1-2022

New accurate, explainable, and unbiased machine learning models for recommendation with implicit feedback.

Khalil Damak
University of Louisville

NEW ACCURATE, EXPLAINABLE, AND UNBIASED MACHINE LEARNING MODELS FOR RECOMMENDATION WITH IMPLICIT FEEDBACK

By

Khalil Damak
M.Sc., Computer Science,
University of Louisville, Louisville, KY

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy in Computer Science and Engineering

Department of Computer Science and Engineering
University of Louisville
Louisville, Kentucky

May 2022

NEW ACCURATE, EXPLAINABLE, AND UNBIASED MACHINE LEARNING MODELS FOR RECOMMENDATION WITH IMPLICIT FEEDBACK

By

Khalil Damak
M.Sc., Computer Science,
University of Louisville, Louisville, KY

A Dissertation Approved On

April 18, 2022

by the following Dissertation Committee:

Dr. Olfa Nasraoui, Dissertation Director

Dr. Hichem Frigui

Dr. Nihat Altiparmak

Dr. Juw Won Park

Dr. Avery Kolers

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor Prof. Olfa Nasraoui for her endless guidance and support. I consider myself extremely fortunate to have had the opportunity to pursue my PhD degree under her guidance and learn from her expertise. She provided me with the perfect balance between freedom and guidance and taught me to be more ambitious, thorough, and creative to become a better scientist. Moreover, her kindness and encouragements have constantly pushed me to work harder and to become a better person overall.

I would like to extend my acknowledgements to my committee members: Dr. Hichem Frigui, Dr. Nihat Altiparmak, Dr. Juw Won Park, and Dr. Avery Kolers for accepting to serve in my dissertation committee and for their valuable feedback and advice.

I would like to express my deep thanks to the Computer Science Department for offering me the opportunity to pursue my graduate studies and for the financial support. I would also like to acknowledge the National Science Foundation for partially supporting my research through grants IIS-1549981 and CNS-1828521.

I address my sincerest thanks to my lab-mates at the Knowledge Discovery & Web Mining lab for their support, motivation, and friendship. Special thanks also to our lab's alumna, Dr. Esin Saka, for her valuable feedback on the presentation. I would like to thank Dr. Gina Bertocci and Dr. Karen Bertocci for their valuable co-mentoring alongside my advisor, for their encouragement and trust, and for their support during my first year at the University of Louisville, while working on a collaborative interdisciplinary project with our lab.

Last but by no means least, I would never forget the motivation I had from my

family and friends. I especially would like to express my deepest gratitude to my parents, my brother, and my beloved one, Nada, for their selfless and unconditional support. Also, I would like to thank my cat Farhood for his company during the late work nights.

ABSTRACT

NEW ACCURATE, EXPLAINABLE, AND UNBIASED MACHINE LEARNING MODELS FOR RECOMMENDATION WITH IMPLICIT FEEDBACK

Khalil Damak

April 18, 2022

Recommender systems have become ubiquitous Artificial Intelligence (AI) tools that play an important role in filtering online information in our daily lives. Whether we are shopping, browsing movies, or listening to music online, AI recommender systems are working behind the scene to provide us with curated and personalized content, that has been predicted to be relevant to our interest. The increasing prevalence of recommender systems has challenged researchers to develop powerful algorithms that can deliver recommendations with increasing accuracy. In addition to the predictive accuracy of recommender systems, recent research has also started paying attention to their fairness, in particular with regard to the bias and transparency of their predictions.

This dissertation contributes to advancing the state of the art in fairness in AI by proposing new Machine Learning models and algorithms that aim to improve the user's experience when receiving recommendations, with a focus that is positioned at the nexus of three objectives, namely accuracy, transparency, and unbiasedness of the predictions. In our research, we focus on state-of-the-art Collaborative Filtering (CF) recommendation approaches trained on implicit feedback data. More specifically, we address the limitations of two established deep learning approaches in two distinct recommendation settings, namely recommendation with user profiles and sequential recommendation.

First, we focus on a state of the art pairwise ranking model, namely Bayesian Personalized Ranking (BPR), which has been found to outperform pointwise models in predictive accuracy in the recommendation with the user profiles setting. Specifically, we address two limitations of BPR: (1) BPR is a black box model that does not explain its outputs, thus limiting the user's trust in the recommendations, and the analyst's ability to scrutinize a model's outputs; and (2) BPR is vulnerable to exposure bias due to the data being Missing Not At Random (MNAR). This exposure bias usually translates into an unfairness against the least popular items because they risk being under-exposed by the recommender system. We propose a novel explainable loss function and a corresponding model called Explainable Bayesian Personalized Ranking (EBPR) that generates recommendations along with item-based explanations. Then, we theoretically quantify the additional exposure bias resulting from the explainability, and use it as a basis to propose an unbiased estimator for the ideal EBPR loss. This being done, we perform an empirical study on three real-world benchmarking datasets that demonstrate the advantages of our proposed models, compared to existing state of the art techniques.

Next, we shift our attention to sequential recommendation systems and focus on modeling and mitigating exposure bias in BERT4Rec, which is a state-of-the-art recommendation approach based on bidirectional transformers. The bi-directional representation capacity in BERT4Rec is based on the Cloze task, a.k.a. Masked Language Model, which consists of predicting randomly masked items within the sequence, assuming that the true interacted item is the most relevant one. This results in an exposure bias, where non-interacted items with low exposure propensities are assumed to be irrelevant. Thus far, the most common approach to mitigating exposure bias in recommendation has been Inverse Propensity Scoring (IPS), which consists of down-weighting the interacted predictions in the loss function in proportion to their propensities of exposure, yielding a theoretically unbiased learning. We first argue and prove that IPS does not extend to sequential recommendation because it fails to account for the sequential nature of the problem. We then propose a novel propensity scoring mechanism, that we name Inverse Temporal Propensity Scoring

(ITPS), which is used to theoretically debias the Cloze task in sequential recommendation. We also rely on the ITPS framework to propose a bidirectional transformer-based model called ITPS-BERT4Rec. Finally, we empirically demonstrate the debiasing capabilities of our proposed approach and its robustness to the severity of exposure bias.

Our proposed explainable approach in recommendation with user profiles, EBPR, showed an increase in ranking accuracy of about 4% and an increase in explainability of about 7% over the baseline BPR model when performing experiments on real-world recommendation datasets. Moreover, experiments on a real-world unbiased dataset demonstrated the importance of coupling explainability and exposure debiasing in capturing the true preferences of the user with a significant improvement of 1% over the baseline unbiased model UBPR. Furthermore, coupling explainability with exposure debiasing was also empirically proven to mitigate popularity bias with an improvement in popularity debiasing metrics of over 10% on three real-world recommendation tasks over the unbiased UBPR model. These results demonstrate the viability of our proposed approaches in recommendation with user profiles and their capacity to improve the user's experience in recommendation by better capturing and modeling their true preferences, improving the explainability of the recommendations, and presenting them with more diverse recommendations that span a larger portion of the item catalog.

On the other hand, our proposed approach in sequential recommendation ITPS-BERT4Rec has demonstrated a significant increase of 1% in terms of modeling the true preferences of the user in a semi-synthetic setting over the state-of-the-art sequential recommendation model BERT4Rec while also being unbiased in terms of exposure. Similarly, ITPS-BERT4Rec showed an average increase of 8.7% over BERT4Rec in three real-world recommendation settings. Moreover, empirical experiments demonstrated the robustness of our proposed ITPS-BERT4Rec model to increasing levels of exposure bias and its stability in terms of variance. Furthermore, experiments on popularity debiasing showed a significant advantage of our proposed ITPS-BERT4Rec model for both the short and long term sequences. Finally, ITPS-BERT4Rec showed respective improvements of around 60%,

470%, and 150% over vanilla BERT4Rec in capturing the temporal dependencies between the items within the sequences of interactions for three different evaluation metrics. These results demonstrate the potential of our proposed unbiased estimator to improve the user experience in the context of sequential recommendation by presenting them with more accurate and diverse recommendations that better match their true preferences and the sequential dependencies between the recommended items.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Recommendation from implicit feedback has recently become the standard setting for training recommender systems thanks to the abundance of implicit feedback data [1], i.e. clicks, views, purchases, etc., compared to explicit feedback data such as ratings. Various recommendation approaches arose that aim to model the user's implicit feedback to provide them with accurate personalized recommendations that are specifically tailored to their needs. One notorious family of such approaches is Collaborative Filtering (CF). Collaborative filtering approaches generate recommendations by relying on the user feedback [2] solely without introducing any external information such as user or item metadata.

Recommender systems can also be categorized in the way the recommendation task is defined. For instance, some works treat the problem as a matrix completion problem. This task, called "recommendation with user profiles", assumes a static rating or interaction matrix of users by items, where the elements of the matrix represent the feedback of the user in the row for the item in the column. Given the existing feedback in the interaction matrix, the task is to predict the remaining missing elements. By predicting the feedback of all the users to all the items, recommendations can hence be inferred. On the other hand, some works rather aim to predict the next interaction in a sequence of interactions. The latter task is called "sequential recommendation".

In this work, we focus on the recommendation problem using collaborative filtering from implicit feedback in two distinct settings, namely, **(1)** the recommendation with user profiles; and **(2)** the recommendation with sequential implicit data. We introduce novel recommendation frameworks that are based on state-of-the-art recommendation approaches in both tasks and that aim to achieve the objectives of accuracy, explainability,

and unbiasedness.

In the following subsections, we introduce and motivate our proposed approaches in recommendation with user profiles and in sequential recommendation, respectively. The first approach aims to promote explainability and mitigate exposure bias in pairwise ranking-based recommendation [3] with user profiles. On the other hand, the second approach investigates and aims to mitigate exposure bias in sequential recommendation with bidirectional transformers [4, 5]. For the sake of clarity, we keep both approaches distinct and self contained. More particularly, each approach will be introduced with its own motivation and objectives. Moreover, each of the two approaches will be presented in its own self-contained chapter with its own notation, methodology and experimental results.

1.1 Debiased Explainable Pairwise Ranking from Implicit Feedback

Pairwise ranking, i.e. Bayesian Personalized Ranking (BPR) [3], is a collaborative filtering approach for recommendation with user profiles. BPR has recently received significant praise in the recommender systems community because of its capacity to rank implicit feedback data with high accuracy compared to pointwise models [6]. Aiming to rank relevant items higher than irrelevant items, pairwise ranking recommender systems often assume all non-interacted items as irrelevant. Hence, these systems rely on the assumption that implicit feedback data is Missing Completely At Random (MCAR), which means that the items are equally likely to be observed by the users [7], consequently any missing user-to-item interaction is missing because the user chose not to interact with it. However, given the abundance of items on most e-commerce, news, entertainment, social media, and other online platforms, it is safe to assume the impossibility of any user being exposed to *all* the items. Thus, missing interactions should be considered Missing Not At Random (MNAR). This means that the user may have been exposed to part of the items but chose not to interact with them, which can be a sign of irrelevance; and was not exposed to the rest of the items. This MNAR property is translated into an exposure bias. This type of bias is usually characterized by a bias against less popular items that have a lower

propensity of being observed [8].

Moreover, most accurate recommender systems tend to be black boxes that do not justify why or how an item was recommended to a user. This might engender unfairness issues if, for example, particularly inappropriate or offensive content gets recommended to a user. The effect of this kind of unfairness can be mitigated with an explanation. In fact, it could be important for the user to know why or how the inappropriate item was recommended. For example, an Italian user might think that the movie recommendation "The Godfather" is offensive because of the way it depicts, in a stereotypical way, a certain Italian community in the US. The explanation "Because you liked the movie "Scarface"" can be important in this case because it clarifies that the movie recommendation has nothing to do with their origins, but it was rather recommended because the user also liked another similar "mafia" movie. Furthermore, explanations in recommendation have been proved to help users make more accurate decisions, which translates into an increased user satisfaction [9, 10]. Bayesian Personalized Ranking (BPR) [3] treats comparisons between any positive and negative items the same, regardless of which ones can be, or cannot be explained. Thus, while BPR aptly captures and models ranking based preference, it does not yet capture an *explainable* preference. It is this explainable preference, in addition to an unbiased preference ranking, that we seek to achieve in this first approach.

1.1.1 Contributions to Pairwise Ranking-Based Recommendation from Implicit Feedback

We propose models that address explainability *and* exposure bias in pairwise ranking from implicit feedback and achieve the following contributions:

- Proposing an explainable loss function based on the state of the art Bayesian Personalized Ranking (BPR) loss [3] along with a corresponding Matrix Factorization (MF)-based model called Explainable Bayesian Personalized Ranking (EBPR). To the extent of our knowledge, no work has introduced neighborhood-based (or any other style of) explainability to pairwise ranking.

- Conducting a theoretical study of the additional exposure bias coming from the item-based explanations.

- Proposing an unbiased estimator for the ideal EBPR loss, called UEBPR, based on the Inverse Propensity Scoring (IPS) estimator [11]. To our knowledge, no prior work has tried to address the additional exposure bias that could result from neighborhood-based explainability.

- Performing an empirical study on three real-world datasets to compare the effectiveness of the proposed models, in terms of ranking, explainability, and both exposure and popularity debiasing.

- Investigating the properties of the proposed neighborhood based explainable models, revealing and explaining a desirable inherent popularity debiasing that is built into these models. This opens the path to a new family of future debiasing strategies, where the debiasing is rooted in an explainable neighborhood-based rationale.

1.2 Debiasing the Cloze Task in Sequential Recommendation with Bidirectional Transformers

Sequential recommendation is a recommendation setting in which the goal is to predict the next best interaction or interactions given a sequence of previous interactions through time [12]. Recent work that succeeded in modeling this sequential behaviour mostly relied on deep learning models including Recurrent Neural Networks (RNNs) [13–17], Convolutional Neural Networks (CNNs) [18, 19], and more recently, self-attention modules [4, 5, 20, 21]. Recent research has also addressed different biases in recommendation [8], in particular exposure bias. As was mentioned in section 1.1, exposure bias stems from the partial exposure of items to the users [8], making items with relatively low exposure often considered to be irrelevant. Ideally, recommender systems should capture the true relevance of the items to the users, regardless of their propensities of exposure. However, this is far from true on real life recommendation platforms. Exposure bias can be mitigated

during the training of recommender systems [8], mainly by making the models aware of the items' exposure propensities. One of the most common approaches consists of building propensity-weighted loss functions that are unbiased estimates of the desirable relevance-based objectives [11, 22]. This approach, called Inverse Propensity Scoring (IPS), showed success in recommendation settings with user profiles including our proposed approach which was mentioned in section 1.1, and which will be presented in Chapter 3.

Despite the progress in this area, to the extent of our knowledge, no previous work has addressed the problem of exposure bias in sequential recommendation.

1.2.1 Contributions to Sequential Recommendation with Bi-Directional Transformers

We propose new models and algorithms to mitigate exposure bias in bi-directional transformer-based recommender systems, which are considered state-of-the-art sequential recommender systems [4], and more specifically, the widely-used BERT4Rec model [4]. More broadly however, our work covers any sequential recommender system that is trained to optimize the *Cloze* task [5, 23]. Our contributions are summarized as follows:

- We theoretically formulate the problem of exposure bias in the Cloze task, and argue and prove that traditional Inverse Propensity Scoring (IPS) based debiasing frameworks do not extend to sequential recommendation.
- We propose an ideal Cloze task loss function that aims to capture the relevance of items within a sequence context.
- We propose a novel framework for debiasing the Cloze task in sequential recommendation, called Inverse Temporal Propensity Scoring (ITPS), and use it to propose a novel loss function that produces an unbiased estimator for the ideal Cloze task loss.
- We conduct experiments that demonstrate the debiasing capabilities of our ITPS-based estimator, and empirically validate our theoretically proven claims. Our experimental results show the advantages of our proposed approach in ranking, exposure

debiasing, popularity debiasing, and capturing the temporal dependencies between the items within the sequence.

1.3 Document Organization

In the following chapters, we start by reviewing related work in recommendation from implicit feedback in Chapter 2, including research on explainability and mitigating exposure bias in recommendation. Then, we present our proposed approaches which we briefly introduced in Sections 1.1 and 1.2, in Chapters 3 and 4, respectively, along with a comprehensive experimental evaluation in each chapter. Finally, we make our conclusions and discuss future work in Chapter 5.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

In this chapter, we start by reviewing recommender system work that is most related to our research. Particularly, we start by presenting the pairwise ranking problem and the Bayesian Personalized Ranking (BPR) recommendation approach. Then, we focus on the problem of sequential recommendation, followed by a review of transformer-based sequential recommender systems. This being done, we review previous work on explainability and counteracting exposure bias in recommendation. While it is impossible to do justice to every past contribution with an exhaustive review, we try to focus on the most representative or related work.

