive function is used to get a positive scalar indicating the confidence in the prediction (ν). The mean and confidence are then mapped to α and β values and bias terms are also added. The resulting values serve as input parameters to a beta distribution. This beta distribution is the prediction of user preferences with explicit uncertainty; preferences are modeled as ratios in the interval $[0, 1]$ and the beta PDF (f_B) indicates how probable each possible value is predicted to be. This distribution is then transformed to match the range of possible rating values and discretized to give exact probabilities per rating. Discretization can be done either using equally sized bins or by adjusting each region's width $w_{ij,r}$ based on the user and item terms (θ). Because each step between the embeddings and the final discrete rating distribution is differentiable, one can compute the gradient of a rating probability w.r.t. the underlying embeddings. We utilize this differentiability to optimize the model with

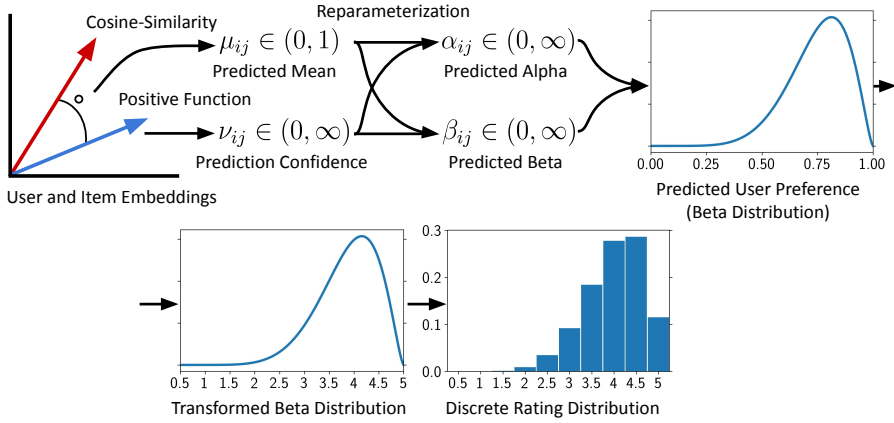


Figure 5.1: Visualization of the learned beta distributions (LBD) method. LBD starts with user and item embeddings, their cosine similarity determines the predicted mean and a positive function its confidence. These values are reparameterized to the input of a beta distribution, which models a prediction of user preference with explicit uncertainty. This distribution is transformed and discretized to produce the predicted discrete rating distribution. The entire LBD method is differentiable and optimized with a cross-entropy loss.

a cross-entropy loss, so that the predicted discrete rating probabilities maximize the likelihood of observed rating data.

Thereby, our LBD method predicts user ratings with explicit uncertainty in its predictions. Importantly, it does this with the same model-complexity as MF, and unlike previous methods that utilize Gaussian distributions [169, 205], LBD can have non-symmetric variance that can capture user-item interactions, never predicts ratings outside of the valid rating range, can be described using just two parameters and does not involve heavy computational costs in its optimization and prediction.

5.4.6 Comparison with existing methods

Before LBD is compared with the existing MF [112], CMF [205] and OrdRec [111] methods in experimental terms in Section 5.6, we first consider key conceptual differences between the methods. To accompany the discussion, Figure 5.2 visualizes the predictions of MF, CMF and LBD for an example scenario with one user and three items.

To start, MF differs from CMF, OrdRec and LBD in that its pointwise predictions are without any sense of confidence, and accordingly, they are presented as points on the x-axis. As a result, while CMF and LBD show

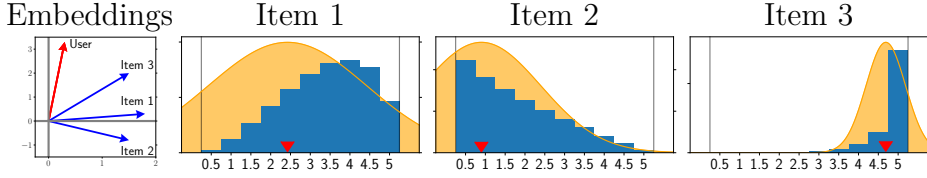


Figure 5.2: Example comparison of how MF, CMF and LBD make predictions for a user and three different items. The prediction of MF is pointwise (red triangle); that of CMF is a Gaussian distribution (orange continuous probability density function); and that of LBD is a discrete rating prediction based on an underlying beta distribution (blue discrete probability mass function).

high uncertainty about their predictions for item 1 and 2, MF gives no indication that it may be less reliable there.

Next, we compare CMF and LBD, as both give predictions in the form of a distribution: CMF as a Gaussian distribution and LBD as a discrete probability distribution based on an underlying beta distribution. We argue that the usage of a beta distribution by LBD provides three key advantages over Gaussians for rating prediction: (i) Beta distributions can be bounded to only give probability inside the possible range of ratings, whereas Gaussians are unbounded. This problem is illustrated across all items in Figure 5.2 as the PDF of CMF gives much probability outside the valid range of ratings. (ii) Beta distributions can have non-symmetric variance, while Gaussians cannot. For instance, LBD thinks the lowest rating of item 2 is most probable, and puts the remainder of its probability on the higher ratings. In contrast, CMF also thinks a low rating is most likely, but has to give equal probability to ratings below and above it, including ratings out of the valid range. (iii) Lastly, LBD can be fully parameterized with minimal model-complexity, since it only requires an interaction between two embeddings to compute all its parameters. Conversely, this is not possible for a Gaussian because its mean is unbounded, and as a result, CMF needs additional parameters for its confidence model (not visualized in Figure 5.2).

Finally, our visualization in Figure 5.2 does not include OrdRec because its differentiating characteristics are difficult to display. OrdRec uses the logistic distribution, which is very similar to the Gaussian distribution but with heavier tails. However, unlike CMF and similar to LBD, it also applies discretization through the same adaptive binning strategy as described in Section 5.4.4. Thus unlike CMF, OrdRec can transform the shape of its rating distribution through adaptive binning, enabling it to have asymmetric variance. Nevertheless, the adaptive binning strategy is relatively limited in its expressiveness (cf. $w_{ij,r}$ and ν_{ij}). Furthermore, it is also unclear whether

the symmetric logistic function, with its infinite coverage and heavy tails, is appropriate to model user preferences. In contrast, LBD captures both a measure of preference and confidence within its embeddings, and can already express asymmetric preference distributions before discretization is applied. The remainder of this paper will investigate whether the theoretical advantages of LBD can be verified experimentally and lead to more effective modeling of uncertainty.

5.5 Experimental Setup

The experiments carried out for this paper are aimed at answering the following four (chapter) research questions:

- RQ5.1** What modeling of the parameters of LBD results in the highest predictive performance for rating prediction?
- RQ5.2** Does our LBD approach provide rating prediction performance that is competitive with MF, CMF and OrdRec?
- RQ5.3** Is there a stronger correlation between the confidence and accuracy of predictions by LBD than for CMF and OrdRec?
- RQ5.4** Does the confidence modeling of LBD translate to higher performance compared to MF, CMF and OrdRec in a high-precision targeted recommendation task?

Correspondingly, the goals behind these research question are: (i) to find out what version of LBD is the most accurate at rating prediction; (ii) to evaluate whether the confidence modeling of LBD sacrifices competitive accuracy; (iii) to verify whether LBD has a better indication of confidence than the existing confidence-aware methods; and (iv) to investigate how the confidence modeling of LBD could be beneficially leveraged in a practical downstream task.

5.5.1 Dataset

We base our experiments on the MovieLens 10M dataset [71], where ratings from 69878 users are spread over 10677 items on a 0.5 point scale between 0.5 and 5. In order to perform cross-validation, we randomly split the user ratings into 10 folds. Each experiment is repeated 10 times, so that each fold is used for evaluation once, while the remaining 9 folds are further split into 95% train and 5% validation data. To avoid irrelevant cold-start

problems in evaluation, we further ensure that the evaluation set never contains ratings involving users or items that are not in the training set.

5.5.2 Baselines, metrics and parameter tuning

We compare our LBD method with three baselines of similar model-complexity: MF [112, 137], CMF [205] and OrdRec [111]. MF is a foundational RecSys method and remains a very strong rating prediction baseline when tuned well [162, 163]. Furthermore, it does not have any confidence modeling, and thereby, allows us to evaluate whether the confidence modeling of LBD prevents it from providing competitive rating prediction accuracy. CMF performs confidence modeling through Gaussian distributions with varying variance. Lastly, OrdRec models confidence by partitioning a logistic distribution into non-overlapping regions, each representing the probability mass assigned to a possible rating value.

For the comparison with the baselines, we use the two best LBD parameterizations identified for **RQ5.1**. Both use α and β bias terms with embeddings of size 512 to jointly model preference and confidence, where the latter is represented via ν^{sum} . We consider both static binning (LBD-S) and adaptive binning (LBD-A).

To evaluate various aspects of recommendation performance, we use a variety of metrics for regression (RMSE, MAE), classification (accuracy, average log-likelihood) and ranking (NDCG@3, NDCG@10). For classification, we count a predicted rating to be correct if it is closer to the true rating than to any other value in $\{0.5, 1, \dots, 5\}$. For the ranking task, we use models' predicted ratings to rerank each user's test set items with known ratings, taking the average over all users. Item relevance is *not* binarized as we aim to achieve the best ordering across all ranks. For regression and ranking, predictions are made using each model's estimate of the rating mean, whilst accuracy is evaluated using the mode of the distribution. Although different point estimates, e.g., quantiles, may be useful for other tasks, we observed that the above estimates resulted in the highest performance for all models.

The hyperparameters of each model were determined by an extensive random search. All models were trained with embedding sizes: 32, 64, 128, 256 and 512, for up to 50 epochs with minibatch size of 8192 using the Adam optimizer [102] with learning rates in the range $[10^{-6}, 10^{-2}]$ and early stopping on validation RMSE with a patience of 10 and a $5 \cdot 10^{-4}$ tolerance. We also applied L_2 regularization on either all embeddings equally or proportional to their frequency in the training data. The final parameters per model were chosen according to the best achieved RMSE on the validation set of the first data fold. MF's embeddings were used to

initialize CMF [205]. Furthermore, [111] only evaluate a version of their model whose adaptive binning is determined solely on the user (OrdRec-U). We also include OrdRec with binning based on both the user and item via the exponent from Section 5.4.4 (OrdRec-UI). And while the authors allow for any model to represent the logistic mean, we use OrdRec on top of MF to allow for a direct comparison with other confidence models.

To evaluate the effectiveness of the confidence modeling, we measure the correlation between confidence and prediction error using both linear correlation (Pearson’s r) as well as rank correlation (Kendall’s τ). The former reflects the association between specific error and confidence values, whereas the latter quantifies the effectiveness of using model confidence to separate low and high-error predictions. Furthermore, we evaluate the relationship qualitatively by clustering interactions into 1000 bins based on the predicted variance of individual ratings. We use rating variance as it is a confidence measure available for all models, and group interactions using equally spaced bins to identify the relationship between accuracy and specific confidence values. However, due to this grouping strategy, we may also observe patterns for a small number of interactions spread over many bins that do not generalize to the data as a whole. Therefore, we also group interactions into bins containing the same number of interactions of monotonically increasing variance, allowing us to evaluate the general direction of the association but not for specific confidence values.

5.5.3 High-precision targeted recommendation

Finally, inspired by work on prediction confidence in other domains [72, 134, 136], we want to evaluate whether the confidence modeling of LBD can be effectively exploited in downstream tasks.

We simulate a high-precision task where only a limited number of targeted users are provided with recommendations. Specifically, each method is only allowed to choose N different users, for each of whom it makes one personalized item recommendation, while the remaining users receive no recommendations. For evaluation purposes, the methods may only choose user-item combinations that are present in the test set, so that we can evaluate performance based on the number of successful recommendations, which are taken to be those with rating of at least 4.5. For each model, we report the proportion of users who received a successful recommendation (Precision@1) over different values of N , as well as the relative improvement over the confidence-unaware MF.

This task is analogous to a notification setting where recommendations are mildly intrusive interruptions. For the best user experience in this

setting, only highly relevant recommendations should be made and the number of recommendations per user should be limited. We expect that confidence modeling allows for a higher precision in selecting highly relevant items, since it can better identify which high rating predictions are most likely correct.