2.1 Collaborative Filtering for Recommendation with User Profiles

Collaborative Filtering (CF) is a family of recommendation techniques which rely solely on user-to-item interaction data to generate recommendations. Collaborative filtering recommendation approaches are based on the assumption that users who had interests that were aligned in the past, are likely to share similar interests in the future [24]. Hence, approaches in recommendation with user profiles often rely on similarity measures within the user-to-item interaction matrix to generate recommendations. More specifically, these approaches usually treat the problem as a matrix completion problem, where the goal is to predict the missing ratings or relevance scores (depending on the type of data available) [25]. Collaborative filtering techniques, in recommendation with user profiles, may be categorized into memory-based and model-based approaches depending on the techniques they rely on [25].

2.1.1 Memory-Based Collaborative Filtering

Memory-based CF approaches rely on similarities within the rating or interaction matrix to predict unobserved ratings [25]. Memory-based approaches can be classified as user-based or item-based depending on whether the similarities are considered between the users or the items. User-based CF approaches rely on similarities between users to generate the recommendations. More specifically, a typical user-based CF approach predicts a user's rating to an item by aggregating the ratings given to that item by similar users [26]. Other user-based approaches introduce fine-grained factors to weight the similar users [27], or rely on a recursive process that allows users that did not rate specific items that the target user rated to still be included in the prediction process [28]. Moreover, other approaches propose to refine the rating prediction by relying on a spectrum of user preferences [29]. On the other hand, item-based CF approaches rely on similarities to the items that the user has previously rated to recommend new unseen items [25, 30–32]. These methods usually rely on similarity measures such as the cosine [31] and adjusted cosine similarity [32].

2.1.2 Model-Based Collaborative Filtering

Model-based CF recommender systems rely on machine learning algorithms to generate recommendations [29]. These recommender systems are usually machine learning models that had been trained on the existing ratings or interactions, and then used to infer the missing interactions. Model-based recommender systems have seen a tremendous interest from the recommender systems research community over the recent decades. Numerous approaches have been proposed spanning a myriad of machine learning algorithms including clustering algorithms [33], matrix factorization [34], artificial neural networks [35–37], etc. Matrix Factorization (MF) [34] approaches have particularly attracted considerable attention thanks to their high prediction capabilities in addition to their advantages in terms of scalability [29]. Despite their simplicity, MF approaches have also recently been shown to outperform neural network based approaches [38]. Additionally, model-based CF recommendation approaches can also be categorized based on the task that they are trying

to achieve. Still, within the scope of matrix completion, three main families of tasks have been proposed for model-based collaborative filtering [39].

2.1.2.1 Pointwise Ranking

The first task is pointwise ranking. This approach trains the machine learning model to predict a rating or a relevance score of an item to a user. For instance, if the data is constituted of explicit feedback, i.e. ratings, then the task can be assimilated to a regression problem where the goal is to train the model to predict the training ratings of users to items [10, 40]. On the other hand, if the data consists of implicit feedback, i.e. clicks, views, purchases, etc., then the task is considered as a classification problem where the goal is to train the model to distinguish the relevant from irrelevant training items to the users [35]. However, unlike explicit feedback, implicit feedback data does not usually include negative feedback. In other words, only positive interactions are recorded [41], which makes the problem a binary classification problem with only the positive class available in the training data. To remedy to this issue, pointwise ranking from implicit feedback approaches usually either consider all non interacted items as negative interactions [42, 43], or rely on negative sampling where they sample some of the non-interacted items to constitute the negative class [35, 43, 44].

2.1.2.2 Pairwise Ranking

The second task that model-based CF recommender systems aim to achieve is pairwise ranking. The latter approach consists of training the model to rank positive interactions higher than negative interactions for a given user. The ranking is performed by comparing two items at a time for a user where the model is trained to rank a positive interaction higher than a negative interaction for the user [3]. Pairwise ranking from implicit feedback has received significant praise recently in the recommender systems community because of its empirical advantages over pointwise ranking in terms of prediction accuracy [6]. One might argue that the advantages of pairwise ranking are due to its ranking goal in the

training which aligns with the main objective of recommender systems, which is providing a relevant ranking of items for the user.

2.1.2.3 Listwise Ranking

Finally, the third approach consists of listwise ranking. This approach aims to model the preferences of each user to a list of items [39]. In other words, the goal is to learn the ranking of the training items for every user. Usually, the items need to be rated to have a ground truth ranking [39]. Thus, listwise ranking models often require explicit feedback data. A relatively few listwise ranking approaches have been proposed in recent years [45–47] compared to pointwise and pairwise approaches. This is probably due to their high computational complexity compared to other approaches.

Given the popularity of matrix factorization approaches and the empirically proven viability of pairwise ranking from implicit feedback, we opted to position our work in recommendation with user profiles within the scope of those approaches and aim to study and improve their limitations. More specifically, in our proposed approach for recommendation with user profiles, we rely on the Bayesian Personalized Ranking (BPR) loss and its corresponding MF-based model [3] which we present in detail in Section 2.2.

2.2 Bayesian Personalized Ranking for Pairwise Ranking

The Bayesian Personalized Ranking (BPR) loss was introduced in [3] as the first loss that is "optimized for ranking" in the implicit feedback pairwise ranking setting. In other words, it learns the users' preference of a positive item over a negative item. In this case, positive and negative items are those that the user has, respectively, interacted with and not interacted with. This is opposed to pointwise prediction, which can be seen as a predictive classification problem of the relevance of an item to a user as was explained in section 2.1.2.1. Pairwise ranking has received increasing attention and praise over the years from the recommender system research community due to its high performance in top-N recommendation compared to pointwise ranking [6]. The BPR objective function is defined

as follows:

$$L_{BPR} = \frac{1}{|D|} \sum_{(u,i_+,i_-)\in D} -log\sigma(f_\Omega(u, i_+, i_-)) \tag{2.1}$$

where $D = \{(u, i_+, i_-)|u\epsilon U, i_+\epsilon I_u^+, i_-\epsilon I_u^-\}$ is the training data. U is the set of users, I_u^+ is the set of positive (interacted) items by user u, and I_u^- is the set of negative (non-interacted) items by user u such that $I_u^- = I \setminus I_u^+$. f_Ω is a hypothesis with parameters Ω that quantifies how much user u prefers (following the order relation $>_u$ defined in [3]) item i_+ over item i_-, and σ is the Sigmoid function. When the BPR loss is applied to Matrix Factorization (MF) with the parameters Ω consisting of the respective user and item latent matrices $P\epsilon\mathbb{R}^{n\times K}$ and $Q\epsilon\mathbb{R}^{m\times K}$, the preference model is given by

$$f_\Omega(u, i_+, i_-) = P_u \cdot Q_{i_+}^T - P_u \cdot Q_{i_-}^T \tag{2.2}$$

Applying the Sigmoid function to the output of the preference model yields the preference probability, which is the probability of user u preferring item i_+ over item i_-:

$$P_\Omega(i_+ >_u i_-) = P(i_+ >_u i_-|\Omega) = \sigma(f_\Omega(u, i_+, i_-)) \tag{2.3}$$

Note that in equation 2.1, and in the remainder of this report, we omitted any regularization terms from the equations for simplicity, although we use L2 regularization in our implementation. Also note that the notation that was introduced in this section will be used when presenting our proposed approaches for recommendation with user profiles in Chapter 3.

2.3 Collaborative Filtering for Sequential Recommendation

Sequential recommendation is a recommendation setting in which the goal is to predict the next best interaction or basket of interactions given a sequence of previous interactions through time [12]. Sequential recommendation approaches can be categorized into three main families of approaches based on the tools that they rely on [48]. The first

family of approaches consists of conventional approaches which rely on conventional data mining and machine learning techniques [48] such as association rule mining [49–56], k-nearest neighbors [57–59], and Markov chains [60–65]. The second family of approaches is latent representation based approaches [48]. These approaches typically rely on latent factor models, such as matrix factorization [34], to represent or complete a transition matrix between items for every user [63,66–69]. Besides, other latent representation approaches rely on shallow neural networks to map interactions into a low dimensional latent space [70–72]. Finally, the third family of approaches relies on deep learning models to model sequential data and generate recommendations [48]. This family of approaches constitutes the standard in modeling sequential recommendation data today thanks to the capability of deep neural network to fit the complex intra- and inter-sequence dependencies [48]. Recent work that succeeded in modeling this sequential behaviour using deep learning models mostly relied on Recurrent Neural Networks [13–15] (RNNs) [16,17], Convolutional Neural Networks (CNNs) [18, 19], and more recently, self-attention modules, i.e. Transformers [4, 5, 20, 21, 73–75].

Given that Transformers have become a standard in modeling sequential data [76], and given the empirically demonstrated advantages that Transformer-based sequential recommender systems have recently exhibited [4, 21], we decided to rely on the state-of-the-art BERT4Rec [4] model as a starting point for our study on sequential recommendation. Thus, we perform a thorough review of sequential recommendation approaches with transformers in Section 2.4.

2.4 Sequential Recommendation with Transformers

Self-attention models, i.e. transformers, were initially proposed for machine translation [20]. Later, the encoder part of the transformer model was implemented and challenged in several Natural Language Processing (NLP) tasks to become the standard approach in modeling textual data [76], changing the landscape of the field [77]. Given the similarities between textual data and sequential recommendation data, transformer-based recommender systems have recently emerged and started dominating the field. To the extent of our

knowledge, SASRec [21] was one of the pioneering transformer-based approaches in sequential recommendation. This approach takes as input a dataset S of interactions comprised of $|S|$ sequences where each element S_s is a sequence of consecutive item interactions. Each input sequence S_s is first input into an embedding layer where a latent representation of every item within the sequence is obtained. These latent vectors are projections of the input items within the sequence into the latent space, which we denote as $E(S_s) \in \mathbb{R}^{T \times d}$, where T is the normalized number of time steps in the sequences and d is the dimensionality of the embedding matrix. This input embedding is later going to be input into a self-attention module [20]. However, the self-attention module does not incorporate any awareness of the temporal context of the interactions within the sequence [21], which may contradict the main goal of next interaction prediction. To address this issue, a positional encoding $P \in \mathbb{R}^{T \times d}$ is added to the sequence embedding, which injects some information about the relative or absolute position of the items within the sequence [20]. Thus, the final input embedding of sequence S_s, we denote as $F(S_s)$, is formulated as follows:

$$F(S_s) = E(S_s) + P \tag{2.4}$$

Note that the positional encoding P can either be learnable as used in [21] or fixed, for instance based on sine and cosine functions of different frequencies, as used in [20].

The final input embedding $F(S_s)$ of sequence S_s is then input into a self-attention block. The self-attention block [20] is constituted of multiple self-attention heads which outputs are aggregated. Assuming the self-attention block is constituted of H heads, then every head, for instance head h, includes three learnable weight matrices $W_h^Q \in \mathbb{R}^{d \times d_K}$, $W_h^K \in \mathbb{R}^{d \times d_K}$, and $W_h^V \in \mathbb{R}^{d \times d_V}$. The input embedding $F(S_s)$ is then projected on the three weight matrices to obtain the Query, Key, and Value matrices as follows:

$$Q_h = F(S_s) \cdot W_h^{Q^T},\ K_h = F(S_s) \cdot W_h^{K^T},\ and\ V_h = F(S_s) \cdot W_h^{V^T}. \tag{2.5}$$

This being done, the Query and Key matrices are combined to form the self-attention matrix [20] A_h as follows:

$$A_h = Softmax(\frac{Q_h \cdot K_h}{\sqrt{d_K}}). \tag{2.6}$$

The self-attention matrix A_h is a matrix in $[0,1]^{T \times T}$ in which the rows add-up to one. The intuition behind the self-attention matrix is that every value in the matrix represents the importance of the item on the column for the row item. The $\sqrt{d_K}$ in equation (2.6) is a scaling factor that aims to avoid the vanishing gradients problem [20].

Finally, the output of the attention head h, we denote $Head_h$, is obtained by multiplying the attention matrix A_h by the Value matrix V_h such that:

$$Head_h = A_h \cdot V_h. \tag{2.7}$$

$Head_h \in \mathbb{R}^{T \times d_V}$ is a new encoding of the input sequence S_s which is processed by the attention head. As we mentioned earlier, the self-attention block is usually constituted of multiple self-attention heads. Assuming there are H heads, the outputs of the multiple heads are then concatenated and multiplied by another weight matrix $W^o \in \mathbb{R}^{H \cdot d_V \times d_o}$ to obtain the multi-head attention output M such that:

$$M = [Head_1, ..., Head_H] \cdot W^o. \tag{2.8}$$

$[.,.]$ represents the concatenation operator. Finally, to introduce non-linearity into the model, the output of the multi-head attention module is input into a Feed-Forward Neural Network (FFN) [21]. If we represent the FNN with a function g_Ω with parameters Ω, then the final output O of the Transformer model is:

$$O = g_\Omega(M). \tag{2.9}$$

Note that self-attention blocks can be stacked to capture more complex patterns. Additionally, dropout, regularization, and layer normalization can be added to avoid overfitting [21].

To adapt self-attention modules to the sequential recommendation task of next interaction prediction, SASRec predicts a relevance score for each item i at each sequence S_s.

The predicted relevance score $\hat{y}_{S_s,i}$ is obtained through a dot product of the encoding of the last item at position T, O_T, and an item embedding vector N_i extracted from an item embedding matrix $N \in \mathbb{R}^{|I| \times d_o}$. The SASRec model [21] is trained with the Cross Entropy loss for the task of next item prediction, also called Causal Language Model (CLM).

Aside from CLM, which is an intuitive task that originated in Natural Language Processing (NLP), other more sophisticated NLP tasks were recently leveraged and adapted to work on sequential recommendation. These include Permutation Language Modeling (PLM) [75, 78], Replacement Token Detection (RTD) [75, 79], and Masked Language Modeling (MLM), a.k.a. Cloze Task [4,5]. Given that the Cloze task is the most established task in sequential recommendation so far, and given that it leads the state-of-the-art in several recommendation tasks [4, 75], we focus our interest towards it and rely on the BERT4Rec model as the backbone of our proposed approaches in sequential recommendation. We present a review of the BERT4Rec model in the following subsection.

2.4.1 BERT4Rec

BERT4Rec [4] has a similar transformer-based architecture as SASRec. However, it introduced bi-directionality when modeling the sequential recommendation data. The advantages of bi-directionality in modeling sequential data were first introduced in [5] in Natural Language Processing (NLP). BERT4Rec [4] relies on a similar methodology to propose a bi-directional transformer model for sequential recommendation. In fact, the bi-directionality is introduced through considering the problem as a Masked Language Model (MLM) [23], also called Cloze task, problem instead of a next interaction prediction, or CLM, problem. BERT4Rec showed a superior performance compared to state-of-the-art transformer-based and non-transformer-based approaches which justifies the viability of introducing the bi-directionality [4]. We will present a thorough review of the Cloze task used in BERT4Rec later in Section 4.1.2.

2.5 Explainability in Recommendation

The types of explanations in recommendation have varied with the type of data used. In fact, some explanations are content-based, meaning that they usually come from user or item features such as reviews, tags or product images. These were used in previous works which employed sentiment analysis on user review data along with learned latent features to generate explanations to recommendations in the form of user or item features [80], textual sentences [80] or word clusters [81]. Other research efforts used attention mechanisms to explain recommendations [82–85]. The generated explanations are important regions in the textual [84] or image [82, 83, 85] inputs. Other methods relied on post-hoc approaches that try to extract explanations to the recommendations after they occur. For instance, [86] and [87] use influence functions to determine the effect of each input interaction on the recommendation. In contrast to the above methods, some explainable recommender systems rely solely on the feedback data such as ratings or interactions. Hence, they have the advantage of not requiring any additional content or metadata to generate the explanations. These explanations tend to depend only on the input rating data and they are mainly neighborhood-based, and can be either user-based or item-based [10, 88]. Explanations can be obtained by using classical, inherently interpretable (white box), user-based or item-based collaborative filtering techniques [88, 89] or by using model-based approaches, which are most related to our work. Among model-based approaches, Explainable Matrix Factorization (EMF) [10] pre-computes a user or item-based neighbor style explainability matrix from the ratings, and then uses this matrix in a regularization term that is added to obtain a new explainable recommendation reconstruction loss to guide the learning and yield explainable recommendations. This approach provides a simple and flexible way to add explainability to latent factor loss-based models to obtain a single integrated explainable model. It also has the advantage of not being a post-hoc approach, and hence not incurring the added cost of learning two separate models, nor risking lack of fidelity from deviations between the explaining model and the predictive model. For all these reasons, EMF was later adopted in several works, such as [90] which extended it and tried to im-

prove the novelty of the recommendations; and in [91] which modified the calculation of the explainability matrix by integrating the neighbors' weights to improve performance. Other works used influence functions to generate neighborhood-based explanations. This includes [92] which proposed a probabilistic factorization model, which employs an influence mechanism to evaluate the importance of the users' historical data and present the most related users and items as explanations to the predicted rating.

2.6 Exposure Bias in Recommendation with User Profiles

Bias in recommendation can be categorized into seven types [8] that occur within the various stages of the recommendation feedback loop between the user, the data, and the model. Among these categories, in the user-to-data phase, we find *exposure bias*, which is the focus of our work in this report. Exposure bias happens when users are only exposed to a portion of the items, and hence, unobserved interactions do not always represent negative preferences [8]. The techniques that were introduced to mitigate exposure bias, vary in whether they treat bias during the training or in the evaluation [8]. The common approach that is used in the evaluation phase incorporates an Inverse Propensity Scoring (IPS) modification of the ranking evaluation metrics, where more popular items are down-weighted and less popular items are up-weighted [93]. Exposure debiasing in the training is usually achieved by considering the unobserved interactions as negatives with a certain confidence [8]. These methods differ in the way they define or approximate the confidence weight. One group of methods [1, 94] considers a uniform weight for all negative items that is lower than one; while a second group [42, 43] utilizes the user activity, for instance the number of interacted items, to weight the negative interactions; and a third group uses item popularity [41, 95] and user-item similarity [96] to achieve a similar goal. Recent work [22] and [11] proposed IPS-based unbiased estimators for the ideal pointwise and pairwise losses, respectively. In their experiments, they estimated the propensity of an interaction using the relative item popularity. Departing from the previously mentioned methods, other work proposed negative sampling processes in order to mitigate exposure

bias. This negative sampling is usually done by exploiting side information such as social network information [97] or item-based knowledge graphs [98]. Another approach is to integrate the capacity to learn the exposure probability within the model [97,99,100], which in turn requires assumptions on the probability distribution of exposure. Finally, [101–104] consider users' sequential behaviors to address exposure bias with multi-task learning [8].

2.7 Exposure Bias in Sequential Recommendation

The aforementioned methods (Section 2.6) were originally proposed for recommendation with user profiles, where the goal is to recommend items to users regardless of the temporal context of the previous interactions. To the extent of our knowledge, no previous work has validated the applicability of these techniques in sequential recommendation. Furthermore, only a few studies [105, 106] have addressed exposure bias in sequential recommendation. However, these approaches treated sequential recommendation in a seq2seq adversarial setting, and use a different formulation of exposure bias which consists of a discrepancy between the training data distribution and the data distribution generated by the model [107], rather than a discrepancy between relevance and interaction.

2.8 Chapter Summary

In this chapter, we reviewed collaborative filtering approaches for two recommendation tasks, namely, recommendation with user profiles and sequential recommendation. In the recommendation with user profiles task, we briefly introduced the different types of existing approaches, then we focused on pairwise ranking, where we presented a through review of the Bayesian Personalized Ranking (BPR) recommender system. This being done, we shifted our interest to collaborative filtering recommender systems for sequential recommendation. We focused on the state-of-the-art sequential recommendation approaches that are based on transformers including the BERT4Rec model which we will be relying on in the following chapters. Finally, we reviewed the literature for approaches in explainability and mitigating exposure bias in recommendation. In the next two chapters, we will present

our proposed approaches that address gaps in existing work, namely optimizing the three objectives of accuracy, explainability, and exposure bias in both recommendation with user profiles and sequential recommendation, respectively.

CHAPTER 3

DEBIASED EXPLAINABLE PAIRWISE RANKING FROM IMPLICIT FEEDBACK

In this chapter, we propose new approaches to address both the need for explainability and exposure bias in the ranking-based recommendation from implicit feedback setting. We start by proposing the Explainable Bayesian Personalized Ranking (EBPR) loss. Then, we theoretically prove the presence of additional exposure bias resulting from the explainability term in the loss and propose an updated EBPR loss function that is unbiased for the ideal loss. This being done, we describe the empirical process we conducted to evaluate the effects of introducing the explainability and counteracting exposure bias. Finally, we tune and compare all the models we described in terms of ranking performance, explainability, and popularity debiasing, and show our results.

3.1 Explainable Bayesian Personalized Ranking

To the extent of our knowledge, no work has introduced neighborhood based explainability to pairwise ranking. More importantly, although neighborhood-based explainability can be expected to be vulnerable to exposure bias, there is a need to mitigate any additional exposure bias coming from the explainability. The BPR loss function (presented in Section 2.2 and formulated below) learns to rank positive (interacted) items by a user higher than any negative (non-interacted) item by that same user.

$$L_{BPR} = \frac{1}{|D|} \sum_{(u,i_+,i_-)\in D} -log\sigma(f_\Omega(u, i_+, i_-)) \tag{3.1}$$

This objective treats comparisons between any positive and negative items the same, regardless of which ones can be or cannot be explained based on any given style of explana-

tion, for instance based on neighborhoods. In other words, while BPR aptly captures and models a ranking based preference, it does not yet capture an *explainable* preference. In fact, as demonstrated in [10], it is important to consider the interpretability of the items to the users, often referred to as explainability, when learning a recommendation objective, and this can be computed based on readily available rating data, for instance from similar items. Hence, given a definition for a measure of explainability E_{ui}, of an item i to a user u, our aim is to condition the BPR objective function to capture what we call *explainable preference*. This means giving more importance to the explainable items that it is learning to rank higher, and less importance to the explainable items that it is learning to rank lower. In other words, if the objective function is learning to rank, for a user u, an item i_+ higher than an item i_-, then we we would additionally want to give an even higher importance to this preference if it is also accompanied by a higher explainability E_{ui_+} of item i_+ to user u and a lower explainability E_{ui_-} of item i_- to user u. We formulate this *explainable preference* desiderata into a modified objective to obtain Explainable Bayesian Personalized Ranking (EBPR) as follows:

Definition 1 (Explainable Bayesian Personalized Ranking (EBPR) Objective Function)**.** *Given an explainability matrix $E = (E_{ui})_{u=1..|U|,i=1..|I|} \in [0,1]^{|U|\times|I|}$, where E_{ui} is a measure of explainability of item i to user u, the EBPR objective function is defined as*

$$L_{EBPR} = \frac{1}{|D|} \sum_{(u,i_+,i_-)\in D} -E_{ui_+}(1 - E_{ui_-}) log\sigma(f_\Omega(u, i_+, i_-)) \tag{3.2}$$

We remind that f_Ω is a function with parameters Ω that quantifies the preference of a user u towards an item i_+ over an item i_-. More details about the notation used in this chapter can be found in section 2.2.

The intuition is to weight the contribution of an instance (u, i_+, i_-) into the loss by $E_{ui_+}(1 - E_{ui_-})$, in proportion to the degree that the positive item is considered to be more explainable and the negative item is considered less explainable. Hence, the higher the explainability E_{ui_+} and the lower the explainability E_{ui_-}, the more the instance (u, i_+, i_-) will contribute to the learning. This also means that, when generating a recommendation

list to a user u, the items ranked at the top of the list would be expected to have higher explainability than the items ranked lower in the list. Thus, the multiplicative explainability term can be seen as one way to formulate an *explainable preference* function, that is furthermore flexible, since any explainability score can be incorporated.

The latter objective function may seem counter-intuitive due to the fact that the loss increases when the explainability weighting term $E_{ui_+}(1 - E_{ui_-})$ increases. However, the model learns with the update equations regardless of the value of the loss. Hence, instead of trying to reduce the loss further when the explainability weighting term $E_{ui_+}(1 - E_{ui_-})$ increases, we aim to increase the *contribution* of the instance (u, i_+, i_-) to the learning objective. For a better insight, we derive the update equations of EBPR, with respect to model parameters Ω, below:

$$\frac{\partial L_{EBPR}}{\partial \Omega} = \frac{-1}{|D|} \sum_{(u,i_+,i_-) \in D} E_{ui_+}(1 - E_{ui_-}) \frac{e^{-f_\Omega(u,i_+,i_-)}}{1 + e^{-f_\Omega(u,i_+,i_-)}} \frac{\partial f_\Omega(u, i_+, i_-)}{\partial \Omega} \tag{3.3}$$

From (2.2), we have

$$\frac{\partial f_\Omega(u,i_+,i_-)}{\partial \Omega} = \begin{cases} Q_{i_+k} - Q_{i_-k} \; if \; \Omega = P_{uk} \\ P_{uk} \; if \; \Omega = Q_{i_+k} \\ -P_{uk} \; if \; \Omega = Q_{i_-k} \\ 0 \; otherwise \end{cases}$$

The amplitude of the gradient with respect to parameter Ω is thus an increasing function of the explainability weighting factor $E_{ui_+}(1 - E_{ui_-})$ in a way that confirms the desired explainable preference aim. For instance, in the extreme case where either the positive item is not explainable at all or the negative item is completely explainable, the update equation is zeroed out. Hence, no contribution will come from the corresponding instance to the learning. This is reasonable and desirable since the aforementioned case depicts a *non* explainable preference, where either the positive item is not explainable or the negative item is explainable. Either case undermines the explainability of the preference.

3.1.1 Explainability Matrix

Various measures of explainability can be defined given the characterized order relation of an item i being "more explainable" than an item j to a user u. The notion of explainability may depend on user or item metadata if using a content-based or hybrid approach. But in a purely collaborative filtering approach such as our case, it should be neighborhood-based as presented in [10] which further categorized the explanations as user-based or item-based. User-based explanations are based on user similarities and generate explanations in the form of "this item was recommended because certain similar users liked it". Item-based explanations use item-similarities and generate explanations in the form "the item was recommended because you liked similar items". We extend the idea of neighborhood-based explainability from [10] because it has shown success as an intuitive method for modifying loss-based recommendation models [90, 91]. Both item-based and user-based measures of explainability can be defined by relying solely on the interaction matrix (or rating matrix, depending on the type of feedback). However, in this work, we focus only on item-based explanations which are expected to be more intuitive and informative to the user than user-based explanations. This is because the user knows the items that they interacted with but does not necessarily know his or her neighbors who has similar interactions with items. That said, a user-based explainability matrix can be similarly defined by applying the same strategy, described below, on the transpose of the interaction matrix. We define the measure of explainability E_{ui} as the probability of user u interacting with item i's neighbors. The latter is defined as follows:

Definition 2 (Item-based explainability)**.**

$$E_{ui} = P(Y_{uj} = 1 | j \in N_i^{\eta}) \tag{3.4}$$

where N_i^{η} is the neighborhood of item i which is a set of item i's η most similar items given a similarity measure. Y_{ui} is a Bernoulli random variable that takes value 1 *if user u interacted with item i and 0 otherwise.*

$$Y_{ui} = \begin{cases} 1 \; if \; i \in I_u^+ \\ 0 \; otherwise \end{cases}$$

The explainability E_{ui} can also be reformulated as follows:

$$E_{ui} = \frac{|N_i^\eta \cap I_u^+|}{\eta} \tag{3.5}$$

This means that for a specific item, the more neighboring items a given user has interacted with, the higher the explainability of that item will be to this user. In our experiments, we use the Cosine similarity between items to generate the neighborhoods.

3.1.1.1 Justifications for the Choice of Explainability

In contrast to post-hoc explainability approaches, which generate explanations after the predictions have been made, our approach pre-computes explanation scores, then uses them to learn an explainable model. This leads to two advantages: (1) better transparency since there is no post-hoc model and (2) avoiding the heavy cost of post-hoc model training and explanation generation at prediction time.

Aiming toward *transparency* is also why we chose to use *neighborhood-based explainability*. More specifically, our aim is to explain recommendations using *only* the input data used by the recommendation algorithm, and not any additional data that is not used to generate predictions. Consequently and because BPR uses no metadata, the explanations must be sourced from only the interaction data.

3.1.2 Training Complexity of EBPR

The complexity of learning the BPR model is $\mathcal{O}(|D|K)$, where $|D|$ is the size of the training data, and K is the latent space dimensionality, or number of of latent factors. This is because the complexity of forward and backward propagating an instance stems from computing two dot products, which is $\mathcal{O}(K)$. Considering that generating the explainability matrix can be done offline in the data pre-processing phase, no additional

time complexity needs to be added to the training process of EBPR compared to BPR. That said, the explainability matrix was computed only once for all our experiments, and the most significant part of the computation was computing the similarity values initially, which can be done very efficiently, owing to the sparsity of the interactions and the power law of the data, allowing the use of sparse structures and locality sensitive hashing.

3.2 Exposure Bias in EBPR

As proved in [11], the estimator optimized in BPR is biased against the ideal pairwise loss. This is because the choice of the positive and negative items depends on the interaction random variable Y_{ui} instead of the relevance. We consider two Bernoulli random variables $O_{u,i} \sim Ber(\theta_{ui})$, where $\theta_{ui} = P(O_{ui} = 1)$ represents the exposure propensity of item i relative to user u; and $R_{u,i} \sim Ber(\gamma_{ui})$, where $\gamma_{ui} = P(R_{ui} = 1)$ represents the probability of item i being relevant to user u. $O_{u,i}$ and $R_{u,i}$ represent, respectively, whether item i is exposed or relevant to user u. We only know if user u interacted with item i when the item is both observed and relevant. In other words, $Y_{ui} = O_{ui}R_{ui}$ [11]. However, there could be relevant unobserved items that the user did not get a chance to observe in order to interact with. To counteract this issue, [11] proposed an Inverse Propensity Scoring (IPS) based estimator, as was done earlier for explicit feedback ratings in [7], that is unbiased with respect to the ideal pairwise estimator. The latter is defined as follows:

Definition 3 (Unbiased estimator for the ideal BPR loss)**.**

$$L_{BPR}^{unbiased} = \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\frac{Y_{ui_+}}{\theta_{ui_+}}(1 - \frac{Y_{ui_-}}{\theta_{ui_-}})log\sigma(f_\Omega(u, i_+, i_-)). \tag{3.6}$$

Given that the explainability scores E_{ui} also rely on the interaction random variable Y_{ui}, it is reasonable to suspect that the explainability weighting of the loss could introduce some additional exposure bias. In fact, it would be ideal to use the relevance to define the explainability matrix such that:

Definition 4 (Ideal explainability matrix)**.**

$$E_{ui}^{ideal} = P(R_{uj} = 1|j \in N_i^\eta) \tag{3.7}$$

This being done, we use the ideal explainability matrix to define the ideal EBPR loss as follows:

Definition 5 (Ideal EBPR loss)**.**

$$L_{EBPR}^{ideal} = \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\gamma_{ui_+}(1-\gamma_{ui_-})E_{ui_+}^{ideal}(1-E_{ui_-}^{ideal}) \times log\sigma(f_\Omega(u,i_+,i_-)). \tag{3.8}$$

To quantify the additional bias, we compare the ideal EBPR loss to an IPS-based estimator similar to the one defined in Definition 3, but with explainability weighting. We call the latter estimator pUEBPR loss, where the "pU" stands for partially unbiased, and formulate it as follows:

Definition 6 (Partially Unbiased Explainable BPR (pUEBPR) loss)**.**

$$L_{pUEBPR} = \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\frac{Y_{ui_+}}{\theta_{ui_+}}(1-\frac{Y_{ui_-}}{\theta_{ui_-}})E_{ui_+}(1-E_{ui_-})log\sigma(f_\Omega(u,i_+,i_-)). \tag{3.9}$$

The pUEBPR loss eliminates the initial exposure bias of BPR without taking into account the impact of adding explainability. Thus it is not a complete debiasing. However, as we will show below, this *partial* debiasing loss will allow us to quantify the additional bias coming from adding the explainability weighting to BPR.