5.6 Results and Discussion

5.6.1 Parameterization of the LBD model

We begin with addressing the first research question (**RQ5.1**) by investigating *what method of modeling the LBD parameters results in the highest predictive performance*. To that end, we consider Table 5.1 which shows the average performance of various LBD configurations over 10 test folds. The configurations are combinations of different confidence functions and (α, β) and (ν, μ) biases proposed in Sections 5.4.2 and 5.4.3. In addition, we also evaluate a model with no bias terms, a model that uses separate embeddings to model preference and confidence (denoted with 256, 256) and LBD-A with adaptively discretized bins (denoted by 512 + 10, for the ten additional learned binning parameters).

We see that the choice of parameterization can have a substantial impact on performance: overall, using ν^{sum} with α, β biases alongside adaptive bins leads to best performance, while using ν^{dot} is clearly the worst choice. Interestingly, the above model with adaptive bins shows the best performance across all non-ranking metrics, as well as second best performance on ranking, with particularly salient improvements in terms of accuracy and log-likelihood. Moreover, the same model but without dynamic bins achieves the best performance for ranking metrics as well as the second highest performance on regression metrics and log-likelihood. This thus suggests that these two parameterizations are best suited to rating prediction, regardless of the specific task.

Based on the observed performance differences, it appears that the adaptive binning allows the model to better capture the rating distribution, as evidenced by the associated size of the improvement in the fit to the rating data. We also hypothesize that ν^{sum} is better at confidence modeling than ν^{norm} due to its ability to capture interactions between specific users and items. On the other hand, we speculate that the dot product performs the worst due to its high correlation with cosine similarity, which could in turn produce a high correlation between predicted means and corresponding confidences. Furthermore, our results also indicate that using separate

Table 5.1: Performance in terms of regression, classification and ranking metrics for various LBD configurations (see Sections 5.4.2-5.4.4). Reported results are averages from 10-fold cross-validation, with standard deviations in parentheses. Two embedding sizes denote separate embeddings for preference and confidence. 512+10 denotes LBD with the adaptive binning strategy applied (LBD-A). Highest performance per metric is denoted in bold.

Size	$\nu^{(\cdot)}$	Bias	RMSE	MAE	Accuracy
512	<i>sum</i>	-	0.7932 (0.0005)	0.6130 (0.0004)	0.3016 (0.0005)
512	<i>sum</i>	μ, ν	0.7864 (0.0005)	0.6058 (0.0003)	0.3092 (0.0003)
512	<i>sum</i>	α, β	0.7761 (0.0005)	0.5895 (0.0005)	0.3139 (0.0003)
512	<i>norm</i>	α, β	0.7852 (0.0007)	0.5942 (0.0005)	0.3140 (0.0005)
512	<i>dot</i>	α, β	0.8387 (0.0005)	0.6391 (0.0006)	0.2972 (0.0005)
256, 256	<i>sum</i>	α, β	0.7850 (0.0006)	0.5964 (0.0006)	0.3133 (0.0006)
512 + 10	<i>sum</i>	α, β	0.7759 (0.0004)	0.5875 (0.0004)	0.4356 (0.0006)
Size	$\nu^{(\cdot)}$	Bias	Ave. Log-L.	NDCG@3	NDCG@10
512	<i>sum</i>	-	-1.769 (0.001)	0.9330 (0.0004)	0.9568 (0.0002)
512	<i>sum</i>	μ, ν	-1.760 (0.001)	0.9339 (0.0003)	0.9568 (0.0001)
512	<i>sum</i>	α, β	-1.755 (0.003)	0.9357 (0.0003)	0.9583 (0.0001)
512	<i>norm</i>	α, β	-1.818 (0.004)	0.9336 (0.0002)	0.9572 (0.0002)
512	<i>dot</i>	α, β	-1.865 (0.005)	0.9330 (0.0004)	0.9495 (0.0001)
256, 256	<i>sum</i>	α, β	-1.774 (0.002)	0.9331 (0.0003)	0.9570 (0.0001)
512 + 10	<i>sum</i>	α, β	-1.432 (0.002)	0.9351 (0.0003)	0.9579 (0.0001)

embeddings to model means and confidences does not boost performance, suggesting that a single set of user and item embeddings are enough to capture both preference and confidence patterns. Lastly, it is also clear that including bias terms can lead to better performance across all metrics, as expected from their popularity for MF-based models. However, we are unsure why bias weights in α, β terms are more effective than in μ, ν terms.

In conclusion, to answer **RQ5.1**: we find that the parameterization of LBD can have a substantial effect on its performance. *Our results indicate that the confidence function ν^{sum} with α and β bias terms, both with and without dynamic binning, lead to the highest overall performance across all our metrics.*

5.6.2 Recommendation performance

Now that we have established the best configurations of the LBD model, we can address our second research question (**RQ5.2**) and consider *whether*

Table 5.2: Model performance on regression, classification and ranking tasks. Results are averages from 10-fold cross-validation, with standard deviations in parentheses. Highest performance per metric is denoted in bold. Significant improvement of an LBD model over *all* baseline models is denoted by Δ , significant improvement of a baseline over *any* LBD version is denoted by ∇ (separate one-sided Wilcoxon signed-rank tests with matched folds, $p < 0.001$).

Model	RMSE	MAE	Accuracy
MF (128)	0.7802 (0.0006)	0.5984 (0.0005)	0.2923 (0.0005)
MF (512)	0.7883 (0.0006)	0.6022 (0.0005)	0.2934 (0.0004)
CMF (128)	0.7760 (0.0007)	0.5936 (0.0004)	0.3226 (0.0004) ∇
CMF (512)	0.7820 (0.0005)	0.5967 (0.0003)	0.2849 (0.0015)
OrdRec-U (512)	0.7821 (0.0006)	0.6043 (0.0004)	0.2322 (0.0007)
OrdRec-UI (512)	0.7765 (0.0006)	0.5896 (0.0005)	0.4187 (0.0006) ∇
LBD-S (512)	0.7761 (0.0005)	0.5895 (0.0005)	0.3139 (0.0003)
LBD-A (512)	0.7759 (0.0004)	0.5875 (0.0004) Δ	0.4356 (0.0006) Δ
Model	Ave. Log-L.	NDCG@3	NDCG@10
MF (128)	-2.004 (0.002)	0.9334 (0.0002)	0.9570 (0.0001)
MF (512)	-2.022 (0.002)	0.9321 (0.0002)	0.9560 (0.0001)
CMF (128)	-1.777 (0.001)	0.9338 (0.0002)	0.9573 (0.0001)
CMF (512)	-1.801 (0.001)	0.9329 (0.0003)	0.9567 (0.0001)
OrdRec-U (512)	-1.881 (0.001)	0.9343 (0.0003)	0.9575 (0.0001)
OrdRec-UI (512)	-1.569 (0.001) ∇	0.9349 (0.0003)	0.9578 (0.0001)
LBD-S (512)	-1.755 (0.003)	0.9357 (0.0003) Δ	0.9583 (0.0001) Δ
LBD-A (512)	-1.432 (0.002) Δ	0.9351 (0.0003)	0.9579 (0.0001)

the predictive performance of LBD is competitive with that of MF, CMF and OrdRec. For this question, we use the results in Table 5.2 which show the performance of all models with different embedding sizes over a variety of performance metrics.

Overall, we generally see no enormous variation in the performance of the models, as most are in a similar range. Whilst LBD-S achieves lower accuracy compared to CMF and OrdRec-UI, its regression performance is comparable to that of the above models, and its ranking performance is significantly better than for any other model. Furthermore, LBD-A, whilst showing a smaller improvement on NDCG, actually achieves the best performance across all other metrics, significantly improving over the baselines in MAE as well as both classification metrics. Importantly, the above results thus show that LBD has performance that is competitive with MF, CMF and OrdRec.

Table 5.3: Correlations between the predicted variance and MAE of baselines and LBD. Results are averages from 10-fold cross-validation, with standard deviations in parentheses. Highest correlation per type is denoted in bold. Significant improvement of LBD over *all* baselines is denoted by Δ (separate one-sided Wilcoxon signed-rank tests with matched folds, $p < 0.001$).

	CMF	OrdRec-U	OrdRec-UI	LBD-S	LBD-A
Pearson’s r (Linear Correlation)	0.079 (0.001)	0.157 (0.001)	0.175 (0.001)	0.304 (0.002) $^\Delta$	0.329 (0.002) $^\Delta$
Kendall’s τ (Rank Correlation)	0.043 (0.002)	0.094 (0.001)	0.112 (0.001)	0.194 (0.001) $^\Delta$	0.216 (0.001) $^\Delta$

As such, we answer **RQ5.2** in the affirmative: *the performance of LBD is competitive with MF, CMF and OrdRec across all our measured metrics*. Furthermore, it appears that LBD could even provide small significant improvements for most metrics. We note that our aim is not to improve these performance metrics, but to evaluate whether confidence modeling has a negative effect on predictive performance. These conclusions thus strongly imply that confidence modeling with LBD allows us to simultaneously model both preference and confidence without sacrificing predictive performance, at least not in terms of our regression, classification and ranking metrics.

5.6.3 Evaluating prediction confidence

Now that we have investigated the predictive performance of LBD, we turn our attention to its key feature —its confidence modeling— and address the third research question (**RQ5.3**) by evaluating *whether there is a stronger correlation between the confidence and accuracy of predictions for LBD than for CMF and OrdRec*. We note that no comparison with MF can be made, since its pointwise predictions do not include any indication of confidence, uncertainty or variance. Table 5.3 shows the strength of linear and rank correlation between mean absolute error (MAE) and predicted variance, representing the association between the prediction errors and confidence. We use predicted variance to quantify confidence, since low variance denotes a prediction with low uncertainty and thus high confidence, and vice versa.

From Table 5.3, it is immediately clear that there are substantial differences across models in terms of both types of correlation. Whilst OrdRec-UI’s correlation is the highest among the baselines, LBD-S and LBD-A demonstrate by far the strongest association between their predicted variance and the size of the error. In fact, compared to OrdRec-UI, both LBD models demonstrate a relative increase in both correlation coefficients of at least 70%, with LBD-A’s correlation being almost twice as strong as that

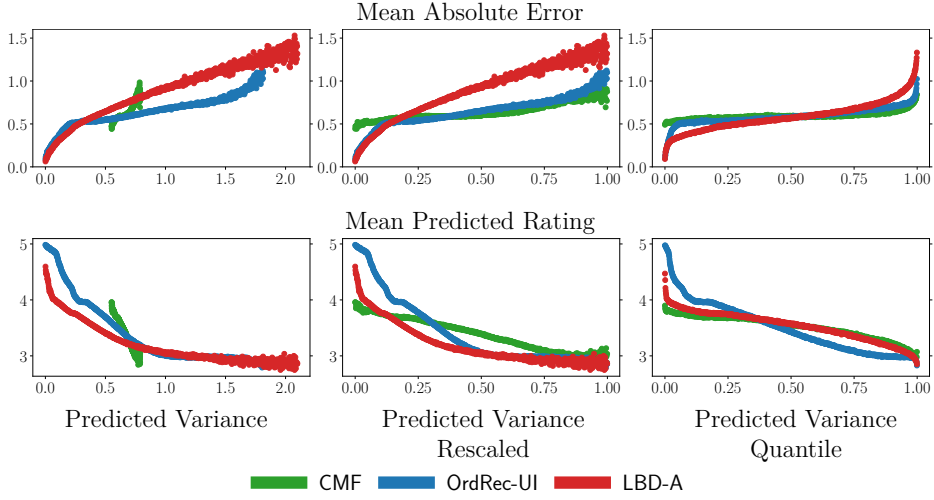


Figure 5.3: Variation of MAE (top) and average predicted rating values (bottom) for different values of predicted variance. Interactions grouped by the predicted variance of each model into 1000 clusters, discarding 0.1% of highest predicted variance outliers. Points represent mean response value of each cluster. Left: interactions grouped into equispaced bins spanning each model’s observed predicted variance. Center: grouping as on the left, but variance rescaled to $[0, 1]$. Right: each bin contains 0.1% of interactions, ordered by predicted variance.

of OrdRec. That LBD-S achieves such a large improvement over OrdRec-UI indicates that modeling user preferences with just a beta distribution and no adaptive binning better captures uncertainty in user ratings compared to the non-linear mapping of OrdRec. Overall, the above results thus suggest that LBD is considerably better at separating low-error predictions from those with a higher error.