Next, we will prove that the *explainability* weighting in the EBPR loss introduces *additional* exposure bias. Then we proceed to eliminate this additional bias in the next section.

Proposition 3.2.1 (Additional exposure bias from explainability weighting in EBPR)**.** *The explainability weighting in the EBPR loss introduces additional non-zero exposure bias, given by*

$$Additional\_Bias\_EBPR = \mathbb{E}[L_{pUEBPR}] - L_{EBPR}^{ideal} \neq 0 \tag{3.10}$$

Proof.

$$
\begin{aligned}
&Additional\_Bias\_EBPR = \mathbb{E}[\frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\frac{Y_{ui_+}}{\theta_{ui_+}}(1-\frac{Y_{ui_-}}{\theta_{ui_-}}) \\
&\times E_{ui_+}(1-E_{ui_-})log\sigma(f_\Omega(u,i_+,i_-))] - \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\gamma_{ui_+} \\
&\times (1-\gamma_{ui_-})E^{ideal}_{ui_+}(1-E^{ideal}_{ui_-})log\sigma(f_\Omega(u,i_+,i_-)) \\
&= \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\frac{\mathbb{E}[Y_{ui_+}]}{\theta_{ui_+}}(1-\frac{\mathbb{E}[Y_{ui_-}]}{\theta_{ui_-}})E_{ui_+}(1-E_{ui_-}) \\
&\times log\sigma(f_\Omega(u,i_+,i_-)) - \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\gamma_{ui_+}(1-\gamma_{ui_-})E^{ideal}_{ui_+} \\
&\times (1-E^{ideal}_{ui_-})log\sigma(f_\Omega(u,i_+,i_-)) \\
&= \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\gamma_{ui_+}(1-\gamma_{ui_-})[E_{ui_+}(1-E_{ui_-}) \\
&- E^{ideal}_{ui_+}(1-E^{ideal}_{ui_-})]log\sigma(f_\Omega(u,i_+,i_-)) \neq 0
\end{aligned}
$$

□

3.3 Unbiased EBPR estimator

We follow the same IPS-based methodology on the explainability weighting to propose an unbiased estimator for the ideal EBPR loss:

Definition 7 (Unbiased EBPR (UEBPR) estimator)**.**

$$
L_{UEBPR} = \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\frac{Y_{ui_+}}{\theta_{ui_+}}(1-\frac{Y_{ui_-}}{\theta_{ui_-}})\frac{E_{ui_+}}{\theta_{uN^\eta_{i_+}}}(1-\frac{E_{ui_-}}{\theta_{uN^\eta_{i_-}}})log\sigma(f_\Omega(u,i_+,i_-)) \tag{3.11}
$$

where $\theta_{uN^\eta_i} = P(O_{uj} = 1|j \in N^\eta_i)$ is the probability of user u being exposed to the neighbors of item i. $\theta_{uN^\eta_i}$ can also be considered as the item's neighborhood propensity relative to user u.

Now, we prove that this new UEPBR estimator is unbiased for the ideal EBPR loss in the following proposition.

Proposition 3.3.1. *The UEBPR estimator is unbiased for the ideal EBPR loss, meaning that*

$$\mathbb{E}[L_{UEBPR}] = L^{ideal}_{EBPR} \tag{3.12}$$

Proof.

$$\begin{aligned}
&\mathbb{E}[L_{UEBPR}] = \mathbb{E}[\frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\frac{Y_{ui_+}}{\theta_{ui_+}}(1-\frac{Y_{ui_-}}{\theta_{ui_-}})\frac{E_{ui_+}}{\theta_{uN^\eta_{i_+}}} \\
&\times (1-\frac{E_{ui_-}}{\theta_{uN^\eta_{i_-}}}) log\sigma(f_\Omega(u,i_+,i_-))] \\
&= \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\frac{\mathbb{E}[Y_{ui_+}]}{\theta_{ui_+}}(1-\frac{\mathbb{E}[Y_{ui_-}]}{\theta_{ui_-}})\frac{E_{ui_+}}{\theta_{uN^\eta_{i_+}}}(1-\frac{E_{ui_-}}{\theta_{uN^\eta_{i_-}}}) \\
&\times log\sigma(f_\Omega(u,i_+,i_-)) \\
&= \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\gamma_{ui_+}(1-\gamma_{ui_-})\frac{E_{ui_+}}{\theta_{uN^\eta_{i_+}}}(1-\frac{E_{ui_-}}{\theta_{uN^\eta_{i_-}}}) \\
&\times log\sigma(f_\Omega(u,i_+,i_-)) \\
&= \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\gamma_{ui_+}(1-\gamma_{ui_-})\frac{P(O_{uj}=1,R_{uj}=1|j\in N^\eta_{i_+})}{\theta_{uN^\eta_{i_+}}} \\
&\times (1-\frac{P(O_{uj}=1,R_{uj}=1|j\in N^\eta_{i_-})}{\theta_{uN^\eta_{i_-}}}) log\sigma(f_\Omega(u,i_+,i_-)) \\
&= \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\gamma_{ui_+}(1-\gamma_{ui_-}) \\
&\times \frac{P(O_{uj}=1|j\in N^\eta_{i_+})P(R_{uj}=1|j\in N^\eta_{i_+})}{\theta_{uN^\eta_{i_+}}} \\
&\times (1-\frac{P(O_{uj}=1|j\in N^\eta_{i_-})P(R_{uj}=1|j\in N^\eta_{i_-})}{\theta_{uN^\eta_{i_-}}}) log\sigma(f_\Omega(u,i_+,i_-)) \\
&= \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\gamma_{ui_+}(1-\gamma_{ui_-})\frac{\theta_{uN^\eta_{i_+}}E^{ideal}_{ui_+}}{\theta_{uN^\eta_{i_+}}} \\
&\times (1-\frac{\theta_{uN^\eta_{i_-}}E^{ideal}_{ui_-}}{\theta_{uN^\eta_{i_-}}}) log\sigma(f_\Omega(u,i_+,i_-)) \\
&= \frac{1}{|U||I|^2} \sum_{(u,i_+,i_-)\in U\times I\times I} -\gamma_{ui_+}(1-\gamma_{ui_-})E^{ideal}_{ui_+}(1-E^{ideal}_{ui_-}) \\
&\times log\sigma(f_\Omega(u,i_+,i_-)) = L^{ideal}_{EBPR}
\end{aligned}$$

□

Note that in the latter proof, we assume conditional independence between exposure

TABLE 3.1

Datasets used for evaluation.

Dataset	Task	Users	Items	Interactions	Sparsity
ml-100k	Movie rec.	943	1,682	100,000	93.6%
yahoo-r3	Song rec.	15,400	1,000	311,704	97.9%
lastfm-2k	Artist rec.	1,874	17,612	92,780	99.7%

and relevance given the neighborhood, a much less restrictive (and thus more realistic) assumption than global independence.

3.4 Experimental Evaluation

3.4.1 Data Used

We use three real benchmarking datasets: The Movielens 100K [108] (ml-100k), The Yahoo! R3 [109] (yahoo-r3) and the Last.FM 2K [110, 111] (lastfm-2k) datasets. These datasets consist of, respectively, 100K movie interactions, over 311K song interactions, and over 92K artist interactions. The interactions consist of either ratings or play counts, which were converted into binary interactions, regardless of their values. In fact, any rating or play count over the threshold of zero is considered a positive interaction. Then we filtered out users with less than 10 interactions in the lastfm-2k dataset to ensure enough training and evaluation samples for every user and reduce the data sparsity. The other two datasets similarly have at least 10 interactions per user. The dataset statistics are summarized in Table 3.1.

3.4.2 Experimental Setting

We follow the standard Leave-One-Out (LOO) procedure [3, 35] that consists of considering the latest interaction of each user as a test item and comparing it to 100 randomly sampled negative items. In the training, we sample, at every epoch, one negative item for every positive user-item interaction. We implement “BPR”, “UBPR”, “EBPR”,

"pUEBPR" and "UEBPR" and tune their hyperparameters on every dataset by comparing the averages over two replicates of 15 random hyperparameter configurations. We further split the training data into training and validation sets for the hyperparameter tuning. We consider the last interaction of every user from the training data along with 100 sampled negatives (disjoint from those in the test set) per user as a validation set. For each random hyperparameter configuration, we choose a value for the number of latent features, batch size and L2 regularization within the respective sets {5, 10, 20, 50, 100}, {50, 100, 500} and {0, 0.00001, 0.001}. We initially fixed the neighborhood size to 20 to ensure a fair comparison in terms of explainability metrics. However we will investigate the impact of neighborhood size later in Section 3.5.6. This being done, we then re-train every model on the merged train and validation sets with its best hyperparameter configuration for three replicates and report the average results on the test set.

3.4.3 Evaluation Metrics

We use Normalized Discounted Cumulative Gain ($NDCG@\mathcal{K}$) and Hit Ratio ($HR@\mathcal{K}$) for the ranking evaluation. $HR@\mathcal{K}$ measures the proportion of users for whom the relevant test/validation items were recommended within their top $\mathcal{K}$ recommendation lists and is formulated as follows

$$HR@\mathcal{K}(TopK) = \frac{1}{|U|}\sum_{u=1}^{|U|} \mathbb{1}_{Test(u)\in TopK(u)} \tag{3.13}$$

where $TopK$ is the top $\mathcal{K}$ recommendation matrix in which every row represents the Top $\mathcal{K}$ recommendations of a user. $Test(u)$ is the test/validation item of user u. $NDCG@\mathcal{K}$ also measures the capacity of the model to recommend relevant items to users but takes into consideration the rank in which the relevant item was recommended. The idea is that the higher the rank of the relevant item, the higher the metric is penalized. Thus, $NDCG@\mathcal{K}$ is formulated as follows

$$NDCG@\mathcal{K}(TopK) = \frac{DCG@\mathcal{K}(TopK)}{IDCG@\mathcal{K}(TopK)} \tag{3.14}$$

where $DCG@\mathcal{K}(TopK)$ is the Discounted Cumulative Gain, which is divided by the Ideal DCG ($IDCG@\mathcal{K}(TopK)$) for normalization purposes. Hence, the $DCG@\mathcal{K}(TopK)$ is defined as follows

$$DCG@\mathcal{K}(TopK) = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{r=1}^{\mathcal{K}}\frac{\mathbb{1}_{TopK(u,r)\in Test(u)}}{log_2(1+r)}. \tag{3.15}$$

Moreover, we use Mean Explainability Precision ($MEP@\mathcal{K}$) [112] and Weighted MEP ($WMEP@\mathcal{K}$) for the explainability evaluation. $MEP@\mathcal{K}$ is an evaluation metric that measures the proportion of explainable items within the Top K list of recommended items, as follows

$$MEP@\mathcal{K}(TopK) = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{|\{i \in TopK(u)\} \cap \{E_{ui} > 0\}|}{\mathcal{K}} \tag{3.16}$$

We further extend $MEP@\mathcal{K}$ to be able to weight the items' contributions to the numerator by their explainability values, since $MEP@\mathcal{K}$ rewards items that are considered to be explainable (i.e., with explainability score above a given threshold) in the same way, regardless of how different their explainability values are. Hence, we propose a weighted version of MEP, or Weighted MEP (WMEP), that weights items' contributions by their explainability values, as follows:

$$WMEP@\mathcal{K}(TopK) = \frac{1}{|U|}\sum_{u=1}^{|U|}E_{ui}\frac{|\{i \in TopK(u)\} \cap \{E_{ui} > 0\}|}{\mathcal{K}} \tag{3.17}$$

Note that when training a model, we hide all test interactions when generating the explainability matrix to avoid any data leakage from the test set. Then, when evaluating the model on the test set, we generate an explainability matrix that considers all interactions to ensure an evaluation of the actual explainability of the test items to users, although these values were not used in training.

Furthermore, we evaluate the popularity debiasing of the models in three aspects, namely Novelty, Popularity and Diversity. To evaluate the novelty of a model, we use Expected Free Discovery (EFD) [113], which is a measure of the ability of a system to recommend relevant long-tail items [113]. EFD is defined as follows

$$EFD@\mathcal{K}(TopK) = -\frac{1}{|U|}\sum_{u=1}^{|U|}\frac{1}{\mathcal{K}}\sum_{i\in TopK(u)} log_2\hat{\theta}_{ui} \tag{3.18}$$

Note that we use an estimator of the propensity $\hat{\theta}_{ui}$ to represent the popularity as we will see later in Section 3.4.4.

Next, to evaluate the popularity of a model's recommendations, we compute the average popularity (Avg_Pop) of the top $\mathcal{K}$ recommended items for every user as follows

$$Avg\_Pop@\mathcal{K}(TopK) = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{1}{\mathcal{K}}\sum_{i\in TopK(u)} \hat{\theta}_{ui} \tag{3.19}$$

Finally, to evaluate the diversity in a model's recommendations, we compute the Average Pairwise Similarity between the items in a top $\mathcal{K}$ recommendation list, defined as follows [113]

$$Div@\mathcal{K}(TopK) = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{1}{\mathcal{K}(\mathcal{K}-1)}\sum_{\substack{i,j\in TopK(u)\\ i<j}} sim(i,j), \tag{3.20}$$

where $sim(i,j)$ is a measure of similarity between item i and item j's interaction vectors. In our experiments, we use the Cosine similarity. All ranking and explainability metrics are computed at a cutoff $\mathcal{K} = 10$ for Top 10 recommendation.

3.4.4 Propensity Estimation

Following [11], we estimate the propensity of an item to a user by the relative item popularity of the item as follows

$$\hat{\theta}_{ui} = \sqrt{\frac{\sum_{j=1}^{|U|} Y_{ji}}{\max_{l\in I}\sum_{j=1}^{|U|} Y_{jl}}} \tag{3.21}$$

The total propensity of item i within its neighborhood can be defined as the average propensity of the items in the neighborhood[1]:

[1] In our implementation, we ended up omitting the constant denominator in the sum as this yielded better results.

TABLE 3.2: Model comparison in terms of ranking performance and explainability on the three real interaction datasets that were described in Table 1. All evaluation metrics are computed at a cutoff $\mathcal{K}$=10 (Top 10) and reported as the averages over 5 replicates. The best results are in **bold** and second to best results are underlined. A value with * is significantly higher than the next best value (p-value < 0.05).

Dataset	ml-100k				yahoo-r3				lastfm-2k			
Model	NDCG	HR	MEP	WMEP	NDCG	HR	MEP	WMEP	NDCG	HR	MEP	WMEP
BPR	0.3807*	**0.6625**	0.9182*	0.3467*	0.3315*	0.5466	0.8910*	0.1594*	**0.7260***	**0.9086***	0.2142	0.0452
UBPR	0.3676*	0.6401	0.9063*	0.3342	0.3203	0.5422	0.8815	0.1562	0.6613*	0.8340*	0.2338	0.0468*
EBPR	**0.3821***	0.6568*	**0.9314**	**0.3645***	**0.3521**	**0.5674**	**0.9461***	**0.1808***	0.6309*	0.7876*	**0.2629***	**0.0485***
pUEBPR	0.3648*	0.6356*	0.9282*	0.3595*	0.3494*	0.5662*	0.9394*	0.1778*	0.5938*	0.7556*	0.2456*	0.0471*
UEBPR	0.3542	0.6204	0.8986	0.3332	0.3421*	0.5565*	0.9234*	0.1710*	0.5567	0.7284	0.2349*	0.0461

$$\hat{\theta}_{uN_i^\eta} = \frac{1}{\eta} \sum_{l \in N_i^\eta} \hat{\theta}_{ul} \tag{3.22}$$

3.5 Results and Discussion

3.5.1 Overall Ranking and Explainability Results

Table 3.2 lists the results of all the models in terms of ranking performance and explainability. Overall, for both the ml-100k and yahoo-r3 datasets, the explainable models EBPR and pUEBPR outperformed all the other models in terms of ranking performance and explainability for almost all the metrics. Moreover, whenever EBPR was not the best performer, it was still second to best. On the lastfm-2k dataset, the non-explainable models (BPR and UBPR) reached better ranking performance than the explainable models (EBPR, pUEBPR and UEBPR). However, the explainable models were still the winners in terms of explainability (MEP and WMEP). Our interpretation of the exception in the lastfm-2k dataset, is that it is likely due to the extremely high sparsity of this dataset (99.7%), which in turn impacts the similarity based computations to determine the neighborhoods used in computing the explainability values. This in turn degrades the learning of the explainable models due to the vanishing gradient problem. We will investigate this issue further in Section 3.5.5, where we will investigate the effect of the data sparsity on the learning of the explainable models.

3.5.2 Advantages of using Explainability Weighting in the Learning Objective

In order to demonstrate the advantages of the proposed explainability weighting in (3.2), we compare EBPR to BPR and pUEBPR to UBPR because these models only differ by the explainability weighting of the loss. In both the ml-100k and yahoo-r3 datasets, going from BPR to EBPR almost always improves both the ranking and explainability performances. However, going from UBPR to pUEBPR improves the explainability but does not always improve the ranking performance. In fact, the ranking performance improves on the yahoo-r3 dataset but not on the ml-100k dataset. Nevertheless, we will see later, in Section 3.5.6, that pUEBPR outperforms UBPR on the ml-100k dataset when further tuning the neighborhood size. These results are somewhat surprising since while our initial aim was to improve the explainability of the recommended list, we ended up also gaining in ranking accuracy. In other words, explainability does not necessarily require sacrificing accuracy.

3.5.3 Impact of Debiasing on Performance

Contrary to what we noticed from the overall improved performance when adding explainability to any of the models, we notice a different trend in the accuracy when debiasing both models. In fact, on all three datasets, all the evaluation metrics decreased overall every time that debiasing was added: from EBPR to pUEBPR to UEBPR, and from BPR to UBPR. Hence, although the explainable models still perform better overall than the non-explainable models, debiasing explainable models seems to be degrading the ranking performance. However, as the IPS weighting aimed to mitigate the exposure bias in the training phase, the evaluation sets still suffer from exposure bias. And given that the ranking metrics are based on the interaction, rather than relevance, they cannot properly quantify the benefits of the debiasing. To truly evaluate the impact of the exposure debiasing, we evaluate the models in terms of their capacity to capture the *true relevance* which is only available in the yahoo-r3 dataset as described in the following subsection.

TABLE 3.3

Model comparison in terms of ranking performance on the unbiased yahoo-r3 test set: Average results over 5 replicates. The best results are in **bold** and second to best are underlined. A value with * is significantly higher than the next best value (p-value < 0.05).

	BPR	UBPR	EBPR	pUEBPR	UEBPR
NDCG@5	0.6140	0.6152	0.6178*	**0.6187**	0.6180
MAP@5	0.4710	0.4727	0.4752*	**0.4764**	0.4756

3.5.4 Impact of Debiasing on Relevance Modeling

The yahoo-r3 dataset provides an unbiased test set, in which a subset of 5,400 users were provided 10 random songs to rate. The fact that the songs were chosen at random ensures that the test set is free of exposure bias, because all the rated songs have the same propensity of exposure. Thus, the ratings in the unbiased test set represent the true relevance of the items to the users. Hence, evaluating a model in terms of ranking performance on this test set reflects its capacity to capture the true relevance. We re-train all the tuned models on the yahoo-r3 dataset, and evaluate it on the test set in terms of Mean Average Precision at cutoff 5 (MAP@5), and $NDCG$@5, where for both metrics, we assess the relevance of the top $\mathcal{K}$ predicted items for each user, given by their true rating-based ranking. We chose a cutoff of 5 because there are 10 test items per user. We summarize the results in Table 4.2. Almost all the unbiased models performed better than their biased versions, except for pUEBPR which performed slightly better than UEBPR. This is probably due to the nature of the neighborhood propensity estimation. However, overall, the explainable and unbiased models, pUEBPR and UEBPR, were the best performers in terms of ranking performance in an unbiased evaluation setting. This demonstrates the impact of the loss debiasing in better accounting for the true relevance.

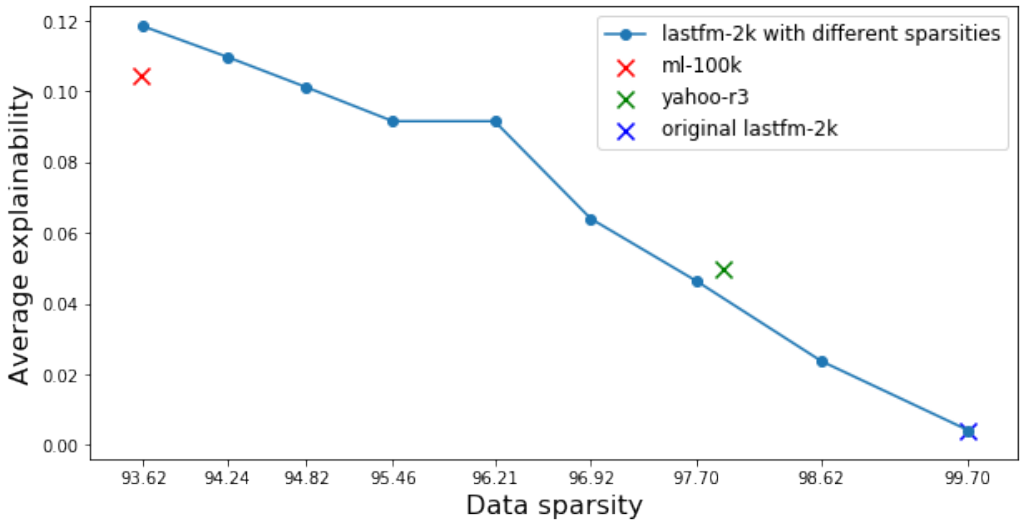


Figure 3.1: Evolution of the average explainability with increasing sparsity of the lastfm-2k dataset. The average explainability values from the ml-100k and yahoo-r3 datasets are also shown for comparison. The sparsity of the lastfm-2k dataset is at least one order of magnitude lower than that of the other two datasets. Moreover, there seems to be a linear relationship between explainability and data sparsity. Thus, the data sparsity engenders a vanishing gradients problem.

3.5.5 Impact of Data Sparsity on Learning

In order to study the effect of the data sparsity on the performance of the explainable models, following our discussion in Section 3.5.1, we decided to explore the relationship between sparsity and explainability for the one data set (lastfm-2k) for which the performance trends differed. We do this by assessing the evolution of the explainability values from the explainability matrix, while gradually decreasing the sparsity of the dataset. To reduce the data sparsity, we gradually, filtered out items with fewer than a certain threshold of interactions, namely 5, 10, 15, 20, 25, 30, 35 and 40 user interactions. For each generated dataset, we compute the explainability matrix and calculate the average explainability value E_{ui} in (3.4). We show the evolution of the average explainability with respect to the sparsity of the lastfm-2k dataset in Fig. 3.1. We also show the average explainability values obtained from the ml-100k and yahoo-r3 datasets for comparison purposes. The original lastfm-2k dataset has an average explainability of 0.0041 which is at least one order of magnitude lower than the average explainability values of 0.1043 and 0.0497 on the ml-100k and yahoo-r3

datasets, respectively. In the explainable models (EBPR, pUEBPR and UEBPR), the explainability values are multiplication factors in the update equations (3.3). Hence, having explainability values that are close to 0 will cause the gradients to vanish and the learning to stall. Fig. 3.1 shows a decreasing linear relationship between the explainability values and the data sparsity. Moreover, when reducing the lastfm-2k data sparsity to values near the respective sparsities of the ml-100k (93.6%) and yahoo-r3 (97.9%) datasets, we obtained average explainability values near those obtained from these two datasets. Thus, the data sparsity directly affects the scale of the explainability values. Higher data sparsity leads to lower explainability values and, consequently, a higher risk of vanishing gradients [114]. This confirms our suspicion, in Section 3.5.1, that the explainable models struggle with extremely sparse data due to the vanishing gradients problem [114].

3.5.6 Impact of Neighborhood Size on Performance

The impact of the neighborhood size is two fold: First, the neighborhood size directly impacts the explainability values of items to users, which in turn impact the values of MEP and WMEP. For that reason, we used the same neighborhood size of 20 for all models in the hyperparameter tuning. Second, the explainability values, which depend on the neighbohood size, also impact the training of the explainable models EBPR, pUEBPR and UEBPR. Thus, to compare all models fairly in terms of ranking performance, the neighborhood size must be tuned for these explainable models. In this section, we study the impact of the neighborhood size on the ranking accuracy and explainability. We vary the neighborhood size and re-train all the models in their optimal hyperparameter configurations. We show the results on the ml-100k dataset in Fig. 3.2. We only show the results on the ml-100k dataset to avoid clutter and because we reached similar conclusions for the other two datasets. As expected, the ranking accuracy (NDCG and HR) did not vary for the non-explainable models (BPR and UBPR) for the varying neighborhood sizes, contrarily to the explainable models (EBPR, pUEBPR and UEBPR), whose ranking prediction metrics showed different trends. EBPR and pUEBPR reached their highest ranking at a neighbor-

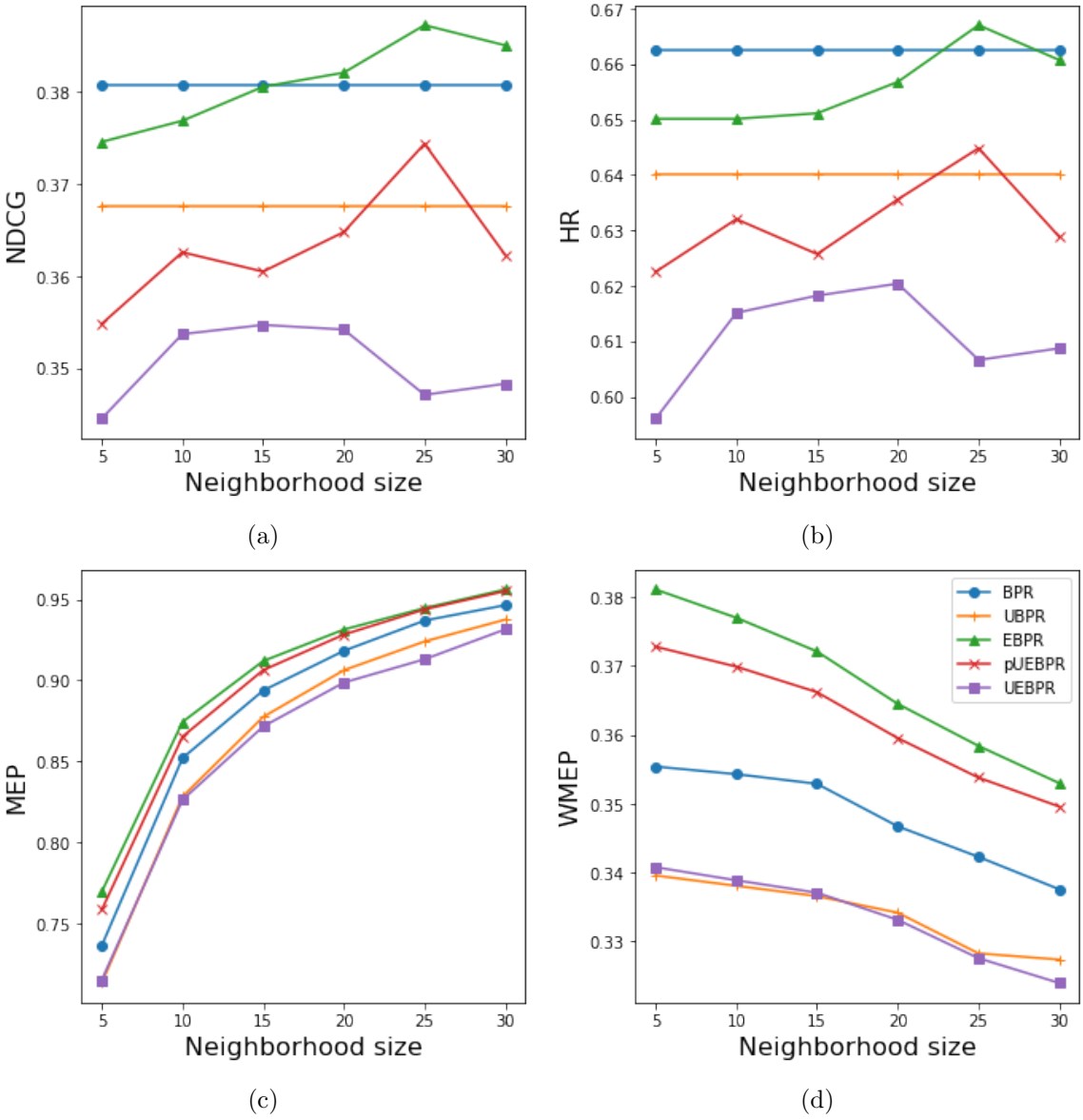


Figure 3.2: Evolution of (a) NDCG@10, (b) HR@10, (c) MEP@10 and (d) WMEP@10 with increasing neighborhood size on the ml-100k dataset. After tuning the neighborhood size, the explainable models outperform their non-explainable counterparts.

TABLE 3.4: Model comparison in terms of Novelty (EFD), Popularity (Avg_Pop) and Diversity (Div) on the three datasets. All evaluation metrics are computed at a cutoff $\mathcal{K}$=10 (Top 10) and reported as the averages over 5 replicates. The best results are in **bold** and second to best results are underlined. HB means the higher the better and LB means the lower the better. Any value with * is significantly higher than the next best value (p-value < 0.05).

Dataset	ml-100k			yahoo-r3			lastfm-2k		
Model	EFD (HB)	Avg_Pop (LB)	Div (LB)	EFD (HB)	Avg_Pop (LB)	Div (LB)	EFD (HB)	Avg_Pop (LB)	Div (LB)
BPR	1.2029	0.4739	0.2675	1.7681	0.3460	0.0811*	2.7714	0.2000	0.0184
UBPR	1.3445*	0.4397*	0.2497*	1.8157	0.3348*	**0.0789***	3.1049*	0.1714*	0.0163*
EBPR	1.2160	0.4677*	0.2650*	1.7682	0.3442	0.0844	**3.4056***	0.1521*	0.0146*
pUEBPR	1.2939*	0.4491*	0.2587*	1.8148*	0.3341	0.0822*	3.3446	0.1531*	0.0137*
UEBPR	**1.4699***	**0.4127***	**0.2414***	**1.8716***	**0.3222***	0.0800*	3.3843*	**0.1478**	**0.0130***

hood size of 25, while UEBPR reached its maximum performance at 20. It is interesting to note that after tuning the neighborhood size, EBPR outperformed BPR and pUEBPR outperformed UBPR in both HR and NDCG which confirms our conclusions in Section 3.5.2, regarding the impact of the explainability weighting on the performance. The explainability metrics show opposite trends with MEP increasing and WMEP decreasing when increasing the neighborhood size. This is due to the fact that larger neighborhood sizes lead to sparser neighborhoods and thus smaller explainability values, and the latter are used as a scale in the WMEP metric. Taking aside the trends, we see that the comparative performance of the models is somewhat consistent for different neighborhood sizes: Overall, EBPR yields the best explainability performance for all neighborhood sizes, followed by pUEBPR.

3.5.7 Explainability as Debiasing or *Explainable Debiasing*

EBPR's superior accuracy with no apparent tradeoff with explainability suggests an inherent popularity debiasing mechanism that is a byproduct of adding explainability. This is certainly possible because the explainability term $E_{ui_+}(1 - E_{ui_-})$, when multiplied into the ranking accuracy loss, captures finer detail about an item's rating from the item's neighbors in addition to the item's own rating. This term has therefore ended up counteracting the bias of very popular items by relying on their neighborhoods. In fact, the explainability weighting term is expected to pull very popular items down, similarly to propensity

debiasing. However what the proposed explainability term, ends up doing, in contrast to propensity debiasing, is succeeding in the estimation of propensity, more accurately and in a local way, namely by using the neighborhood around each item, and not solely the item itself. The advantage of the explainability term is also that it takes into account the local neighborhood to provide a rationale for both positive and negative interactions. Indeed the explainability score is not only providing intuitive quantitative explanation scores for output predictions, but also providing a rationale for debiasing, effectively providing what can be considered an *explainable local debiasing* strategy for each item. Next, we investigate this powerful idea for local explainable propensity estimation by evaluating and comparing the models in terms of Novelty (EFD), Popularity (Avg_Pop) and Diversity (Div). We summarize our results in Table 3.4. For all datasets and for almost all evaluation metrics, the explainable model EBPR outperformed the vanilla BPR, thus supporting our aforementioned claims of popularity debiasing with explainability weighting. Moreover, adding the exposure debiasing (moving from BPR to UBPR or moving from EBPR to pUEBPR then UEBPR) almost always improves the popularity bias metrics. This demonstrates a relationship between exposure bias and popularity bias [8,115] where mitigating the former consequently mitigates the latter. Finally, UEBPR showed the best popularity debiasing overall on all the datasets. The considerably high debiasing performance of UEBPR is likely due to its down-weighting of the items with popular *neighborhoods*, in addition to the popular items, hence allowing the less popular items to be discovered. We plan to investigate this further in future work.