For a more qualitative understanding of the above differences, Figure 5.3 shows how MAE and the average predicted rating vary over interactions grouped by predicted variance. Furthermore, as the predicted variance of distinct models lies over different ranges (see left column of Figure 5.3), we also show the correlation with predicted variance rescaled to the range $[0, 1]$ (center column), and variance quantiles (right column). We omit OrdRec-U and LBD-S from the visualization to reduce visual clutter.

We start by considering the top-left of Figure 5.3, which shows how the predicted variance of CMF, LBD-A and OrdRec-UI correlates with their accuracy, with higher variance corresponding to greater MAE for all. Rescaling each model’s variance for a more direct comparison, as done in the top-center of Figure 5.3, shows that there is a larger difference in terms

of MAE between the lowest and highest variance predictions of LBD-A and OrdRec-UI compared to CMF. This suggests that LBD-A and OrdRec-UI are more efficient at separating high and low-error predictions.

Interestingly, from top-center of Figure 5.3, one might also (albeit erroneously) conclude that, because MAE of OrdRec-UI and LBD-A is comparable when their predicted variance is low and otherwise generally lower for OrdRec-UI, that must mean that OrdRec-UI’s confidence modeling is better. We can see that this is however not true by inspecting the top-right of Figure 5.3, which presents the same results but with predictions ordered in quantiles of variance. There we see that using the predicted variance of LBD-A achieves a better separation between high and low MAE predictions when compared to OrdRec-UI (and CMF). OrdRec does appear to achieve a similar error to LBD-A on the interactions with the very lowest predicted variance, whilst also having the highest error in the right tail. However, the majority of the interactions, which are in the center, all have only minor differences in their MAE. LBD-A, on the other hand, whilst being particularly efficient in the tails, is also better at separating all of the other interactions. As such, for a comparable level of predictive performance (Section 5.6.2), LBD is better at predicting when it is more or less accurate, i.e., it has a substantially better model of confidence.

To get further insight into the different confidence models, the bottom row of Figure 5.3 also displays how the average predicted rating varies over the predicted variances. We see that all three models have more confidence in high value ratings, with this trend being the strongest for OrdRec. Potentially, this is an indication that it is easier to predict the positive preferences of a user than their neutral or negative preference. However, for the LBD model, we speculate that the usage of the ν^{sum} confidence function could also produce this trend. It assigns higher confidence when embeddings have a similar direction, and similar directions also result in higher ratings due to the usage of cosine similarity. Due to the scope of this work, we have to leave a further investigation into the implications of these trends for future work.

Altogether, we answer **RQ5.3** strongly in the affirmative: *LBD provides a significantly and considerably stronger correlation between the confidence and accuracy of predictions than CMF or OrdRec.* Consequently, LBD is substantially better at separating accurate predictions from those with moderate or high prediction errors.

5.6.4 High-precision targeted recommendation

Finally, in order to investigate whether the better confidence modeling of LBD could translate into meaningful improvements for downstream tasks, we turn to **RQ5.4** and ask *whether LBD can reach higher performance in a high-precision recommendation task*.

For this question, we simulate a notification task where each user can only be recommended a single item, and due to the mild intrusiveness of a notification, only N total users will receive a recommendation. Therefore, this task could be divided into two stages: first to recognize what is the best item to recommend for each user, and second, to determine which N recommendations of the first stage are most likely to result in a positive reaction. We expect that an accurate confidence model is better suited for both stages of this task.

We divide our simulation in these stages: first, for each user we select the test set item that has the highest predicted probability of a 4.5+ star rating (i.e., for models not relying on binning: the probability of being at least 4.25). For MF, this is also equivalent to simply using its predicted rating value [137]. We then use the above estimates to order the resulting user-item combinations. Then, for a given N , we calculate what percentage of the top- N user-item combinations truly did receive a 4.5+ star rating in the test set (Precision@1), i.e., the number of users who are happy with their notification. Note that N does not exceed the true number of users who have a 4.5+ star test set rating.

Figure 5.4 shows the results for this experiment in terms of Precision@1 for various values of N as well as models' relative improvement in the metric over MF. From the figure we see that all methods can identify a highly relevant item for most users, especially when N is small, with the task becoming progressively more difficult as N increases. Surprisingly, our results only show minor differences between CMF and MF, with the relative improvement of CMF limited to below 2% for most values of N . In contrast, OrdRec-UI, LBD-S and LBD-A start off with a substantially higher precision@1 for $N = 100$ and consistently outperform CMF even as N increases and users with lower confidence are included. Model performance for a selection of user numbers is also shown in Table 5.4. We can see that the performance of LBD-S is similar to that of OrdRec-UI, with frequently overlapping means and no significant improvement of one model over other. It appears that OrdRec-UI's weaker confidence modeling may still be quite efficient at identifying low-error high-rating items. LBD-A, on the other hand, consistently achieves the highest performance, with particularly noticeable improvement over OrdRec-UI of approximately 1.5% (absolute

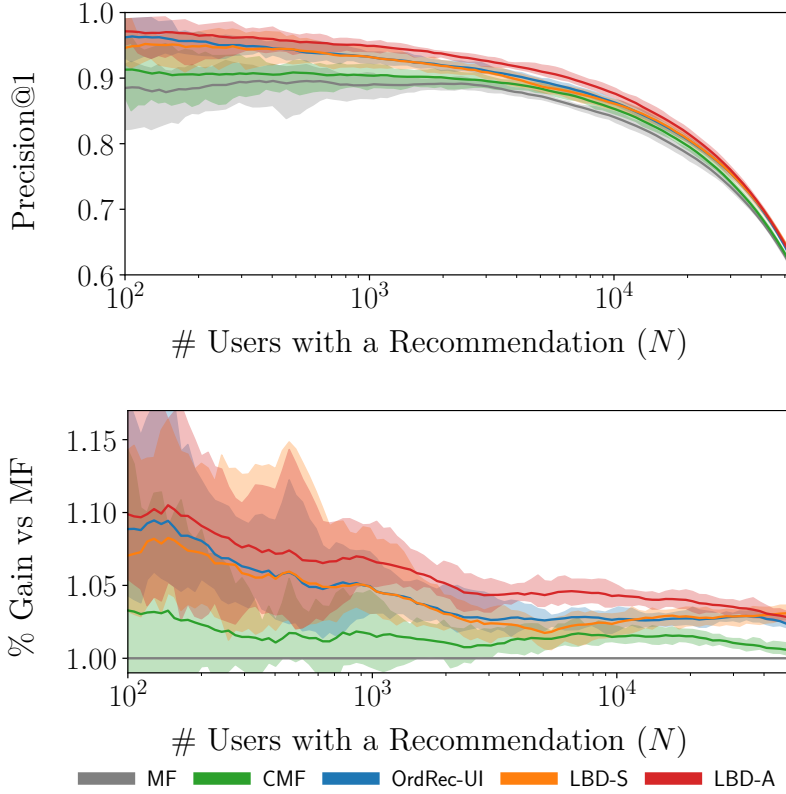


Figure 5.4: Model performance on the high-precision targeted recommendation task for distinct numbers of targeted users (N) with MF, CMF, OrdRec and LBD methods. Left: Percentage of users for whom a 4.5+ star recommendation is made (Precision@1). Right: Relative Precision@1 improvement over Matrix Factorization. Results are averages over 10 folds. Shaded regions denote minimum and maximum across folds.

Table 5.4: Model performance on the high-precision targeted recommendation task for distinct numbers of targeted users (N , half-powers of 10). Results are Precision@1 averages from 10-fold cross-validation, with standard deviations in parentheses. Highest performance for each value of N is denoted in bold. Significant improvement of an LBD model over *all* baseline models is denoted by Δ , no significant improvement of any of the baselines over either LBD model (separate one-sided Wilcoxon signed-rank tests with matched folds, $p < 0.001$).

Model	Precision@1 for top N users		
	$N = 100$	$N = 320$	$N = 1000$
MF	0.885 (0.034)	0.893 (0.019)	0.889 (0.011)
CMF	0.913 (0.030)	0.907 (0.016)	0.904 (0.013)
OrdRec-UI	0.962 (0.016)	0.949 (0.007)	0.932 (0.006)
LBD-S	0.947 (0.019)	0.944 (0.013)	0.932 (0.009)
LBD-A	0.971 (0.012)	0.962 (0.011) $^\Delta$	0.949 (0.005) $^\Delta$

Model	Precision@1 for top N users		
	$N = 3200$	$N = 10000$	$N = 32000$
MF	0.886 (0.004)	0.841 (0.004)	0.726 (0.002)
CMF	0.895 (0.005)	0.853 (0.005)	0.733 (0.002)
OrdRec-UI	0.910 (0.003)	0.863 (0.003)	0.747 (0.002)
LBD-S	0.908 (0.006)	0.861 (0.004)	0.747 (0.003)
LBD-A	0.925 (0.004) $^\Delta$	0.877 (0.004) $^\Delta$	0.751 (0.003) $^\Delta$

differences) for $1000 \leq N \leq 10,000$. As the improvements of LBD-A are also significant for all listed values $N > 100$, we conclude that LBD-A is meaningfully better at this high-precision targeted recommendation task, likely due to its better confidence modeling capabilities. As such, we also answer **RQ5.4** in the affirmative: *LBD, and particularly LBD-A, brings a meaningful performance improvement for the high-precision targeted recommendation task over MF, CMF and OrdRec.* Combined with the conclusion from **RQ5.3**, our results strongly indicate that the confidence model of LBD is both a better indication of its prediction accuracy and that this higher accuracy can be translated to meaningful benefits for downstream tasks.

5.7 Conclusion and Future Work

In this work, we investigated whether confidence modeling for RecSys is possible while maintaining high predictive accuracy and without introducing

high model-complexity or heavy computational costs. To this end, we proposed learned beta distributions (LBD) as a simple RecSys with an explicit model of confidence while having the same model-complexity as MF.

Our experimental results show that LBD has competitive predictive performance with MF, CMF and OrdRec, where no decreases were observed for any metric. Furthermore, unlike CMF and OrdRec, we found that the confidence of LBD correlates strongly with its accuracy, allowing it to better separate predictions on their accuracy. Finally, we found that LBD’s superior confidence modeling translates to a meaningfully higher precision in a high-precision targeted recommendation task, compared to the other methods.

We conclude that with the introduction of LBD, we have shown that accurate recommendation can be combined with accurate confidence modeling and low model-complexity. This also allows us to answer our final thesis research question **RQ5** in the affirmative: variance estimation be incorporated into rating prediction algorithms as confidence modeling in a computationally efficient manner.

Our findings also indicate that such confidence models can lead to meaningful improvements in downstream tasks: LBD thereby provides a very promising potential for novel future applications. Future work could investigate whether the confidence model of LBD can be used to better understand user preferences, or the uncertainty contained in datasets. While this work focused on matching the model-complexity of MF, the concept of LBD is easily applicable or extendable to more complex RecSys models. Similarly, while we tackled rating-prediction, LBD could be extended to confidence modeling for predictions of implicit feedback signals, e.g., clicks, purchases or watchtime. Additionally, it would be of interest to better understand how confidence modeling of LBD interacts with various cognitive and statistical biases, and the approaches used to correct them [77, 105, 178, 196]. Finally, LBD is designed to be practical for real-world settings; therefore, we hope practitioners will consider LBD and its many potential applications.

Chapter 6

Conclusion

In this thesis we examined the role of bias, variance, consistency and efficiency on estimation methods for ranking and recommendation. In this final chapter we revisit the findings of our previous chapters, summarizing and discussing them in their entirety. Finally, we will complete the thesis by considering the potential future research directions in addressing statistical and interaction biases in recommender systems and learning to rank.