3.6 Chapter Summary

In this chapter, we presented new approaches for promoting explainability and mitigating exposure bias in pairwise ranking recommendation with user profiles. We started by motivating the importance of explainability in recommendation and proposed a novel loss function, called EBPR, that is based on the BPR loss and which is able to capture an explainable preference of the items to the users. Then, we focused on exposure bias and

theoretically proved that not only the proposed EBPR loss suffers from exposure bias, but there is an additional exposure bias introduced from adding the explainability weighting component. This led us to propose a second loss function, called UEBPR, which jointly solves the problems of lack of explainability and exposure bias. Finally, we conducted an extensive experimental evaluation to study the strengths and limitations of our proposed approaches.

Our proposed EBPR approach showed an increase in ranking accuracy of about 4% and an increase in explainability of about 7% over the vanilla BPR model when performing experiments on real-world recommendation datasets. Moreover, experiments on a real-world unbiased testing dataset demonstrated the importance of coupling explainability and exposure debiasing in capturing the true preferences of the user with a significant improvement of 1% over the unbiased model UBPR [11]. Also, coupling explainability with exposure debiasing showed high popularity bias mitigation capabilities with an improvement in popularity debiasing metrics of over 10% overall in three real-world recommendation tasks over the unbiased UBPR model. These results demonstrate the viability of our proposed approaches and their capacity to improve the user's experience by better capturing and modeling their true preferences, improving the explainability of the recommendations, and presenting them with more diverse recommendations that span a larger portion of the item catalog.

In the next chapter, we will present our proposed approach for mitigating exposure bias in sequential recommendation with bidirectional transformers.

CHAPTER 4

DEBIASING THE CLOZE TASK IN SEQUENTIAL RECOMMENDATION WITH BIDIRECTIONAL TRANSFORMERS

In this chapter, we propose a new approach to address exposure bias in the task of sequential recommendation from implicit feedback. We start by formulating the sequential recommendation problem in alignment with the Cloze task [23]. Then, we state the exposure bias problem and theoretically prove that the Cloze task loss, in sequential recommendation, is biased against the ideal loss function which we also define. This being done, we discuss the shortcomings of the Inverse Propensity Scoring (IPS) method [7] in eliminating exposure bias in the sequential recommendation setting and present our novel proposed debiasing framework, that we name *Inverse Temporal Propensity Scoring* (ITPS). Finally, we conduct experiments that demonstrate the advantages of our proposed approach.

4.1 Problem Formulation and Motivation

4.1.1 Sequential Recommendation

Let S be a sequential recommendation dataset comprised of $|S|$ sequences. Each sequence S_s in the dataset is a succession of consecutive item interactions by a user during a certain period of time. An interaction could be defined as a click, rating, review, or consumption depending on the dataset. Similarly, the time span of the sequence could be short or long. Also, consider a set of items I. The sequence S_s can be represented by its item interactions, for example $S_s = [I_1, I_5, I_9, I_2, I_3]$. Each sequence has a distinct number of time steps, or number of item interactions. We assume that all the sequences are normalized to the same number of time steps T to fit the input requirements of transformer-based models. To do so, sequences that are longer than T time steps are truncated to the

most recent T interactions, and sequences that are shorter than T time steps are padded with a padding item 0 at the beginning. Hence, the dataset S is converted to a matrix $S \in I \cup \{0\}^{|S| \times T}$, where element $S_{s,t}$ represents the item, belonging to I, in sequence S_s at time step t. The goal of sequential recommendation is to build a model that is able to accurately predict the next item interaction given a context of previous interactions in a sequence. We represent the trained model by the function f_Ω, with parameters Ω, such that $f_\Omega : [1, |S|] \times [1, T] \times [1, |I|] \to \mathbb{R}; (s, t, i) \mapsto f_\Omega(S_{s,t}, I_i)$. The model f_Ω outputs a prediction of the relevance of item I_i for sequence $S_{s,t}$ at time step t. More specifically, in our work, f_Ω is the bi-directional transformer-based model BERT4Rec [4]. We refer the reader to Section 2.4 for a review on transformer-based approaches for sequential recommendation including BERT4Rec. That said, we note that all the findings described in this paper are model-agnostic, as long as the model is trained for the Cloze task, and is capable of modeling sequential data.

4.1.2 The Cloze Task in Sequential Recommendation

The Cloze task [23] consists of randomly masking a percentage ρ of the tokens, in our case items in the sequence, and training the machine learning model to predict those masked tokens. This approach, also called "Masked Language Model" (MLM) [5], allows for learning a bidirectional context in the training sequence without any information leakage [4] from the future. This ability of modeling a bidirectional context through the Cloze objective is what gives BERT4Rec its prediction power compared to other models, such as uni-directional self-attention based recommender systems [20]. Consider a training dataset $S^m \in I \cup \{0, \langle mask \rangle\}^{|S| \times T}$. S^m is a masked version of the ground truth dataset S where a fraction ρ of the items is replaced with the token $\langle mask \rangle$ in each sequence. The goal of the Cloze task is to train the hypothesis f_Ω to reconstruct the ground truth dataset S from the masked training dataset S^m. Hence, the loss function associated with the Cloze task is defined as the negative log-likelihood of the predicted probability of correctly predicting the masked tokens, which we formulate as follows:

Definition 8 (Cloze Task Loss Function)**.**

$$L_{Cloze} = \frac{-1}{|S||I|T} \sum_{s=1}^{|S|} \sum_{t=1}^{T} \sum_{i=1}^{|I|} \mathbb{1}_{\{S_{s,t}^m = \langle mask \rangle\}} Y_{S_s,I_i,t} log\ softmax(f_\Omega(S_{s,t}^m, I_i)) \tag{4.1}$$

Where $softmax(f_\Omega(S_{s,t}^m, I_i)) = \frac{e^{f_\Omega(S_{s,t}^m, I_i)}}{\sum_{k=1}^{|I|} e^{f_\Omega(S_{s,t}^m, I_k)}}$ accounts for the predicted probability $P(S_{s,t} = I_i | S_s^m)$ of the ground truth item in sequence S_s at time step t being I_i given the masked sequence S_s^m. $Y_{S_s,I_i,t}$ is a binary random variable that equals 1 when item $I_i \in I$ is interacted within sequence $S_s \in S$ at time step $t \in [1, T]$, and 0 otherwise.

4.1.3 Exposure Bias in the Cloze Task

The Cloze loss function in Definition 8 considers the interacted ground truth item $S_{s,t}$ as the desirable and relevant target item for the input $S_{s,t}^m$. However, as shown in previous work [7, 11, 22], interaction does not necessarily signify relevance. In other words, an item could be interacted because it was the most relevant item among the items that the user was exposed to within the item sequence at the corresponding time step. Moreover, non-interacted items could be relevant to some extent, and it could be that the user did not interact with them because they were not exposed to the user. It is this estimation of the relevance of an item with the interaction that engenders the exposure bias within a sequence. Hence, we can define the ideal Cloze task loss function by replacing the interaction random variable $Y_{S_s,I_i,t}$ by the relevance of the item that the user chose to interact with in sequence S_s at time step t, assuming that the user is aware of all items. The awareness of the user of all items completely eliminates the exposure bias because it infers that all items were exposed to the user. Moreover, weighting the interaction by the relevance allows the loss to capture the true relevance of the item at the corresponding time step. Hence, we consider a Bernoulli random variable $R_{S_s,I_i,t} \sim Ber(\gamma_{S_s,I_i,t})$, where $\gamma_{S_s,I_i,t} = P(R_{S_s,I_i,t} = 1)$ represents the probability of item I_i being relevant in sequence S_s at time step t (i.e., $R_{S_s,I_i,t}$ equals 1).

Moreover, we define a Choice random variable that simulates the user behaviour when choosing to interact with item I_i within sequence S_s at time step t. We assume that

this choice is contingent upon its relevance compared to the relevance of all the other items given the sequence context. Hence, we can model the Choice random variable $C_{S_s,I_i,t}$ by a Categorical (Generalized Bernoulli) distribution as follows:

$$C_{S_s,t} \sim Cat(|I|, [\gamma_{S_s,I_1,t}, .., \gamma_{S_s,I_{|I|},t}]) \tag{4.2}$$

The outcome of the random variable is a vector of $|I|$ zeroes except for a 1 for the item the user chooses to interact with. This means that the user chooses one of the $|I|$ items based on their relevance to the context $S_{s,t}$. We denote the outcome of $C_{S_s,t}$ for item I_i by $C_{S_s,I_i,t}$. Finally, we define the ideal Cloze task loss function as follows:

Definition 9 (Ideal Cloze Task Loss Function)**.**

$$L^{ideal}_{Cloze} = \frac{-1}{|S||I|T} \sum_{s=1}^{|S|} \sum_{t=1}^{T} \sum_{i=1}^{|I|} \mathbb{1}_{\{S^m_{s,t}=\langle mask \rangle\}} C_{S_s,I_i,t} \gamma_{S_s,I_i,t} log\ softmax(f_\Omega(S^m_{s,t}, I_i)) \tag{4.3}$$

The discrepancy between the interaction random variable $Y_{S_s,I_i,t}$ and the product $C_{S_s,I_i,t}\ \gamma_{S_s,I_i,t}$ causes the Cloze task loss function to be biased against the ideal loss, as stated in the following Proposition:

Proposition 4.1.1 (Exposure Bias of the Cloze Task Loss Function)**.** *The Cloze task loss function is biased against the ideal Cloze task loss, such that:*

$$\mathbb{E}[L_{Cloze}] \neq L^{ideal}_{Cloze} \tag{4.4}$$

Proof.

$$\mathbb{E}[L_{Cloze}] = \mathbb{E}[\frac{-1}{|S||I|T} \sum_{s=1}^{|S|} \sum_{t=1}^{T} \sum_{i=1}^{|I|} \mathbb{1}_{\{S^m_{s,t}=\langle mask \rangle\}} Y_{S_s,I_i,t} log\ softmax(f_\Omega(S^m_{s,t}, I_i))]$$

$$= \frac{-1}{|S||I|T} \sum_{s=1}^{|S|} \sum_{t=1}^{T} \sum_{i=1}^{|I|} \mathbb{1}_{\{S^m_{s,t}=\langle mask \rangle\}} C_{S_s,I_i,t}\ \theta_{S_s,I_i,t}\ \gamma_{S_s,I_i,t} log\ softmax(f_\Omega(S^m_{s,t}, I_i))$$

Given that the temporal propensities $\theta_{S_s,I_i,t}$ cannot always be equal to 1, $\forall(S_s, I_i, t) \in S \times I \times [1..T]$. Thus, $\mathbb{E}[L_{Cloze}] \neq L^{ideal}_{Cloze}$. □

Note that the proof relies on the probabilistic model of the interaction random variable that is proposed later in Definition 11.

4.1.4 Inverse Propensity Scoring in the Cloze Task and Its Limitations

The common solution to debiasing a maximum likelihood-based loss function for recommendation is Inverse Propensity Scoring (IPS), where an IPS-based estimator of the ideal pointwise loss is obtained by weighting every item prediction for a user by the reciprocal of its exposure propensity for that user [22]. The IPS framework is suitable for debiasing loss functions for recommendation with user profiles. However, we argue that it does not extend to sequential recommendation for the following two reasons:

(1) Inadequacy of the interaction random variable representation: First, the IPS-based framework for recommendation with user profiles [22], models the interaction random variable $Y_{u,i}$, that represents whether user u interacted with item i, by the product of the relevance and the exposure of the item to the user. In fact, the framework relies on two random variables, $O_{u,i} \sim Ber(\theta_{u,i})$ and $R_{u,i} \sim Ber(\gamma_{u,i})$, of exposure and relevance respectively, and models the interaction using $Y_{u,i} = O_{u,i}R_{u,i}$. This means that an item is interacted with by a user if and only if it is both observed by the user and relevant to the user. If we extend this modeling of the interaction to sequential recommendation by mapping users to sequences and introducing the temporal component, we would obtain for a sequence S_s, an item I_i and a time step t: $Y_{S_s,I_i,t} = O_{S_s,I_i,t}R_{S_s,I_i,t}$, where $R_{S_s,I_i,t}$ is the relevance random variable, and $O_{S_s,I_i,t}$ is a Bernoulli exposure random variable that takes value 1 if item I_i was exposed in sequence S_s at time step t, such that $O_{S_s,I_i,t} \sim Ber(\theta_{S_s,I_i,t})$. θ is the probability of exposure such that $\theta_{S_s,I_i,t} = P(O_{S_s,I_i,t} = 1)$. This modeling of the interaction random variable is inadequate for the sequential recommendation setting. In fact, in traditional recommendation, it is safe to assume that any item that is exposed and relevant to a user is interacted. However, when introducing the temporal component into the equation, the assumption does not hold anymore. This is because a user can only interact with one item at a time. Multiple items can be relevant for the same sequence at the same time step, but only one of them can be interacted with. For this reason, the IPS-based framework for recommendation with user profiles does not extend to sequential recommendation.

(2) Ignoring the temporal component: The IPS estimator for the ideal pointwise loss function down-weights every interaction $Y_{u,i}$ by the propensity of exposure of item i to user u, $\theta_{u,i}$. In order to define an IPS-based Cloze loss for sequential recommendation, we assimilate the users to sequences and consider the propensity of exposure of an item I_i in a sequence S_s as $\theta_{S_s,I_i} = P(O_{S_s,I_i} = 1)$, where $O_{S_s,I_i} \sim Ber(\theta_{S_s,I_i})$ is a Bernoulli random variable that takes the value 1 when item I_i is exposed in sequence S_s. We define the IPS-based Cloze loss as follows:

Definition 10 (Inverse Propensity Scoring-based Cloze Loss Function)**.**

$$L^{IPS}_{Cloze} = \frac{-1}{|S||I|T} \sum_{s=1}^{|S|} \sum_{t=1}^{T} \sum_{i=1}^{|I|} \mathbb{1}_{\{S^m_{s,t}=\langle mask \rangle\}} \frac{Y_{S_s,I_i,t}}{\theta_{S_s,I_i}} log\ softmax(f_\Omega(S^m_{s,t}, I_i)) \tag{4.5}$$

The IPS-based Cloze loss function can only be completely unbiased, that is $\mathbb{E}[L^{IPS}_{Cloze}] = L^{ideal}_{Cloze}$, if the propensity of every item I_i in every sequence S_s at time step t, $\theta_{S_s,I_i,t}$, is equal to the "static" propensity, θ_{S_s,I_i}, of item I_i in sequence S_s. We state this in the following proposition:

Proposition 4.1.2 (Unbiasedness condition of the IPS-based Cloze loss function)**.**

$$\mathbb{E}[L^{IPS}_{Cloze}] = L^{ideal}_{Cloze} \Leftrightarrow \theta_{S_s,I_i,t} = \theta_{S_s,I_i}, \forall (S_s, I_i, t) \in S \times I \times [1..T]. \tag{4.6}$$

Proof.

$$\mathbb{E}[L^{IPS}_{Cloze}] = L^{ideal}_{Cloze}$$

$$\Leftrightarrow \mathbb{E}[\frac{-1}{|S||I|T}\sum_{s=1}^{|S|}\sum_{t=1}^{T}\sum_{i=1}^{|I|}\mathbb{1}_{\{S^m_{s,t}=\langle mask\rangle\}}\frac{Y_{S_s,I_i,t}}{\theta_{S_s,I_i}} log\ softmax(f_\Omega(S^m_{s,t},I_i))]$$

$$= \frac{-1}{|S||I|T}\sum_{s=1}^{|S|}\sum_{t=1}^{T}\sum_{i=1}^{|I|}\mathbb{1}_{\{S^m_{s,t}=\langle mask\rangle\}}C_{S_s,I_i,t}\gamma_{S_s,I_i,t} log\ softmax(f_\Omega(S^m_{s,t},I_i))$$

$$\Leftrightarrow \mathbb{E}[\frac{-1}{|S||I|T}\sum_{s=1}^{|S|}\sum_{t=1}^{T}\sum_{i=1}^{|I|}\mathbb{1}_{\{S^m_{s,t}=\langle mask\rangle\}}\frac{C_{S_s,I_i,t}\ O_{S_s,I_i,t}\ R_{S_s,I_i,t}}{\theta_{S_s,I_i}} log\ softmax(f_\Omega(S^m_{s,t},I_i))]$$

$$= \frac{-1}{|S||I|T}\sum_{s=1}^{|S|}\sum_{t=1}^{T}\sum_{i=1}^{|I|}\mathbb{1}_{\{S^m_{s,t}=\langle mask\rangle\}}C_{S_s,I_i,t}\gamma_{S_s,I_i,t} log\ softmax(f_\Omega(S^m_{s,t},I_i))$$

$$\Leftrightarrow \frac{-1}{|S||I|T}\sum_{s=1}^{|S|}\sum_{t=1}^{T}\sum_{i=1}^{|I|}\mathbb{1}_{\{S^m_{s,t}=\langle mask\rangle\}}\frac{C_{S_s,I_i,t}\ \theta_{S_s,I_i,t}\ \gamma_{S_s,I_i,t}}{\theta_{S_s,I_i}} log\ softmax(f_\Omega(S^m_{s,t},I_i))$$

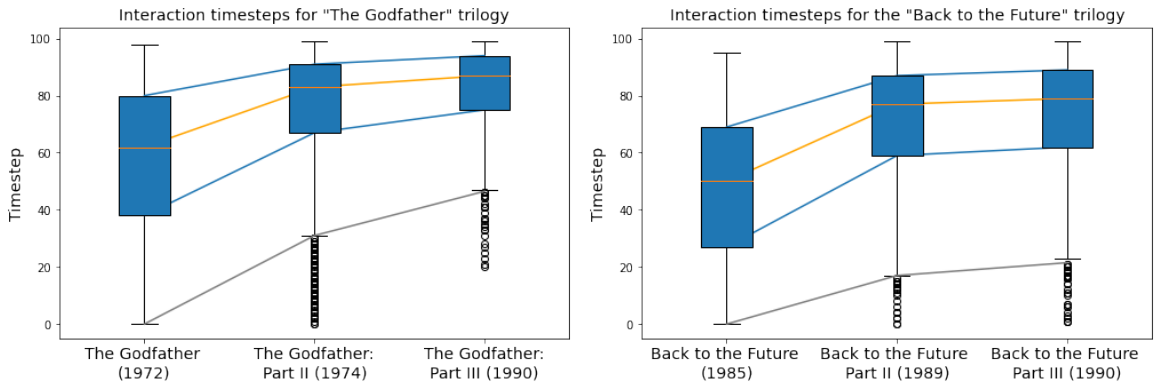
$$= \frac{-1}{|S||I|T}\sum_{s=1}^{|S|}\sum_{t=1}^{T}\sum_{i=1}^{|I|}\mathbb{1}_{\{S^m_{s,t}=\langle mask\rangle\}}C_{S_s,I_i,t}\gamma_{S_s,I_i,t} log\ softmax(f_\Omega(S^m_{s,t},I_i))$$

$$\Leftrightarrow \theta_{S_s,I_i,t} = \theta_{S_s,I_i}, \forall(S_s,I_i,t) \in S \times I \times [1..T].$$ □

Note that the proof also relies on the probabilistic model of the interaction random variable that is proposed later in Definition 11.

This unbiasedness condition of the IPS estimator is unlikely and hard to satisfy as the propensities of exposure tend to vary with the temporal context. We demonstrate this in Figure 4.1 where we show boxplots of the interaction time steps for two movie trilogies in the Movielens 1M dataset [116]. The boxplots show that there are movies that tend to be watched later than others in the sequence; for instance, sequels tend to be watched after the original movies. We chose movies that are older than the dataset to ensure that the differences in observation time are not related to the release dates of the movies, but rather to the temporal context within the trilogies. Hence, given that the interaction distribution tends to vary with time, it is safe to assume that the exposure propensities also vary with time. Thus, in contrast to the IPS framework, they should not be considered static in sequential recommendation.

The latter observation additionally shows how the IPS framework does not extend



(a) (b)

Figure 4.1: Boxplots of the interaction timesteps for (a) "The Godfather" and (b) "Back to the Future" trilogies. The interaction distributions vary through time, meaning that the exposure propensities must not be considered static.

to sequential recommendation. This consequently calls for proposing a new framework that is specifically tailored for debiasing the Cloze task in sequential recommendation, which is the subject of the next section.

4.2 Inverse Temporal Propensity Scoring for an Unbiased Cloze Task

The Inverse Propensity Scoring technique fails to capture the temporal component of the sequential recommendation setting, and hence fails to provide an unbiased estimation of the ideal Cloze task loss. We propose a debiasing framework that is tailored to the Cloze task in sequential recommendation, and that we call **Inverse Temporal Propensity Scoring (ITPS)**. In ITPS, we address the two main limitations of IPS that prevent it from generalizing to sequential recommendation. First, to address the issue of the inadequacy of the interaction random variable representation, we include the outcome of the Choice random variable for item I_i in the interaction model for the following formulation:

Definition 11 (Interaction Random Variable Representation in the ITPS Framework)**.**

$$Y_{S_s,I_i,t} = C_{S_s,I_i,t}\, O_{S_s,I_i,t}\, R_{S_s,I_i,t} \tag{4.7}$$

The latter formulation of the interaction allows for only one item to be interacted within a sequence at a given time step, which is adequate for sequential recommendation.

Now, an item I_i is interacted by a user ($Y_{S_s,I_i,t} = 1$) in a sequence S_s at time step t if and only if the item is exposed ($O_{S_s,I_i,t} = 1$), relevant ($R_{S_s,I_i,t} = 1$) and chosen by the user based on its relevance ($C_{S_s,I_i,t} = 1$). Finally, to account for the temporal component in sequential recommendation in ITPS, we weight the prediction of every item I_i in every sequence S_s at every time step t by the temporal propensity $\theta_{S_s,I_i,t}$ of the item in the sequence at that time step, as opposed to the static propensity θ_{S_s,I_i} of IPS. Thus, we define the ITPS-based Cloze task loss function as follows:

Definition 12 (Inverse Temporal Propensity Scoring-based Cloze Loss Function)**.**

$$L_{Cloze}^{ITPS} = \frac{-1}{|S||I|T} \sum_{s=1}^{|S|} \sum_{t=1}^{T} \sum_{i=1}^{|I|} \mathbb{1}_{\{S_{s,t}^m = \langle mask \rangle\}} \frac{Y_{S_s,I_i,t}}{\theta_{S_s,I_i,t}} log\ softmax(f_\Omega(S_{s,t}^m, I_i)) \tag{4.8}$$

This new ITPS-based loss function is an unbiased estimator of the ideal Cloze task loss, as stated in the following proposition:

Proposition 4.2.1. *The ITPS-based Cloze task loss is unbiased for the ideal Cloze task loss, meaning that*

$$\mathbb{E}[L_{Cloze}^{ITPS}] = L_{Cloze}^{ideal} \tag{4.9}$$

Proof.

$$\mathbb{E}[L_{Cloze}^{ITPS}] = \mathbb{E}[\frac{-1}{|S||I|T} \sum_{s=1}^{|S|} \sum_{t=1}^{T} \sum_{i=1}^{|I|} \mathbb{1}_{\{S_{s,t}^m = \langle mask \rangle\}} \frac{C_{S_s,I_i,t}\ O_{S_s,I_i,t}\ R_{S_s,I_i,t}}{\theta_{S_s,I_i,t}} log\ softmax(f_\Omega(S_{s,t}^m, I_i))]$$

$$= \frac{-1}{|S||I|T} \sum_{s=1}^{|S|} \sum_{t=1}^{T} \sum_{i=1}^{|I|} \mathbb{1}_{\{S_{s,t}^m = \langle mask \rangle\}} \frac{C_{S_s,I_i,t}\ \theta_{S_s,I_i,t}\ \gamma_{S_s,I_i,t}}{\theta_{S_s,I_i,t}} log\ softmax(f_\Omega(S_{s,t}^m, I_i))$$

$$= \frac{-1}{|S||I|T} \sum_{s=1}^{|S|} \sum_{t=1}^{T} \sum_{i=1}^{|I|} \mathbb{1}_{\{S_{s,t}^m = \langle mask \rangle\}} C_{S_s,I_i,t}\ \gamma_{S_s,I_i,t} log\ softmax(f_\Omega(S_{s,t}^m, I_i)) = L_{Cloze}^{ideal} \qquad \square$$

Note that the proof assumes independence between exposure and relevance. Also, it assumes that the outcome of the choice model for an item is deterministic, which is reasonable if we assume a rational user who tends to choose the most relevant item among the exposed items.

4.2.1 Complexity Analysis of the ITPS Framework

The time complexity of forward-propagating a sequence S_s of T time steps through the BERT4Rec model is $\mathcal{O}(T^2d + Td^2)$ per multi-head self-attention layer, where d represents the dimensionality of the query, key, and value weight matrices. The latter complexity corresponds to the sum of complexities of the following two operations: first, linearly transforming the input sequence to compute the query, key, and value matrices, and second, computing the self-attention head's output. However, empirical results in previous related work [4] showed how those operations can be effectively parallelized using GPU acceleration.

Incorporating our proposed ITPS framework into the BERT4Rec model does not impact the training time complexity of the model. In fact, assuming the availability of the temporal exposure propensities, the only additional operation that the ITPS framework introduces into the training of the BERT4Rec model is the division of the loss of every training instance (S_s, I_i, t) with a positive interaction $(Y_{S_s,I_i,t} = 1)$ by the corresponding temporal exposure propensity $\theta_{S_s,I_i,t}$. The latter operation refers to the ITPS-based Cloze task loss in equation 4.8. However, as we will discuss later in section 4.3.2.2, the temporal exposure propensities are unavailable in real recommendation data, hence they need to be estimated. In our real-world experiments in section 4.3.2, we estimate the temporal exposure propensities by the temporal item popularities. Computing those temporal item popularities introduces an additional time complexity of $\mathcal{O}(|S|T)$ because we have to loop over all the interactions to determine the frequency of each item at every time step. That said, the temporal item popularities can be computed once for every dataset for all experiments. Thus, this ensures that there is no difference in the time complexity of training the BERT4Rec model with and without our proposed ITPS-based exposure debiasing.

4.3 Experimental Evaluation

We perform experiments to assess the validity of our theoretical claims of unbiasedness and the applicability of our approach in real recommendation settings. We use semi-synthetic and real world datasets. The semi-synthetic data, used in Section 4.3.1, pro-

TABLE 4.1

Statistics of the real (ml-100k) and semi-synthetic (ss-ml-100k) Movielens 100K datasets.

Dataset	# sequences	# items	# ratings	Avg. length	Sparsity
ml-100k	943	1,349	99,287	105.28	92.19%
ss-ml-100k	943	229	94,104	99.79	56.42%

vides a full visibility of the data properties, allowing us to evaluate the debiasing capabilities of our proposed approach. Moreover, it allows us to control the data properties in order to evaluate the robustness of our approach to varying bias levels. The real datasets, used in Section 4.3.2, allow us to evaluate the applicability of our approach in real recommendation settings. Additionally, we simulate a feedback loop to evaluate the long term effects of the proposed debiasing framework in addition to its ability to better capture the temporal dependencies in the recommendation data.

4.3.1 Experiments on Semi-Synthetic Data

We perform experiments with the aim to answer the following three research questions:

RQ1: How well does the proposed ITPS estimator capture the true relevance?

RQ2: How robust is the proposed ITPS estimator to increasing levels of exposure bias?

RQ3: How important is an unbiased evaluation in assessing exposure debiasing?

In the following subsections, we start by stating our experimental setting, namely the semi-synthetic data creation, proposed unbiased evaluation, hyperparameter tuning, and models compared. Finally, we present the results for the aforementioned research questions.

4.3.1.1 Data

Due to the unavailability of any unbiased sequential recommendation dataset, semi-synthetic experiments were deemed necessary. In fact, only an exposure-unbiased testing

dataset would allow us to truly compare the debiasing capabilities of the different approaches - and we do validate this claim in RQ3. To the extent of our knowledge, the only dataset with an unbiased test set, up to this point, is the Yahoo!R3 dataset[1], which is unfortunately not suitable for sequential recommendation as it does not include timestamps. We therefore use the Movielens 100K (ml-100k)[2] dataset because it is a benchmark dataset that can be used for sequential recommendation since it includes interaction timestamps. This data is described in the first row of Table 4.1. The choice of this dataset is justified due to its relatively low number of sequences (users) and items, compared to other sequential datasets. In fact, our first task is to generate all data properties, including relevance, exposure, and interaction for all sequence, item and timestep tuples; a task that is resource-expensive, especially in terms of memory requirements. Considering a dataset with $|S|$ sequences, $|I|$ items and T time steps, the number of parameters that need to be predicted and kept into memory for each controlled property is $|S| \times |I| \times T$. Hence, given the ml-100k dataset statistics, we would be predicting over 127 Million values for every property. For this reason, using other benchmark datasets with tens of thousands of sequences or items, is simply prohibitive with our current resources. Moreover, similar conclusions could be drawn regardless of the dataset, assuming a high reconstruction quality.

Our goal is to use the available ratings to infer all the data properties, namely the relevance, exposure, and interaction of all items $I_i \in I$, in all sequences $S_s \in S$, and at all time steps $t \in [1, T]$. This is done in the following steps:

(1) We normalize the dataset to $T = 100$ time steps.

(2) We tune and train a Tensor Factorization (TF) model [117,118] on the available (sequence, item, timestep, rating) tuples to reconstruct the missing ratings. We train the model on the Mean Squared Error (MSE) loss for rating prediction. Finally, we use the trained TF model to reconstruct the rating tensor by predicting the missing ratings. Given that the rating represents an explicit measure of satisfaction of a user with an item, we can approximate the probability of relevance of an item I_i in a sequence S_s at a timestep t by

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=r
[2] https://grouplens.org/datasets/movielens/100k/

normalizing the predicted rating with the sigmoid function as follows: $\gamma_{S_s,I_i,t} \approx \sigma(\hat{r}_{s,i,t})$. Here, $\hat{r}_{s,i,t}$ is the predicted rating, obtained by $\hat{r}_{s,i,t} = \sum_{k=1}^{d} P_{s,k} Q_{i,k} W_{t,k}$, where P, Q, and W are respectively the sequence, item, and time latent factor matrices, which all have d latent features.

(3) We use another Tensor Factorization model, but this time trained on implicit feedback to approximate the probabilities of exposure. We convert every rating in the dataset to a positive exposure, and sample a portion of non-interacted tuples as negative exposures. We assume that an item has a higher probability of not being exposed than of being exposed, which is a realistic assumption given the abundance of items in recommendation platforms among which, only a small portion is presented to a user at a time. Thus, we sample 3 negative exposure tuples for every positive exposure tuple. We train the TF model using the Binary Cross Entropy loss for exposure classification. Similarly to step (2), we approximate the propensity of exposure of an item I_i in a sequence S_s at a time step t by the predicted exposure as follows: $\theta_{S_s,I_i,t} \approx \hat{o}_{s,i,t}$. Here, $\hat{o}_{s,i,t}$ is the predicted exposure probability of item i in sequence s at time step t, obtained by: $\hat{o}_{s,i,t} = \sigma(\sum_{k=1}^{d} P_{s,k} Q_{i,k} W_{t,k})$.

(4) Following [22], we introduce a hyperparameter p that controls the skewness of the exposure distribution, and hence the level of exposure bias, as follows:

$$\theta_{S_s,I_i,t} \approx \hat{o}_{s,i,t}^{p}. \tag{4.10}$$

The higher the value of p, the higher the level of exposure bias introduced. We will control the value of p to study RQ2.