6.1 Main Findings

Chapter 2 examined the impact of the bandwagon effect [193, 218] on the statistical properties of relevance estimators, asking:

RQ1 *Is the bandwagon effect in user interactions a problem of selection bias or is it better described in terms of other statistical properties?*

In our analysis, we argued that the bandwagon effect should not be seen as a problem of statistical bias, as we showed that under the simple form of the bandwagon effect, the sample average of ratings of all users is in fact unbiased. Nevertheless, we proved that the bandwagon effect can make estimators inconsistent and less data-efficient, introducing a distinct set of problems for convergence in relevance estimation. Furthermore, we proved a novel necessary condition for the consistency of the rating mean under the bandwagon effect, and proposed methods for mitigating the effect’s impact on data-efficiency. Therefore, the answer to our research question **RQ1** is: the bandwagon effect is best viewed as a problem of consistency and data-efficiency, not statistical bias. The importance of this finding is two-fold. Firstly, it highlights that the bandwagon effect poses a potentially significant yet underinvestigated open problem that is fundamentally distinct from the

well-studied position and selection biases in ranking in recommendation. And secondly, it underscores the importance of understanding the effects of cognitive biases on statical properties of estimators, as our analysis allowed us to both identify the mechanism affecting relevance estimation under the effect as well as to propose potential solutions to this problem.

In Chapter 3, we turned our attention to accurate and computationally efficient estimation of document placement probabilities (a.k.a. propensities) for the Plackett-Luce (PL) ranking model [143, 154]. As the accuracy of current sampling-based estimation methods [69, 143] may be in practice limited by their sample- and computational efficiency, we asked our second research question:

RQ2 *Can we accurately and efficiently estimate document placement probabilities for the Plackett-Luce ranking model without sampling?*

We then proposed the novel Marginalized Plackett-Luce (MPL) method to efficiently and accurately calculate document-rank placement probabilities under the PL ranking model. We argued and subsequently showed that MPL provides near-exact estimation when computing the function over a practical number of evaluation points, allowing for accurate and computationally efficient estimation and thus an affirmative answer to our research question **RQ2**.

The same chapter also explored the downstream application of the estimated propensities to off-policy evaluation (OPE) [118]. In particular, we asked:

RQ3 *Do our new propensity estimates allow us to reduce bias and improve data-efficiency in off-policy learning to rank?*

We also answered this question in the affirmative. In particular, we showed that OPE performance using propensity estimates of MPL was comparable to that of the asymptotically unbiased sampling-based propensity estimation method using a large number of samples, thus indicating a very low level of bias for MPL. Furthermore, compared to the sampling approach with the same computational cost, MPL propensities also allowed us to take advantage of higher amounts of user interactions due to lower needed levels of propensity clipping. Taken together, our findings thus underscore the importance of developing low-variance and computationally efficient estimation methods, which may allow for both lower runtimes as well as better performance in both the estimation as well as downstream tasks. Importantly, our improvement in these statistical properties also opens the door to the improved applicability of Plackett-Luce models to both OPE as well as other ranking and recommendation tasks.

Chapter 4 then explored the interaction between the order and the presentation of ranking results, particularly in the context of optimization efficiency. In this chapter, we first formally introduced the variable result presentation length ranking setting. We then examined whether the PL ranking model [154, 143] can also be used to provide high-quality rankings in an efficient manner in this setting, leading to our research question **RQ4**.

RQ4 *Can the Plackett-Luce ranking model be extended to simultaneously learn document ordering and presentation size in a computationally and sample-efficient way?*

We answered this question in the affirmative by introducing variable length Plackett-Luce (VLPL) – a new family of Plackett-Luce list-wise gradient estimation methods for the joint optimization of document ordering and lengths. Through our semi-synthetic experiments we showed that VLPL can effectively balance the expected exposure and attractiveness of all documents, achieving the best performance across different ranking settings, with our VLPL-2 estimator providing particularly high sample- and computational efficiency. Our findings in this chapter thus demonstrate, that despite the inherent complexity of this frequently occurring [176, 236] yet understudied setting [150, 151], computationally and sample-efficient low-variance solutions also allow for improved ranking performance.

Finally, Chapter 5 focused on the computationally efficient confidence modeling for recommendation. In particular, current confidence modeling approaches in recommender systems are either inaccurate heuristics, conceptually complex or computationally very expensive [3, 14, 134, 169, 205]. As prediction variance may also be a natural method for representing model confidence, our final research question **RQ5** asked:

RQ5 *Can variance estimation be incorporated into rating prediction algorithms as confidence modeling in a computationally efficient manner?*

We answered this question in the affirmative by proposing learned beta distributions (LBD) as a simple and practical recommendation method with an explicit measure of confidence through its prediction variance. We showed that LBD maintains competitive accuracy to existing methods while also having a significantly stronger correlation between its accuracy and confidence, allowing it to achieve higher performance when applied to a high-precision targeted recommendation task. Overall, our results thus show that understanding the statistical properties of relevance estimators and their predictions can play an important role in recommendation, and that this understanding does not necessarily have to come at the cost of increased computational complexity.

6.2 Summary of Main Findings

We will now reflect on our findings and summarize them in the form of our guidance to researchers and practitioners. To begin, in Section 1.1 we outline the significance of the following three properties of statistical estimators: variance, consistency and efficiency. The importance of these statistical properties is also confirmed by our work on relevance estimation under the bandwagon effect (Chapter 2), where we observed that this cognitive bias may reduce estimator data-efficiency and lead to convergence issues due to estimator variance and inconsistency. Furthermore, when considering the sample- and computational efficiency from Section 1.2, we found that lower estimator variance also allowed for more sample- and computationally efficient propensity estimation (Chapter 3) and model optimization in the variable presentation length setting (Chapter 4). Based on our findings, variance and efficiency may thus have a significant impact on both the accuracy as well other statistical properties of the estimators – our first conclusion is therefore that: *it is important to develop and leverage low-variance and efficient estimators wherever feasible.*

In addition to the above properties, as highlighted in Subsection 1.1.1, another desirable estimator quality is unbiasedness. Importantly, a significant amount of research in off-policy evaluation (OPE) and learning for ranking and recommendation focuses on developing unbiased estimators [28, 145, 168], analogous to our work for conditional bias in Chapter 2. Nevertheless, our results across Chapters 2 – 4 highlight the importance of further considering other statistical properties in addition to bias. For instance, we previously already noted the connection between estimator variance, consistency and efficiency under the bandwagon effect. Bias and variance on the other hand, may exhibit more complex interaction patterns, that may, in turn, also impact other properties. Whilst traditional methods like clipping and self-normalized inverse propensity scoring may introduce bias to reduce estimator variance, in Chapter 3 we also observe that lower variance estimates in one system component (e.g., propensities) may allow for a reduction in bias in another (e.g., relevance estimation error), for example through improved data-efficiency. These interaction patterns between estimator properties thus lead to our second conclusion, building on the previous one: *optimal estimation generally requires the joint consideration of multiple statistical properties, such as bias, variance, consistency and efficiency, of both the estimator as well as the estimation pipeline as a whole.* Notably, this conclusion is also consistent with some of the recent work in OPE [141, 202], where Udagawa et al. [202] instead focus on a combination

of statistical properties, i.e., estimator bias and variance, combining them in the form of the mean squared error.

Finally, as highlighted in Chapter 5, another important consideration may be the conceptual complexity of an estimator, along with its trade-off with the statistical estimator properties. Whilst not directly addressed by our research questions, conceptual complexity may have a strong impact on the usability of an estimator, particularly for practitioners. For instance, whilst our results in Chapter 3 highlight the substantial improvement of MPL in propensity estimation and OPE, the sampling-based approach may in some cases still provide sufficiently accurate propensities for a sufficiently accurate relevance estimate: if the computational cost of sampling is not important, or under strong clipping, the lower conceptual complexity of the sampling approach may instead allow for easier implementation and interpretability of the method. Conversely, as in the variable presentation length setting the performance improvements of VLPL may merit the increased computational cost and the reduced conceptual simplicity suggests that this trade-off may be unique to each setting and estimator. In particular, we believe there may be only limited instances where improved performance or downstream use may be achieved by a conceptually and computationally lightweight method, as the one we propose in Chapter 5. Altogether, combined with our earlier conclusions, our final and main conclusion is therefore that: *estimation methods for ranking and recommendation should holistically consider the statistical properties, such as bias, variance, consistency and efficiency, together with the non-statistical properties, such as conceptual complexity and interpretability, of the entire estimation pipeline to ensure the wide applicability and usefulness of these methods.*

6.3 Future Work

To conclude the thesis, we highlight the remaining open questions resulting from this thesis. To begin, whilst our analysis of the bandwagon effect in Chapter 2 focused on a specific model of the effect, other more complex formulations [229, 234], e.g., where the effect’s strength may differ across users, may have a different effect on statistical properties of estimators. Importantly, whilst users are frequently exposed to feedback from other users [176, 236], we found no easy solution even under our simple model, highlighting the complex yet understudied nature of the bandwagon effect.

For the task of sample-efficient propensity estimation under the PL model in Chapter 3, our proposed estimator demonstrated high computational efficiency at standard ranking sizes. Nevertheless, its runtime may be suboptimal with longer ranking sizes due to its quadratic scaling. Furthermore, the method may be also used for direct ranking optimization, which was outside of the scope of this thesis.

PL models can also be extended to the variable result presentation length setting, as in Chapter 4 VLPL was shown to be efficient under a fixed number of presentation slots. Nonetheless, it remains unclear whether there may be more computationally efficient approaches for gradient estimation in the setting where document presentation lengths are measured in pixels or are instead distinct but fixed for each document.

Furthermore, whilst our other chapters focus on minimizing variance, our approach for Beta distributions in Chapter 5 may instead be further leveraged to better understand the uncertainty contained in datasets and preferences of individual users. Yet, at the same time, it remains unclear how the confidence modeling through variance of our method may be impacted when combining it with inverse propensity scoring and associated variance reduction methods for learning from biased user interactions.

Finally, as highlighted by our entire work and underscored in Chapter 6, there is often no structured way to combine and balance accounting for one or more statistical properties of any given estimation method with the implementational and theoretical difficulty of the approach. Future research may thus wish to focus on more measurable and principled balancing of conceptual complexity of estimation methods for ranking and recommendation with their ability to address bias, variance, consistency and efficiency.

Bibliography

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. In *12th USENIX Symposium on Operating Systems Design and Implementation*), pages 265–283, 2016.
- [2] Gediminas Adomavicius and Alexander Tuzhilin. Context-Aware Recommender Systems. In *Recommender Systems Handbook*, pages 217–253. Springer US, 2011.
- [3] Gediminas Adomavicius, Sreeharsha Kamireddy, and YoungOk Kwon. Towards More Confident Recommendations: Improving Recommender Systems Using Filtering Approach Based on Rating Variance. *Proceedings of the 17th Workshop on Information Technology and Systems*, page 6, 2007.
- [4] Gediminas Adomavicius, Jesse C. Bockstedt, Shawn P. Curley, and Jingjing Zhang. Do Recommender Systems Manipulate Consumer Preferences? A Study of Anchoring Effects. *Information Systems Research*, 24(4):956–975, 2013.
- [5] Gediminas Adomavicius, Jesse C. Bockstedt, Shawn Curley, and Jingjing Zhang. Understanding Effects of Personalized vs. Aggregate Ratings on User Preferences. *CEUR Workshop Proceedings*, 1679: 14–21, 2016.
- [6] Aman Agarwal, Xuanhui Wang, Cheng Li, Michael Bendersky, and Marc Najork. Addressing Trust Bias for Unbiased Learning-to-Rank. In *The World Wide Web Conference*, pages 4–14. ACM, 2019.

- [7] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. Unbiased Learning to Rank with Unbiased Propensity Estimation. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 385–394. ACM, 2018.
- [8] Luca Aiello, Ioannis Arapakis, Ricardo Baeza-Yates, Xiao Bai, Nicola Barbieri, Amin Mantrach, and Fabrizio Silvestri. The Role of Relevance in Sponsored Search. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 185–194. ACM, 2016.
- [9] Vito Walter Anelli, Amra Delić, Gabriele Sottocornola, Jessie Smith, Nazareno Andrade, Luca Belli, Michael Bronstein, Akshay Gupta, Sofia Ira Ktena, Alexandre Lung-Yut-Fong, Frank Portman, Alykhan Tejani, Yuanpu Xie, Xiao Zhu, and Wenzhe Shi. RecSys 2020 Challenge Workshop: Engagement Prediction on Twitter’s Home Timeline. In *Fourteenth ACM Conference on Recommender Systems*, pages 623–627, 2020.
- [10] Imad Aouali, Victor-Emmanuel Brunel, David Rohde, and Anna Korba. Exponential Smoothing for Off-Policy Learning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 984–1017. PMLR, 2023.
- [11] Solomon E Asch. Effects of Group Pressure Upon the Modification and Distortion of Judgments. In *Organizational Influence Processes*, pages 295–303. Routledge, 2016.
- [12] Leif Azzopardi. Cognitive Biases in Search: A Review and Reflection of Cognitive Biases in Information Retrieval. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, pages 27–37. Association for Computing Machinery, 2021.
- [13] Leonard E Baum. An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes. *Inequalities*, 3(1):1–8, 1972.
- [14] Cesare Bernardis, Maurizio Ferrari Dacrema, and Paolo Cremonesi. Estimating Confidence of Individual User Predictions in Item-based Recommender Systems. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, pages 149–156. Association for Computing Machinery, 2019.

- [15] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside PageRank. *ACM Transactions on Internet Technology*, 5(1):92–128, 2005. ISSN 1533-5399, 1557-6051.
- [16] William Biscarri, Sihai Dave Zhao, and Robert J. Brunner. A Simple and Fast Method for Computing the Poisson Binomial Distribution Function. *Computational Statistics & Data Analysis*, 122:92–100, 2018. ISSN 0167-9473.
- [17] Guy Blanc and Steffen Rendle. Adaptive Sampled Softmax with Kernel Based Sampling. In *Proceedings of the 35th International Conference on Machine Learning*, pages 590–599. PMLR, 2018.
- [18] Eugene M Bland, Gregory S Black, and Kay Lawrimore. Determinants of Effectiveness and Success for EBay Auctions. *The Coastal Business Journal*, 4(1), 2005.
- [19] Ignace Bogaert. Iteration-Free Computation of Gauss-Legendre Quadrature Nodes and Weights. *SIAM Journal on Scientific Computing*, 36(3):A1008–A1026, 2014. ISSN 1064-8275.
- [20] Théo Bontempelli, Benjamin Chapus, François Rigaud, Mathieu Morlon, Marin Lorant, and Guillaume Salha-Galvan. Flow Moods: Recommending Music by Moods on Deezer. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 452–455. Association for Computing Machinery, 2022.
- [21] Christian Borghesi and Jean-Philippe Bouchaud. Spatial Correlations in Vote Statistics: A Diffusive Field Model for Decision-Making. *The European Physical Journal B*, 75(3):395–404, 2010. ISSN 1434-6036.
- [22] Djallel Bouneffouf, Amel Bouzeghoub, and Alda Lopes Ganarski. Risk-Aware Recommender Systems. In *Neural Information Processing*, pages 57–65. Springer, 2013.
- [23] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Neca, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable Transformations of Python+NumPy Programs, 2018.

- [24] Léa Briand, Théo Bontempelli, Walid Bendada, Mathieu Morlon, François Rigaud, Benjamin Chapus, Thomas Bouabça, and Guillaume Salha-Galvan. Let's Get It Started: Fostering the Discoverability of New Releases on Deezer. In *Advances in Information Retrieval*, pages 286–291. Springer Nature Switzerland, 2024.
- [25] Sebastian Bruch, Shuguang Han, Michael Bendersky, and Marc Najork. A Stochastic Treatment of Learning to Rank Scoring Functions. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 61–69, 2020.
- [26] Alexander Buchholz, Jan Malte Lichtenberg, Giuseppe Di Benedetto, Yannik Stein, Vito Bellini, and Matteo Ruffini. Low-variance estimation in the Plackett-Luce Model via Quasi-Monte Carlo Sampling. *arXiv preprint arXiv:2205.06024*, 2022.
- [27] Alexander Buchholz, Ben London, Giuseppe di Benedetto, and Thorsten Joachims. Off-Policy Evaluation for Learning-to-Rank via Interpolating the Item-Position Model and the Position-Based Model. *arXiv preprint arXiv:2210.09512*, 2022.
- [28] Alexander Buchholz, Ben London, Giuseppe Di Benedetto, Jan Malte Lichtenberg, Yannik Stein, and Thorsten Joachims. Counterfactual Ranking Evaluation with Flexible Click Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1200–1210. ACM, 2024.
- [29] 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, 2005.
- [30] Christopher J.C. Burges. From RankNet to LambdaRank to LambdaMART: An Overview. Technical Report MSR-TR-2010-82, Microsoft, 2010.
- [31] Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press, 2016.

- [32] Rocío Cañamares and Pablo Castells. Should I Follow The Crowd? A Probabilistic Analysis of the Effectiveness of Popularity in Recommender Systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 415–424, 2018.
- [33] 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, 2007.
- [34] Franco Caparrelli, Anna M. D’Arcangelis, and Alexander Cassuto. Herding in the Italian Stock Market: A Case of Behavioral Finance. *Journal of Behavioral Finance*, 5(4):222–230, 2004. ISSN 1542-7560.
- [35] George Casella and Edward I. George. Explaining the Gibbs Sampler. *The American Statistician*, 46(3):167–174, 1992. ISSN 0003-1305.
- [36] Simon Caton and Christian Haas. Fairness in Machine Learning: A Survey. *ACM Computing Surveys*, 56(7):1–38, 2024. ISSN 0360-0300, 1557-7341.
- [37] Olivier Chapelle and Yi Chang. Yahoo! Learning to Rank Challenge Overview. *Journal of Machine Learning Research*, 14:1–24, 2011.
- [38] Ruey-Cheng Chen, Qingyao Ai, Gaya Jayasinghe, and W. Bruce Croft. Correcting for Recency Bias in Job Recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2185–2188. Association for Computing Machinery, 2019.
- [39] Xiaoshuang Chen, Yibo Wang, Yao Wang, Husheng Liu, Kaiqiao Zhan, Ben Wang, and Kun Gai. Creator-Side Recommender System: Challenges, Designs, and Applications. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 162–170. ACM, 2025.
- [40] 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. JMLR Workshop and Conference Proceedings, 2011.
- [41] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. *Click Models for Web Search*. Morgan & Claypool Publishers, 2015.

- [42] William G Cochran. *Sampling Techniques*. Wiley, 1977.
- [43] Dan Cosley, Shyong K. Lam, Istvan Albert, Joseph A. Konstan, and John Riedl. Is Seeing Believing? How Recommender Interfaces Affect Users' Opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 585–592, 2003.
- [44] Paul Covington, Jay Adams, and Emre Sargin. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198. ACM, 2016.
- [45] 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, 2008.
- [46] Martin J. Crowder. Maximum Likelihood Estimation for Dependent Observations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 38(1):45–53, 1976.
- [47] J. Shane Culpepper, Charles L. A. Clarke, and Jimmy Lin. Dynamic Cutoff Prediction in Multi-Stage Retrieval Systems. In *Proceedings of the 21st Australasian Document Computing Symposium*, pages 17–24. Association for Computing Machinery, 2016.
- [48] Edward Cutrell and Zhiwei Guan. What Are You Looking for?: An Eye-Tracking Study of Information Usage in Web Search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 407–416. ACM, 2007.
- [49] Annie AM Cuyt, Vigdis Petersen, Brigitte Verdonk, Haakon Waadeland, and William B Jones. *Handbook of Continued Fractions for Special Functions*. Springer, 2008.
- [50] Yashar Deldjoo, Markus Schedl, Paolo Cremonesi, and Gabriella Pasi. Recommender Systems Leveraging Multimedia Content. *ACM Computing Surveys*, 53(5):106:1–106:38, 2020. ISSN 0360-0300.
- [51] Utpal M. Dholakia and Kerry Soltysinski. Coveted or Overlooked? The Psychology of Bidding for Comparable Listings in Digital Auctions. *Marketing Letters*, 12(3):225–237, 2001.

- [52] Fernando Diaz, Bhaskar Mitra, Michael D. Ekstrand, Asia J. Biega, and Ben Carterette. Evaluating Stochastic Rankings with Expected Exposure. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20*, page 275–284. Association for Computing Machinery, 2020.
- [53] Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. Doubly Robust Policy Evaluation and Optimization. *Statistical Science*, 29(4):485–511, 2014.
- [54] Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. Pixie: A System for Recommending 3+ Billion Items to 200+ Million Users in Real-Time. In *Proceedings of the 2018 World Wide Web Conference*, pages 1775–1784, 2018.
- [55] 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.
- [56] Manuel Fernandez and Stuart Williams. Closed-Form Expression for the Poisson-Binomial Probability Density Function. *IEEE Transactions on Aerospace and Electronic Systems*, 46(2):803–817, 2010. ISSN 1557-9603.
- [57] Erasmo M. Ferreira, Anderson K. Kohara, and Javier Sesma. New Properties of the Lerch’s Transcendent. *Journal of Number Theory*, 172:21–31, 2017. ISSN 0022314X.
- [58] Andrew J. Flanagin and Miriam J. Metzger. Trusting Expert- versus User-Generated Ratings Online: The Role of Information Volume, Valence, and Consumer Characteristics. *Computers in Human Behavior*, 29(4):1626–1634, 2013. ISSN 07475632.
- [59] Michael A. Fligner and Joseph S. Verducci. Multistage Ranking Models. *Journal of the American Statistical Association*, 83(403): 892–901, 1988. ISSN 0162-1459.
- [60] W. Wayne Fu and Clarice C. Sim. Aggregate Bandwagon Effect on Online Videos’ Viewership: Value Uncertainty, Popularity Cues, and Heuristics. *Journal of the American Society for Information Science and Technology*, 62(12):2382–2395, 2011.

- [61] Norbert Fuhr. Optimum Polynomial Retrieval Functions Based on the Probability Ranking Principle. *ACM Transactions on Information Systems*, 7(3):183–204, 1989. ISSN 1046-8188.
- [62] Aurel Galántai. The Theory of Newton’s Method. *Journal of Computational and Applied Mathematics*, 124(1):25–44, 2000.
- [63] Chen Gao, Yu Zheng, Wenjie Wang, Fuli Feng, Xiangnan He, and Yong Li. Causal Inference in Recommender Systems: A Survey and Future Directions. *ACM Transactions on Information Systems*, 42(4): 88:1–88:32, 2024. ISSN 1046-8188.
- [64] Sahin Cem Geyik, Qi Guo, Bo Hu, Cagri Ozcaglar, Ketan Thakkar, Xianren Wu, and Krishnaram Kenthapadi. Talent Search and Recommendation Systems at LinkedIn: Practical Challenges and Lessons Learned. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1353–1354. ACM, 2018.
- [65] Oded Goldreich. A Sample of Samplers: A Computational Perspective on Sampling. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 302–332. Springer, 2011.
- [66] Congying Guan, Shengfeng Qin, Wessie Ling, and Guofu Ding. Apparel Recommendation System Evolution: An Empirical Review. *International Journal of Clothing Science and Technology*, 28(6):854–879, 2016. ISSN 0955-6222.
- [67] Akhil Gudivada and Nasseh Tabrizi. A Literature Review on Machine Learning Based Medical Information Retrieval Systems. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 250–257, 2018.
- [68] Arjun K Gupta and Saralees Nadarajah. *Handbook of Beta Distribution and its Applications*. CRC press, 2004.
- [69] Philipp Hager, Onno Zoeter, and Maarten de Rijke. Unidentified and Confounded? Understanding Two-Tower Models for Unbiased Learning to Rank. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, pages 347–357. Association for Computing Machinery, 2025.

- [70] Ward A. Hanson and Daniel S. Putler. Hits and Misses: Herd Behavior and Online Product Popularity. *Marketing Letters*, 7(4):297–305, 1996. ISSN 1573-059X.
- [71] F Maxwell Harper and Joseph A Konstan. The Movielens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):1–19, 2015.
- [72] 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.
- [73] José M. Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. Probabilistic Matrix Factorization with Non-random Missing Data. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1512–1520, 2014.
- [74] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, 2008.
- [75] Jen-Hung Huang and Yi-Fen Chen. Herding in Online Product Choice. *Psychology & Marketing*, 23(5):413–428, 2006.
- [76] Jin Huang, Harrie Oosterhuis, Maarten de Rijke, and Herke van Hoof. Keeping Dataset Biases out of the Simulation: A Debiased Simulator for Reinforcement Learning based Recommender Systems. In *Fourteenth ACM Conference on Recommender Systems*, pages 190–199. ACM, 2020.
- [77] Jin Huang, Harrie Oosterhuis, and Maarten De Rijke. It Is Different When Items Are Older: Debiasing Recommendations When Selection Bias and User Preferences Are Dynamic. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 381–389. ACM, 2022.
- [78] Iris A. M. Huijben, Wouter Kool, Max B. Paulus, and Ruud J. G. van Sloun. A Review of the Gumbel-max Trick and its Extensions for Discrete Stochasticity in Machine Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1353–1371, 2023. ISSN 1939-3539.

- [79] Eyke Hüllermeier and Willem Waegeman. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. *Machine Learning*, 110(3):457–506, 2021.
- [80] Ilija Ilievski and Sujoy Roy. Personalized News Recommendation Based On Implicit Feedback. In *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*, pages 10–15, 2013.
- [81] Rolf Jagerman, Ilya Markov, and Maarten de Rijke. When People Change their Mind: Off-Policy Evaluation in Non-stationary Recommendation Environments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, pages 447–455, New York, NY, USA, January 2019. Association for Computing Machinery.
- [82] Dietmar Jannach, Lukas Lerche, and Markus Zanker. Recommending Based on Implicit Feedback. In Peter Brusilovsky and Daqing He, editors, *Social Information Access: Systems and Technologies*, pages 510–569. Springer International Publishing, 2018.
- [83] Gawesh Jawaheer, Peter Weller, and Patty Kostkova. Modeling User Preferences in Recommender Systems: A Classification Framework for Explicit and Implicit User Feedback. *ACM Transactions on Interactive Intelligent Systems*, 4(2):8:1–8:26, 2014. ISSN 2160-6455.
- [84] Olivier Jeunen and Bart Goethals. Pessimistic Reward Models for Off-Policy Learning in Recommendation. In *Fifteenth ACM Conference on Recommender Systems*, pages 63–74, 2021.
- [85] Olivier Jeunen, Ivan Potapov, and Aleksei Ustimenko. On (Normalised) Discounted Cumulative Gain as an Off-Policy Evaluation Metric for Top- n Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1222–1233. ACM, 2024.
- [86] Rong Jin, Luo Si, ChengXiang Zhai, and Jamie Callan. Collaborative Filtering with Decoupled Models for Preferences and Ratings. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 309–316. ACM, 2003.

- [87] Thorsten Joachims. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.
- [88] Thorsten Joachims. Evaluating Retrieval Performance Using Clickthrough Data. In *Text Mining*. Physica Verlag, 2003.
- [89] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately Interpreting Clickthrough Data as Implicit Feedback. In *ACM SIGIR Forum*, volume 51, pages 4–11. ACM, 2017.
- [90] 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, 2017.
- [91] Christopher C Johnson. Logistic Matrix Factorization for Implicit Feedback Data. *Advances in Neural Information Processing Systems*, 27(78):1–9, 2014.
- [92] Norman L Johnson, Samuel Kotz, and N Balakrishnan. *Beta Distributions*. John Wiley and Sons, 1994.
- [93] Clark Mingxuan Ju, Leonardo Neves, Bhuvesh Kumar, Liam Collins, Tong Zhao, Yuwei Qiu, Qing Dou, Yang Zhou, Sohail Nizam, Rengim Aykan Ozturk, Yvette Liu, Sen Yang, Manish Malik, and Neil Shah. Learning Universal User Representations Leveraging Cross-domain User Intent at Snapchat. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4345–4349. Association for Computing Machinery, 2025.
- [94] Michael Jugovac and Dietmar Jannach. Interacting with Recommenders—Overview and Research Directions. *ACM Transactions on Interactive Intelligent Systems*, 7(3):1–46, 2017. ISSN 2160-6455, 2160-6463.
- [95] Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. On Application of Learning to Rank for E-Commerce Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 475–484. ACM, 2017.

- [96] Diane Kelly and Leif Azzopardi. How Many Results per Page?: A Study of SERP Size, Search Behavior and User Experience. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 183–192. ACM, 2015.
- [97] Zahid Younas Khan, Zhendong Niu, Sulis Sandiwarno, and Rukundo Prince. Deep Learning Techniques for Rating Prediction: A Survey of the State-of-the-Art. *Artificial Intelligence Review*, 54(1):95–135, 2021. ISSN 1573-7462.
- [98] Ashish Khetan and Sewoong Oh. Computational and Statistical Tradeoffs in Learning to Rank. In *Advances in Neural Information Processing Systems*, 2016.
- [99] Kirill Khrylchenko, Vladimir Baikalov, Sergei Makeev, Artem Matveev, and Sergei Liamaev. Correcting the LogQ Correction: Revisiting Sampled Softmax for Large-Scale Retrieval. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pages 545–550. Association for Computing Machinery, 2025.
- [100] Jaewon Kim, Paul Thomas, Ramesh Sankaranarayana, Tom Gedeon, and Hwan-Jin Yoon. Pagination versus Scrolling in Mobile Web Search. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 751–760. ACM, 2016.
- [101] Jinyoung Kim and Andrew Gambino. Do We Trust the Crowd or Information System? Effects of Personalization and Bandwagon Cues on Users’ Attitudes and Behavioral Intentions toward a Restaurant Recommendation Website. *Computers in Human Behavior*, 65:369–379, 2016.
- [102] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [103] Michael Kläs and Anna Maria Vollmer. Uncertainty in machine learning applications: A practice-driven classification of uncertainty. In *International Conference on Computer Safety, Reliability, and Security*, pages 431–438. Springer, 2018.

- [104] Bart P Knijnenburg, Martijn C Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4): 441–504, 2012.
- [105] Norman Knyazev and Harrie Oosterhuis. The Bandwagon Effect: Not Just Another Bias. In *Proceedings of the 2022 ACM SIGIR International Conference on the Theory of Information Retrieval*, 2022.
- [106] Ron Kohavi. Online Controlled Experiments: Lessons from Running A/B/n Tests for 12 Years. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015.
- [107] Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic Beams and Where To Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3499–3508. PMLR, 2019.
- [108] Wouter Kool, Herke van Hoof, and Max Welling. Estimating Gradients for Discrete Random Variables by Sampling without Replacement. In *International Conference on Learning Representations*, 2019.
- [109] Yehuda Koren. Collaborative Filtering with Temporal Dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 447–456. ACM, 2009.
- [110] Yehuda Koren and Robert Bell. Advances in Collaborative Filtering. In *Recommender Systems Handbook*, pages 145–186. Springer US, 2011.
- [111] Yehuda Koren and Joe Sill. OrdRec: An Ordinal Model for Predicting Personalized Item Rating Distributions. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, pages 117–124. ACM, 2011.
- [112] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37, 2009.
- [113] László Kozma. Useful inequalities cheat sheet, 2021. URL http://www.lkozma.net/inequalities_cheat_sheet/ineq.pdf.

- [114] Anagha Kulkarni, Almer S. Tigelaar, Djoerd Hiemstra, and Jamie Callan. Shard Ranking and Cutoff Estimation for Topically Partitioned Collections. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 555–564. Association for Computing Machinery, 2012.
- [115] Bibb Latané and Sharon Wolf. The Social Impact of Majorities and Minorities. *Psychological Review*, 88(5):438–453, 1981.
- [116] Isaac Leonard. Notes on the Riemann Zeta Function. MATH 324, Summer 2012, Elementary Number Theory, University of Alberta, 2012. URL <http://www.math.ualberta.ca/~isaac/math324/s12/zeta.pdf>.
- [117] Ming Li, Ali Vardasbi, Andrew Yates, and Maarten De Rijke. Repetition and Exploration in Sequential Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2532–2541. ACM, 2023.
- [118] Shuai Li, Yasin Abbasi-Yadkori, Branislav Kveton, S Muthukrishnan, Vishwa Vinay, and Zheng Wen. Offline Evaluation of Ranking Policies with Click Models. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1685–1694. ACM, 2018.
- [119] Yen-Chieh Lien, Daniel Cohen, and W. Bruce Croft. An Assumption-Free Approach to the Dynamic Truncation of Ranked Lists. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 79–82. ACM, 2019.
- [120] Yew Jin Lim and Yee Whye Teh. Variational Bayesian Approach to Movie Rating Prediction. In *Proceedings of KDD Cup and Workshop*, volume 7, pages 15–21, 2007.
- [121] Roderick J. A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, 2019.
- [122] Tie-Yan Liu. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.
- [123] Yiming Liu, Xuezhi Cao, and Yong Yu. Are You Influenced by Others When Rating?: Improve Rating Prediction by Conformity Modeling. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 269–272, 2016.

- [124] M. Loève. Probability Concepts. In *Probability Theory I*, pages 151–176. Springer, 1977.
- [125] Ben London and Ted Sandler. Bayesian Counterfactual Risk Minimization. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4125–4133. PMLR, 2019.
- [126] Lori Lorigo, Maya Haridasan, Hrönn Brynjarsdóttir, Ling Xia, Thorsten Joachims, Geri Gay, Laura Granka, Fabio Pellacini, and Bing Pan. Eye Tracking and Online Search: Lessons Learned and Challenges Ahead. *Journal of the American Society for Information Science and Technology*, 59(7):1041–1052, 2008. ISSN 1532-2890.
- [127] R. Duncan Luce. The Choice Axiom after Twenty Years. *Journal of Mathematical Psychology*, 15(3):215–233, 1977. ISSN 1096-0880.
- [128] Jiaqi Ma, Xinyang Yi, Weijing Tang, Zhe Zhao, Lichan Hong, Ed Chi, and Qiaozhu Mei. Learning-to-Rank with Partitioned Preference: Fast Estimation for the Plackett-Luce Model. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pages 928–936. PMLR, 2021.
- [129] Haitao Mao, Lixin Zou, Yujia Zheng, Jiliang Tang, Xiaokai Chu, Jiashu Zhao, Qian Wang, and Dawei Yin. Whole Page Unbiased Learning to Rank. In *Proceedings of the ACM Web Conference 2024*, pages 1431–1440. ACM, 2024.
- [130] Mari-Carmen Marcos, Ferran Gavin, and Ioannis Arapakis. Effect of Snippets on User Experience in Web Search. In *Proceedings of the XVI International Conference on Human Computer Interaction*, pages 1–8. ACM, 2015.
- [131] Benjamin Marlin. *Collaborative Filtering: A Machine Learning Perspective*. Graduate Thesis, University of Toronto, 2004.
- [132] Benjamin M. Marlin and Richard S Zemel. Collaborative Prediction and Ranking with Non-Random Missing Data. In *Proceedings of the Third ACM Conference on Recommender Systems*, pages 5–12, 2009.
- [133] David Maxwell, Leif Azzopardi, and Yashar Moshfeghi. A Study of Snippet Length and Informativeness: Behaviour, Performance and User Experience. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 135–144. ACM, 2017.

- [134] Maciej A. Mazurowski. Estimating Confidence of Individual Rating Predictions in Collaborative Filtering Recommender Systems. *Expert Systems with Applications*, 40(10):3847–3857, 2013. ISSN 0957-4174.
- [135] Sean McNee, Shyong Lam, Catherine Guetzlaff, Joseph Konstan, and John Riedl. Confidence Displays and Training in Recommender Systems. In *Proceedings of Human-Computer Interaction (INTERACT)*, 2003.
- [136] Rus M. Mestas and Alejandro Bellogín. Exploiting Recommendation Confidence in Decision-Aware Recommender Systems. *Journal of Intelligent Information Systems*, 54(1):45–78, 2020. ISSN 0925-9902, 1573-7675.
- [137] Andriy Mnih and Russ R Salakhutdinov. Probabilistic Matrix Factorization. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.
- [138] Lillio Mok, Samuel F. Way, Lucas Maystre, and Ashton Anderson. The Dynamics of Exploration on Spotify. *Proceedings of the International AAAI Conference on Web and Social Media*, 16:663–674, 2022. ISSN 2334-0770.
- [139] Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-Aware Learning for Deep Neural Networks. In *International Conference on Machine Learning*, pages 7034–7044. PMLR, 2020.
- [140] Yusuke Narita, Shota Yasui, and Kohei Yata. Debiased Off-Policy Evaluation for Recommendation Systems. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 372–379. Association for Computing Machinery, 2021.
- [141] Allen Nie, Yash Chandak, Christina J Yuan, Anirudhan Badrinath, Yannis Flet-Berliac, and Emma Brunskill. Opera: Automatic offline policy evaluation with re-weighted aggregates of multiple estimators. *Advances in Neural Information Processing Systems*, 37: 103652–103680, 2024.
- [142] Harrie Oosterhuis. *Learning from User Interactions with Rankings: A Unification of the Field*. PhD thesis, Informatics Institute, University of Amsterdam, 2020.

- [143] Harrie Oosterhuis. Computationally Efficient Optimization of Plackett-Luce Ranking Models for Relevance and Fairness. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1023–1032. ACM, 2021.
- [144] Harrie Oosterhuis. Learning-to-Rank at the Speed of Sampling: Plackett-Luce Gradient Estimation with Minimal Computational Complexity. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2266–2271. ACM, 2022.
- [145] Harrie Oosterhuis and Maarten de Rijke. Policy-Aware Unbiased Learning to Rank for Top-k Rankings. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 489–498, 2020.
- [146] Harrie Oosterhuis and Maarten de Rijke. Unifying Online and Counterfactual Learning to Rank. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. ACM, 2021.
- [147] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. Correcting for Selection Bias in Learning-to-rank Systems. In *Proceedings of The Web Conference 2020*, pages 1863–1873, 2020.
- [148] Iván Palomares, Carlos Porcel, Luiz Pizzato, Ido Guy, and Enrique Herrera-Viedma. Reciprocal Recommender Systems: Analysis of state-of-art literature, challenges and opportunities towards social recommendation. *Information Fusion*, 69:103–127, 2021. ISSN 1566-2535.
- [149] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

- [150] Kanaad Pathak, Leif Azzopardi, and Martin Halvey. The Influence of Presentation and Performance on User Satisfaction. In *Proceedings of the 2024 Conference on Human Information Interaction and Retrieval*, 2024.
- [151] Kanaad Pathak, Leif Azzopardi, and Martin Halvey. Ranking Heterogeneous Search Result Pages using the Interactive Probability Ranking Principle. In *European Conference on Information Retrieval*, 2024.
- [152] Ladislav Peska and Stepan Balcar. The Effect of Feedback Granularity on Recommender Systems Performance. In *Sixteenth ACM Conference on Recommender Systems*, pages 586–591. ACM, 2022.
- [153] Aleksandr Vladimirovich Petrov and Craig Macdonald. gSASRec: Reducing Overconfidence in Sequential Recommendation Trained with Negative Sampling. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 116–128. Association for Computing Machinery, 2023.
- [154] Robin L Plackett. The Analysis of Permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2):193–202, 1975.
- [155] Bruno Pradel, Nicolas Usunier, and Patrick Gallinari. Ranking with Non-Random Missing Ratings: Influence of Popularity and Positivity on Evaluation Metrics. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, pages 147–154, 2012.
- [156] Tao Qin and Tie-Yan Liu. Introducing LETOR 4.0 Datasets. *arXiv preprint arXiv:1306.2597*, 2013.
- [157] Zhen Qin, Honglei Zhuang, Rolf Jagerman, Xinyu Qian, Po Hu, Dan Chary Chen, Xuanhui Wang, Michael Bendersky, and Marc Najork. Bootstrapping Recommendations at Chrome Web Store. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3483–3491, 2021.
- [158] Ramsey M. Raafat, Nick Chater, and Chris Frith. Herding in Humans. *Trends in Cognitive Sciences*, 13(10):420–428, 2009.
- [159] Shaina Raza and Chen Ding. News Recommender System: A Review of Recent Progress, Challenges, and Opportunities. *Artificial Intelligence Review*, 55(1):749–800, 2022. ISSN 1573-7462.

- [160] Amy Rechkemmer and Ming Yin. When Confidence Meets Accuracy: Exploring the Effects of Multiple Performance Indicators on Trust in Machine Learning Models. In *CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2022.
- [161] James Reilly, Barry Smyth, Lorraine McGinty, and Kevin McCarthy. Critiquing with Confidence. In *Case-Based Reasoning Research and Development*, pages 436–450. Springer, 2005.
- [162] Steffen Rendle, Li Zhang, and Yehuda Koren. On the Difficulty of Evaluating Baselines: A Study on Recommender Systems. *arXiv preprint arXiv:1905.01395*, 2019.
- [163] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. Neural Collaborative Filtering vs. Matrix Factorization Revisited. In *Fourteenth ACM Conference on Recommender Systems*, pages 240–248. ACM, 2020.
- [164] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender Systems: Introduction and Challenges. In *Recommender Systems Handbook*. Springer, 2015.
- [165] S.E. Robertson. The Probability Ranking Principle in IR. *Journal of Documentation*, 33(4):294–304, 1977. ISSN 0022-0418.
- [166] James Robinson-Cox and Robert Boik. Derivatives of the Incomplete Beta Function. *Journal of Statistical Software*, 03, 1998.
- [167] Yuta Saito. Doubly Robust Estimator for Ranking Metrics with Post-Click Conversions. In *Fourteenth ACM Conference on Recommender Systems*, pages 92–100, 2020.
- [168] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. Unbiased Recommender Learning from Missing-Not-At-Random Implicit Feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 501–509, 2020.
- [169] Ruslan Salakhutdinov and Andriy Mnih. Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo. In *Proceedings of the 25th International Conference on Machine Learning*, pages 880–887. Association for Computing Machinery, 2008.

- [170] Matthew J. Salganik and Duncan J. Watts. Leading the Herd Astray: An Experimental Study of Self-fulfilling Prophecies in an Artificial Cultural Market. *Social Psychology Quarterly*, 71(4):338–355, 2008.
- [171] Matthew J. Salganik, Peter S. Dodds, and Duncan J. Watts. Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market. *Science*, 311(5762):854–856, 2006.
- [172] Mark Sanderson. Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends® in Information Retrieval*, 4(4):247–375, 2010. ISSN 1554-0669, 1554-0677.
- [173] Carlo Sansone and Giancarlo Sperlí. Legal Information Retrieval Systems: State-of-the-art and Open Issues. *Information Systems*, 106: 101967, 2022. ISSN 0306-4379.
- [174] Masahiro Sato, Sho Takemori, Janmajay Singh, and Tomoko Ohkuma. Unbiased Learning for the Causal Effect of Recommendation. *Fourteenth ACM Conference on Recommender Systems*, pages 378–387, 2020.
- [175] Masahiro Sato, Janmajay Singh, Sho Takemori, and Qian Zhang. Causality-Aware Neighborhood Methods for Recommender Systems. In *European Conference on Information Retrieval*, pages 603–618, 2021.
- [176] Brennan Schaffner, Antonia Stefanescu, Olivia Campili, and Marshini Chetty. Don’t Let Netflix Drive the Bus: User’s Sense of Agency Over Time and Content Choice on Netflix. *Proceedings of the ACM on Human-Computer Interaction*, 7(CSCW1):1–32, 2023. ISSN 2573-0142.
- [177] Linda Schamber. Relevance and Information Behavior. *Annual Review of Information Science and Technology (ARIST)*, 29:3–48, 1994. ISSN 0066-4200.
- [178] 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*, pages 1670–1679, 2016.

- [179] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. AutoRec: Autoencoders Meet Collaborative Filtering. In *Proceedings of the 24th International Conference on World Wide Web*, pages 111–112. ACM, 2015.
- [180] Upendra Shardanand and Pattie Maes. Social Information Filtering: Algorithms for Automating “Word of Mouth”. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '95*, pages 210–217. ACM Press, 1995.
- [181] Nicolás Silva, Heitor Werneck, Thiago Silva, Adriano C. M. Pereira, and Leonardo Rocha. Multi-Armed Bandits in Recommendation Systems: A Survey of the State-of-the-Art and Future Directions. *Expert Systems with Applications*, 197:116669, 2022. ISSN 0957-4174.
- [182] Herbert A. Simon. Bandwagon and Underdog Effects and the Possibility of Election Predictions. *The Public Opinion Quarterly*, 18(3): 245–253, 1954.
- [183] 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.
- [184] Ashudeep Singh and Thorsten Joachims. Policy Learning for Fairness in Ranking. In *Advances in Neural Information Processing Systems*, pages 5426–5436, 2019.
- [185] Brent Smith and Greg Linden. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Computing*, 21(3):12–18, 2017. ISSN 1941-0131.
- [186] Ian Soboroff. Overview of TREC 2021. *NIST*, 2022.
- [187] Giuseppe Spillo, Allegra De Filippo, Cataldo Musto, Michela Milano, and Giovanni Semeraro. Comparing Data Reduction Strategies for Energy-Efficient Green Recommender Systems. *Journal of Intelligent Information Systems*, 2025. ISSN 1573-7675.
- [188] Harald Steck. Item Popularity and Recommendation Accuracy. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, pages 125–132, 2011.

- [189] Harald Steck. Evaluation of Recommendations: Rating-Prediction and Ranking. In *Proceedings of the Seventh ACM Conference on Recommender Systems*, pages 213–220, 2013.
- [190] Alexander L. Strehl, John Langford, Lihong Li, and Sham M. Kakade. Learning from Logged Implicit Exploration Data. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, pages 2217–2225, 2010.
- [191] Xiaoyuan Su and Taghi M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009.
- [192] Zhongxiang Sun, Zihua Si, Xiaoxue Zang, Dewei Leng, Yanan Niu, Yang Song, Xiao Zhang, and Jun Xu. KuaiSAR: A Unified Search And Recommendation Dataset. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 5407–5411. ACM, 2023.
- [193] S. Shyam Sundar, Anne Oeldorf-Hirsch, and Qian Xu. The Bandwagon Effect of Collaborative Filtering Technology. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, pages 3453–3458, 2008.
- [194] Bernard Surlemont and Colin Johnson. The Role of Guides in Artistic Industries: The Special Case of the “Star System” in the Haute-cuisine Sector. *Managing Service Quality*, 15(6):577–590, 2005. ISSN 0960-4529.
- [195] Adith Swaminathan and Thorsten Joachims. Counterfactual Risk Minimization: Learning from Logged Bandit Feedback. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 814–823, 2015.
- [196] Adith Swaminathan and Thorsten Joachims. The Self-Normalized Estimator for Counterfactual Learning. In *Advances in Neural Information Processing Systems*, volume 28, pages 3231–3239, 2015.
- [197] Simon Sweeney and Fabio Crestani. Effective Search Results Summary Size and Device Screen Size: Is There a Relationship? *Information Processing & Management*, 42(4):1056–1074, 2006. ISSN 0306-4573.

- [198] Duyu Tang, Bing Qin, Ting Liu, and Yuekui Yang. User Modeling with Neural Network for Review Rating Prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1340–1346, 2015.
- [199] I.J. Thompson and A.R. Barnett. Coulomb and Bessel Functions of Complex Arguments and Order. *Journal of Computational Physics*, 64(2):490–509, 1986. ISSN 00219991.
- [200] Daria Tikhonovich, Nikita Zelinskiy, Aleksandr V. Petrov, Mayya Spirina, Andrei Semenov, Andrey V. Savchenko, and Sergei Kuliev. eSASRec: Enhancing Transformer-based Recommendations in a Modular Fashion. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pages 1175–1180. Association for Computing Machinery, 2025.
- [201] Lloyd N. Trefethen. Exactness of Quadrature Formulas. *SIAM Review*, 64(1):132–150, 2022. ISSN 0036-1445, 1095-7200.
- [202] Takuma Udagawa, Haruka Kiyohara, Yusuke Narita, Yuta Saito, and Kei Tateno. Policy-Adaptive Estimator Selection for Off-Policy Evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):10025–10033, 2023. ISSN 2374-3468.
- [203] Aleksei Ustimenko and Liudmila Prokhorenkova. StochasticRank: Global Optimization of Scale-Free Discrete Functions. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9669–9679. PMLR, 2020.
- [204] Ali Vardasbi, Maarten de Rijke, and Ilya Markov. Cascade Model-Based Propensity Estimation for Counterfactual Learning to Rank. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2089–2092, 2020.
- [205] Chao Wang, Qi Liu, Runze Wu, Enhong Chen, Chuanren Liu, Xunpeng Huang, and Zhenya Huang. Confidence-Aware Matrix Factorization for Recommender Systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018. ISSN 2374-3468, 2159-5399.
- [206] Ting Wang and Dashun Wang. Why Amazon’s Ratings Might Mislead You: The Story of Herding Effects. *Big Data*, 2(4):196–204, 2014. ISSN 2167-6461.

- [207] Ting Wang, Dashun Wang, and Fei Wang. Quantifying Herding Effects in Crowd Wisdom. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1087–1096, August 2014.
- [208] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. Doubly Robust Joint Learning for Recommendation on Data Missing Not at Random. In *International Conference on Machine Learning*, pages 6638–6647, 2019.
- [209] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. Learning to Rank with Selection Bias in Personal Search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 115–124. ACM, 2016.
- [210] 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.
- [211] Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. The LambdaLoss Framework for Ranking Metric Optimization. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1313–1322. ACM, 2018.
- [212] Yixin Wang, Dawen Liang, Laurent Charlin, and David M. Blei. Causal Inference for Recommender Systems. In *Fourteenth ACM Conference on Recommender Systems*, pages 426–431, 2020.
- [213] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. Model-Agnostic Counterfactual Reasoning for Eliminating Popularity Bias in Recommender System. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1791–1800, 2021.
- [214] Ryen W. White. *Interactions with Search Systems*. Cambridge University Press, 2016.
- [215] Dominik Wied and Rafael Weißbach. Consistency of the Kernel Density Estimator: A Survey. *Statistical Papers*, 53(1):1–21, 2012. ISSN 0932-5026, 1613-9798.

- [216] Ronald J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8(3): 229–256, 1992. ISSN 1573-0565.
- [217] 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.
- [218] Hong Xie and Mingze Zhong. Robust Product Rating Rules Against Herding Effects: Theory and Applications. In *2020 IEEE International Conference on Data Mining*, pages 1352–1357, 2020.
- [219] Hong Xie, Yongkun Li, and John C. S. Lui. Understanding Persuasion Cascades in Online Product Rating Systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5490–5497, 2019.
- [220] Hong Xie, Mingze Zhong, Yongkun Li, and John C. S. Lui. Understanding Persuasion Cascades in Online Product Rating Systems: Modeling, Analysis, and Inference. *ACM Transactions on Knowledge Discovery from Data*, 15(3):1–29, 2021.
- [221] Emine Yilmaz, Evangelos Kanoulas, and Javed A. Aslam. A simple and efficient sampling method for estimating AP and NDCG. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 603–610. ACM, 2008.
- [222] Sheng You, Chao Wang, Baohua Wu, Jingping Liu, Quan Lu, Guanzhou Han, and Yanghua Xiao. What Image do You Need? A Two-stage Framework for Image Selection in E-commerce. In *Companion Proceedings of the ACM Web Conference 2023*, pages 452–456. ACM, 2023.
- [223] Yang Yu. Towards Sample Efficient Reinforcement Learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5739–5743. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/820.

- [224] Bowen Yuan, Yaxu Liu, Jui-Yang Hsia, Zhenhua Dong, and Chih-Jen Lin. Unbiased Ad Click Prediction for Position-aware Advertising Systems. In *Fourteenth ACM Conference on Recommender Systems*, pages 368–377, 2020.
- [225] Yisong Yue, Rajan Patel, and Hein Roehrig. Beyond Position Bias: Examining Result Attractiveness as a Source of Presentation Bias in Clickthrough Data. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1011–1018. ACM, 2010.
- [226] Eva Zangerle and Christine Bauer. Evaluating Recommender Systems: Survey and Framework. *ACM Computing Surveys*, 55(8):170:1–170:38, 2022. ISSN 0360-0300.
- [227] Mingyue Zhang, Xunhua Guo, and Guoqing Chen. Prediction Uncertainty in Collaborative Filtering: Enhancing Personalized Online Product Ranking. *Decision Support Systems*, 83:10–21, 2016. ISSN 0167-9236.
- [228] Qianru Zhang, Peng Yang, Junliang Yu, Haixin Wang, Xingwei He, Siu-Ming Yiu, and Hongzhi Yin. A Survey on Point-of-Interest Recommendation: Models, Architectures, and Security. *IEEE Transactions on Knowledge and Data Engineering*, 37(6):3153–3172, 2025. ISSN 1558-2191.
- [229] Xiaoying Zhang, Junzhou Zhao, and John C.S. Lui. Modeling the Assimilation-Contrast Effects in Online Product Rating Systems: Debiasing and Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 98–106, 2017.
- [230] Xiaoying Zhang, Hong Xie, Junzhou Zhao, and John C. S. Lui. Understanding Assimilation-Contrast Effects in Online Rating Systems: Modelling, Debiasing, and Applications. *ACM Transactions on Information Systems*, 38(1):1–25, 2020.
- [231] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. Causal Intervention for Leveraging Popularity Bias in Recommendation. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 11–20, 2021.
- [232] Yongfeng Zhang, Xu Chen, et al. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020.

- [233] Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. Bayesian CP Factorization of Incomplete Tensors with Automatic Rank Determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1751–1763, 2015. ISSN 1939-3539.
- [234] Mingze Zhong, Hong Xie, and Qingsheng Zhu. Quantifying Assimilate-Contrast Effects in Online Rating Systems: Modeling, Analysis and Application. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2369–2377, 2021.
- [235] Honglei Zhuang, Zhen Qin, Xuanhui Wang, Michael Bendersky, Xinyu Qian, Po Hu, and Dan Chary Chen. Cross-Positional Attention for Debiasing Clicks. In *Proceedings of the Web Conference 2021*, pages 788–797, 2021.
- [236] Lixin Zou, Haitao Mao, Xiaokai Chu, Jiliang Tang, Wenwen Ye, Shuaiqiang Wang, and Dawei Yin. A Large Scale Search Dataset for Unbiased Learning to Rank. *Advances in Neural Information Processing Systems*, 35:1127–1139, 2022.

Appendix A

Research Data Management

This thesis research has been carried out under the research data management policy of the Institute for Computing and Information Science of the Radboud University, the Netherlands.¹

The following research datasets have been produced during this PhD research:

- Chapter 2: Knyazev, ir. N. (RU Radboud University); Oosterhuis, dr. H.R. (RU Radboud University): Code for "The Bandwagon Effect: Not Just Another Bias".
Zenodo. <https://doi.org/10.5281/zenodo.17351148>
- Chapter 3: Knyazev, ir. N. (RU Radboud University); Oosterhuis, dr. H.R. (RU Radboud University): Code for "Sample-Free Almost-Exact Estimation of Plackett-Luce Propensities for Off-Policy Ranking".
Radboud Data Repository.
Zenodo. <https://doi.org/10.5281/zenodo.17352314>
- Chapter 4: Knyazev, ir. N. (RU Radboud University); Oosterhuis, dr. H.R. (RU Radboud University): Code for "Learning to Rank with Variable Result Presentation Lengths".
Radboud Data Repository.
Zenodo. <https://doi.org/10.5281/zenodo.17352329>
- Chapter 5: Knyazev, ir. N. (RU Radboud University); Oosterhuis, dr. H.R. (RU Radboud University): Code for "A Lightweight Method for Modeling Confidence in Recommendations with Learned Beta Distributions".
Radboud Data Repository.
Zenodo <https://doi.org/10.5281/zenodo.17352353>

¹<https://www.ru.nl/rdm/>, last accessed October 10th, 2025

About the Author

Norman Knyazev began his academic career by obtaining a bachelor's degree in Genetics from the University of Manchester. This was followed by a master's degree in Computer Science from TU Delft, with his thesis investigating modeling of user journeys in sequential recommender systems. Building on this earlier work, Norman then commenced his PhD studies at the Institute for Computing and Information Sciences at Radboud University, with a focus on ranking and recommendation systems that learn from user interactions. His scientific work combines statistical approaches with practical applicability in order to provide efficient solutions to complex large-scale real-world problems.

Summary

Evaluation and optimization of ranking and recommendation systems involves estimating various quantities such as relevance, gradients or probabilities. Importantly, different estimation procedures may vary in terms of their statistical and computational properties, which can in turn significantly affect the downstream performance for a given task. In this thesis, we study ranking and recommendation estimation methods with a focus on some of their key properties: bias, variance, consistency and efficiency.

We first explore the bandwagon effect – a cognitive bias whereby users’ ratings may be affected by those of other users. Through our theoretical analysis we investigate whether the bandwagon effect is a problem of statistical bias, or is better described in terms of other statistical properties, and propose methods to reduce its impact on relevance estimation.

In the second research chapter, we focus on estimation of document placement probabilities under the Plackett-Luce (PL) ranking model for off-policy evaluation (OPE). To this end, we propose a novel, accurate and sample- and computationally efficient estimation method by leveraging the connection of the PL ranking model to Gumbel and Poisson binomial distributions, allowing for more data-efficient and lower-variance OPE.

In the third research chapter, we introduce the variable presentation length setting, whereby document order and presentation lengths have to be decided jointly. We explore the theoretical implications and the unique challenges of this setting, and propose a highly sample- and computationally efficient extension of the PL ranking model tailored to this task.

Our final research chapter looks at the use of the variance over the predicted ratings to represent model confidence. We employ the beta distribution to jointly model user preference and prediction confidence, avoiding significant computational complexity or reduced accuracy on the original rating prediction problem, whilst simultaneously allowing us to successfully leverage the learned confidence values for downstream use.

In its entirety, this thesis demonstrates that statistical and computational properties of estimators play a significant role in learning from user interactions. It highlights their importance across a wide range of settings and provides actionable guidance for developing estimation methods for ranking and recommendation that address bias, variance, consistency and efficiency.

Samenvatting

Bij de evaluatie en optimalisatie van rangschikkings- en aanbevelingssystemen moeten diverse grootheden worden geschat, zoals relevantie, gradiënten of waarschijnlijkheden. Belangrijk is dat verschillende schattingsprocedures kunnen verschillen qua statistische en computationele eigenschappen, wat op zijn beurt een aanzienlijke invloed kan hebben op de prestaties bij vervolgstappen voor een bepaalde taak. In dit proefschrift bestuderen we schattingsmethoden voor rangschikking en aanbevelingen, waarbij we ons richten op enkele van hun belangrijkste eigenschappen: vertekening, variantie, consistentie en efficiëntie.

We onderzoeken eerst het bandwagon-effect — een cognitieve vertekening waarbij de beoordelingen van gebruikers kunnen worden beïnvloed door die van andere gebruikers. Aan de hand van onze theoretische analyse gaan we na of het bandwagon-effect een probleem van statistische vertekening is, of dat het beter kan worden beschreven aan de hand van andere statistische eigenschappen, en stellen we methoden voor om de invloed ervan op de relevantieschatting te verminderen.

In het tweede onderzoekshoofdstuk richten we ons op het schatten van de waarschijnlijkheden voor de plaatsing van documenten binnen het Plackett-Luce (PL)-rangschikkingsmodel voor off-policy evaluation (OPE). Daartoe stellen we een nieuwe, nauwkeurige en zowel qua steekproefomvang als qua rekenkracht efficiënte schattingsmethode voor, waarbij we gebruikmaken van het verband tussen het PL-rangschikkingsmodel en de Gumbel- en Poisson-binomiale verdelingen, wat een meer data-efficiënte OPE met een lagere variantie mogelijk maakt.

In het derde onderzoekshoofdstuk introduceren we de instelling met variabele presentatielengte, waarbij de volgorde van de documenten en de presentatielengtes gezamenlijk moeten worden bepaald. We onderzoeken de theoretische implicaties en de specifieke uitdagingen van deze instelling, en stellen een zeer steekproef- en rekenkracht-efficiënte uitbreiding voor van het PL-rangschikkingsmodel, die speciaal op deze taak is afgestemd.

In ons laatste onderzoekshoofdstuk bekijken we het gebruik van de variantie van de voorspelde beoordelingen om het vertrouwen in het model weer te geven. We gebruiken de bèta-verdeling om gebruikersvoorkeur en voorspellingsvertrouwen gezamenlijk te modelleren. Hierdoor vermijden we aanzienlijke rekencomplexiteit of een verminderde nauwkeurigheid bij het oorspronkelijke probleem van de beoordelingsvoorspelling, terwijl we

tegelijkertijd de geleerde vertrouwenswaarden succesvol kunnen benutten voor verder gebruik.

In zijn geheel toont dit proefschrift aan dat de statistische en computationele eigenschappen van schatters een belangrijke rol spelen bij het leren op basis van gebruikersinteracties. Het benadrukt het belang ervan in een breed scala aan toepassingen en biedt praktische richtlijnen voor het ontwikkelen van schattingsmethoden voor rangschikking en aanbevelingen, waarbij rekening wordt gehouden met vertekening, variantie, consistentie en efficiëntie.