(5) We generate the interaction random variable for every sequence S_s, item I_i, and timestep t combination by following the probabilistic model presented in Equation 4.7, such that:

$$O_{S_s,I_i,t} \sim Ber(\theta_{S_s,I_i,t}) \tag{4.11}$$

$$R_{S_s,I_i,t} \sim Ber(\gamma_{S_s,I_i,t}) \tag{4.12}$$

$$C_{S_s,I_i,t} \sim Cat(|I|, [\gamma_{S_s,I_1,t}, .., \gamma_{S_s,I_{|I|},t}]) \tag{4.13}$$

$$Y_{S_s,I_i,t} = C_{S_s,I_i,t}\, O_{S_s,I_i,t}\, R_{S_s,I_i,t}. \tag{4.14}$$

In our implementation, we obtain $C_{S_s,I_i,t}$ by considering a rational user that interacts with the exposed item ($O_{S_s,I_i,t} = 1$) with the highest relevance $\gamma_{S_s,I_i,t}$.

(6) Finally, we filter the interacted instances to construct the semi-synthetic sequential recommendation dataset. The statistics of a sample generated semi-synthetic dataset are presented in the second row of Table 4.1.

4.3.1.2 Evaluation Process

The goal of the debiasing process is to build an unbiased estimator that approximates the ideal loss function in Equation 4.3. The main characteristic of this ideal Cloze loss is that it captures the true relevance of an interaction. Hence, our estimators should be evaluated in terms of their capacity to capture the true relevance of the test interactions. However, our sequence interactions are obtained with the interaction probabilistic model in Equation 4.7, which requires all interactions to be exposed. Hence, sampling the test and validation interactions from the semi-synthetic sequences would not allow for an evaluation in terms of the true relevance. This is because the most relevant items are not necessarily exposed to the user. We cope with this issue using the following evaluation process: We start by splitting the data into training, validation and test sets by considering the last item interaction in each sequence for testing and the second to last for validation. Then, we replace every item interaction in the validation and test sets by the item I_i with the highest relevance $\gamma_{S_s,I_i,t}$ in the corresponding sequence S_s and at the corresponding timestep t. This way, the model is evaluated on its ability to predict the most relevant item, which translates to its ability to capture the true relevance of the items. This being done, we compare the

ranking of the test and validation instances to 100 randomly sampled items. Note that negative sampling does not introduce any bias because, regardless of their exposure, all the negative items are less relevant than the test and validation items, which are the most relevant overall. Thus, our evaluation process is unbiased and evaluates the models in terms of their capacity to capture the true relevance of the items. We use Normalized Discounted Cumulative Gain ($NDCG@k$) and Recall ($R@k$) for the ranking evaluation.

4.3.1.3 Models Compared

We compare the following models:

- **BERT4Rec:** This is the original BERT4Rec model, trained to optimize the Cloze task loss in Equation 4.1. It relies solely on the interaction information and does not incorporate any exposure debiasing.

- **IPS-BERT4Rec:** This is the BERT4Rec model trained with the IPS-based Cloze loss function in Equation 4.5. We estimate the "static" exposure propensities by averaging the temporal exposures, such that $\theta_{S_s,I_i} = \frac{1}{T}\sum_{t=1}^{T}\theta_{S_s,I_i,t}, \forall(S_s, I_i) \in S \times I$.

- **ITPS-BERT4Rec:** This is the BERT4Rec model, trained with our ITPS-based Cloze task loss in Equation 4.8. The loss relies on the temporal propensities $\theta_{S_s,I_i,t}$ to provide an unbiased estimation of the ideal Cloze task loss.

- **Oracle:** This is the BERT4Rec model, trained with the ideal Cloze task loss in Equation 4.3. The latter loss function has access to the true relevance of the items $\gamma_{S_s,I_i,t}$ in the training, and hence, is able to provide a completely unbiased representation of the user preferences. Hence, this model provides an upper bound on capturing the true relevance.

Because the goal of the experiments is to assess the impact of the different debiasing frameworks, we leave the comparison to additional baselines for future work.

4.3.1.4 Hyperparameter Tuning

We tune all the models presented in Section 4.3.1.3, along with the Tensor Factorization models presented in steps 2 and 3 of Section 4.3.1.1 as described below.

Tuning the BERT4Rec models:

Using random search, we tune the number of hidden units within the set {8, 16, 32, 64}, the number of transformer blocks within {1, 2}, the number of attention heads within {1, 2}, the batch size within {8, 16, 32}, the dropout rate within {0, 0.1, 0.2, 0.4}, and finally, the masking probability ρ of the Cloze task within {0.1, 0.15, 0.2, 0.4, 0.6}. We try 30 random combinations, and compare the average $NDCG@10$ results over 3 replicates on the validation set.

Tuning the Tensor Factorization models:

We randomly split the data into training, validation and test sets with the respective ratios 80%, 10% and 10%. We adopt a grid search by trying all combinations of number of latent features within {50, 100, 200}, and batch size within {64, 128, 256}. We replicate every experiment 3 times and compare the average performances on the validation set. The rating-based TF model from step 2 is tuned in terms of Mean Squared Error (MSE) for rating prediction, while the exposure-based TF-model from step 3 is tuned in terms of Area Under the ROC Curve (AUC) for exposure classification.

4.3.1.5 RQ1: How well does the proposed ITPS estimator capture the true relevance?

To answer this research question, we train all the tuned models on the training sequences and use the evaluation process described in Section 4.3.1.2 to evaluate them in terms of their capacity to capture the true relevance by ranking the most relevant items. We summarize the results, which are the average test results over 5 replicates, in Table 4.2. The best performer on all metrics is the Oracle model, owing to its explicit optimization using the relevance levels. The ITPS-BERT4Rec model was second-to-best in all configurations, outperforming the naive BERT4Rec and IPS-BERT4Rec. These findings demonstrate the

TABLE 4.2

Model comparison in terms of capturing the true relevance: Average Recall@k and NDCG@k results over 5 replicates. The best results are in **bold** and second to best results are underlined. A value with * is significantly higher than the next best value (p-value < 0.05).

Model	R@10	NDCG@10	R@5	NDCG@5
BERT4Rec	0.7992	0.6065	0.6917	0.5716
IPS-BERT4Rec	0.7890	0.5961	0.6868	0.5628
ITPS-BERT4Rec	0.8027*	0.6110*	0.6997*	0.5777*
Oracle	**0.8218***	**0.6247***	**0.7083***	**0.5880***

power of the ITPS debiasing framework and validate the theoretical claims of exposure debiasing of the proposed estimator. Finally and interestigly, IPS-BERT4Rec performed worse than the naive BERT4Rec. This is probably due to the fact that it is trained on estimated static propensities, obtained by averaging the temporal propensities, rather than true propensities.

4.3.1.6 RQ2: How robust is the proposed ITPS estimator to increasing levels of exposure bias?

To answer this research question, we train and evaluate the models on semi-synthetic datasets generated with increasing levels of exposure bias. The level of exposure bias is controlled by the power p that governs the propensities $\theta_{S_s,I_i,t}$ in Equation 4.10. We increase p from 1 to 4 with an increment of 1, where the higher the value of p, the stronger the exposure bias introduced in the data, and show the evolution of the ranking metrics on the test set in Figure 4.2. All the models' performances decrease with increasing levels of exposure bias, however with different slopes. The IPS-BERT4Rec model shows the worst performance in handling increasing exposure bias. Its performance quickly degrades starting with the exposure level corresponding to $p = 2$. This shows the inability of the IPS framework to mitigate exposure bias in sequential recommendation. On the other hand, ITPS-BERT4Rec shows the best performance overall in approximating the Oracle. These

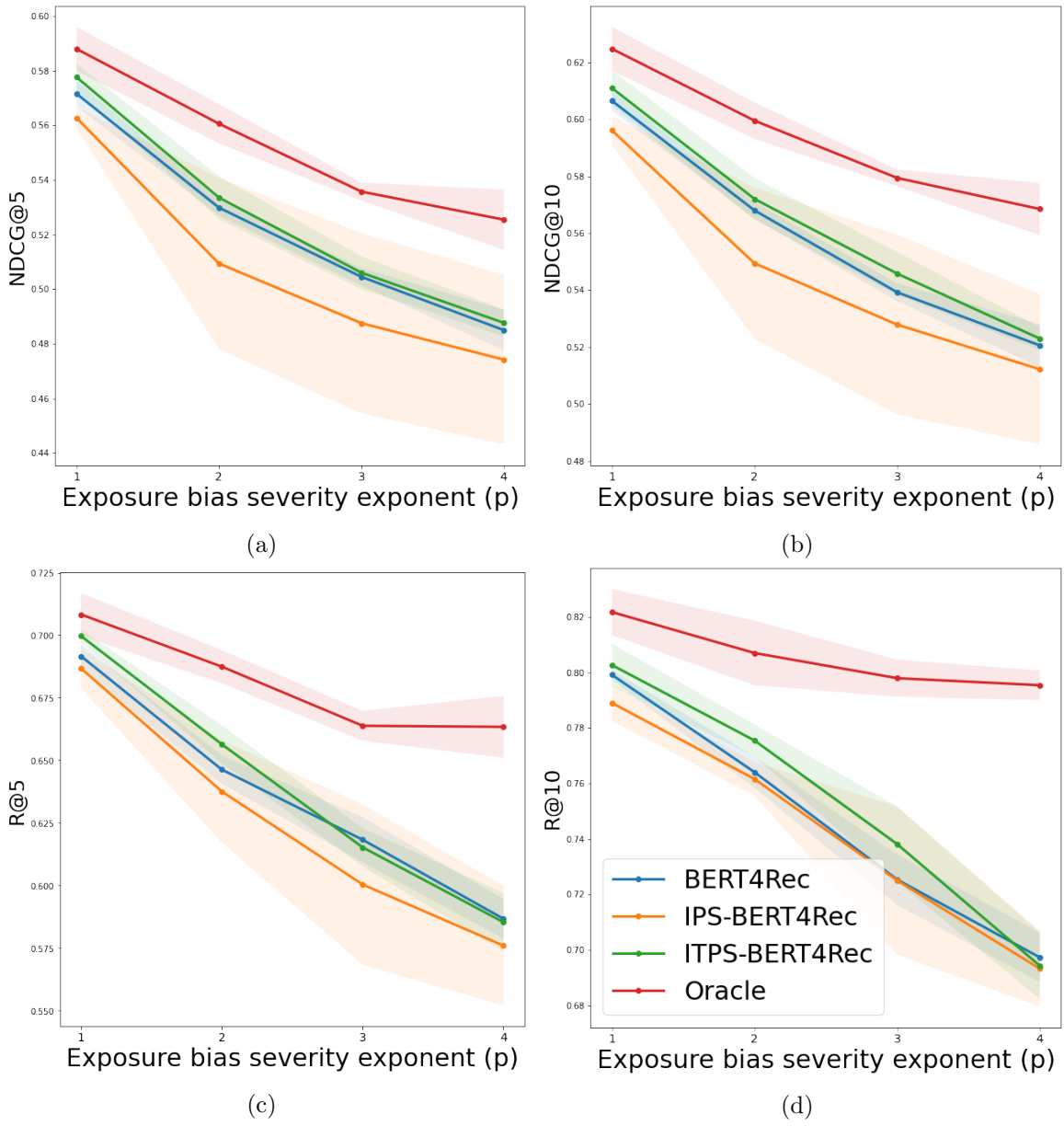


Figure 4.2: Robustness of the ranking performance - NDCG@5, NDCG@10, R@5, and R@10, in (a)-(d), respectively - of the different models to increasing levels of exposure bias. All the values are averages over 5 replicates and the 90% confidence intervals are highlighted. ITPS-BERT4Rec was the best in withstanding increasing levels of exposure bias overall.

TABLE 4.3

Average R@k and NDCG@k results over 5 replicates obtained with a standard evaluation process. The best results are in **bold** and second to best results are underlined. Arrows mean a change in the rank compared to the results from the unbiased evaluation in section 4.3.1.5. ↑ means the ranking increased and ↓ means the ranking decreased. A value with * is significantly higher than the next best value (p-value < 0.05).

Model	R@10	NDCG@10	R@5	NDCG@5
BERT4Rec	0.7782 ↓	0.5851 ↓	0.6655 ↓	0.5486
IPS-BERT4Rec	0.7835 ↑	0.5854 ↑	0.6665 ↑	0.5475
ITPS-BERT4Rec	0.7873*	0.5909*	0.6754*	0.5545
Oracle	**0.8000**	**0.5983**	**0.6795**	**0.5593**

findings validate the robustness of the proposed ITPS estimator in handling even extreme levels of exposure bias, and in capturing the true relevance of the items in a sequence and temporal context. Finally, in contrast to IPS-BERT4Rec, which shows a significantly high and increasing variance, ITPS-BERT4Rec shows a relatively low and steady variance that compares well to the variance of BERT4Rec. This further demonstrates the robustness of our proposed approach when facing increasing levels of exposure bias.

4.3.1.7 RQ3: How important is an unbiased evaluation in assessing exposure debiasing?

In this research question, we aim to demonstrate the importance of the unbiased evaluation process, explained in Section 4.3.1.2, in evaluating the capacity of the models to capture the true preferences of the users. To do so, we try to re-evaluate the tuned models using a standard Leave One Out (LOO) evaluation process, in which we compare the interacted test items to 100 randomly sampled items. This evaluation process is biased because the test items are not the most relevant items due to their exposure requirement. Thus, non-exposed items, possibly within the 100 randomly sampled items, could be more relevant within the same context, which engenders exposure bias. This results in an over-estimation of the ranking performance of the biased models, and their capacity to capture the true relevance of the items. We summarize the results obtained with the standard LOO

evaluation process in Table 4.3. We notice a discrepancy between the results obtained with the standard and unbiased evaluation processes. In fact, with the standard evaluation process, the IPS-BERT4Rec model outperformed BERT4Rec in almost all the settings, which reflects an over-estimation of the debiasing capabilities of the IPS framework and its ability to capture the relevance of items given the sequence context. The ITPS-BERT4Rec model was nonetheless still the top performer following the Oracle, although the difference between ITPS-BERT4Rec and the Oracle became insignificant. Thus, the debiasing performance of the ITPS framework was also inflated by the LOO evaluation process. These findings validate the necessity of relying on the unbiased evaluation setting in our experiments, as it allows us to truly evaluate the properties of the different estimators.

4.3.2 Experiments on Real Data

We perform offline experiments on real recommendation datasets that aim to answer the following research questions:

RQ4: How well does our proposed ITPS estimator perform in terms of ranking accuracy?

RQ5: How well does our proposed ITPS estimator help mitigate popularity bias in the short and long terms?

RQ6: How well does our proposed ITPS estimator help capture the temporal dependencies between items?

4.3.2.1 Data

We rely on three benchmark datasets that are commonly used in sequential recommendation research [4], which are the Movielens 1M (ml-1m)[1] [116], Movielens 20M (ml-20m)[2] [116], and Amazon Beauty (beauty)[3] [119]. For each of the datasets, we consider any rating, regardless of its value, as a positive interaction, then, we filter out users with

[1]https://grouplens.org/datasets/movielens/1m/
[2]https://grouplens.org/datasets/movielens/20m/
[3]https://nijianmo.github.io/amazon/index.html

TABLE 4.4

Real dataset statistics.

Dataset	Task	Sequences	Items	Interactions	Avg. length	Sparsity
ml-1m	Movie rec.	6,040	3,416	999,611	165.49	95.15%
ml-20m	Movie rec.	138,493	18,345	19,984,024	144.29	99.21%
beauty	Product rec.	40,226	54,542	353,962	8.79	99.98%

less than 5 interactions to reduce the data sparsity. The dataset properties and statistics are summarized in Table 4.4.

4.3.2.2 Evaluation and Propensity Estimation

As we mentioned in Section 4.3.1.2, the goal of the debiasing is to approximate the ideal loss which learns the ranking of items based on the true relevance. Theoretically, our proposed ITPS estimator relies on the temporal exposure propensities of the items to the users to provide an unbiased estimation of the relevance-based loss. Previously (Section 4.3.1), we were able to train our models using the true (temporal) exposure propensities and to evaluate their ability to model the relevance using the temporal relevance levels, which were available through the use of semi-synthetic data. However, in real-world datasets, neither the (temporal) exposure propensities, nor the temporal relevance levels are available. This causes the following two issues: (1) We cannot evaluate the models' ability to learn the true relevance of the items to the users because we do not know the true temporal relevance levels; and (2) we cannot train the IPS-BERT4Rec and ITPS-BERT4Rec models as they rely on the exposure and temporal exposure propensities.

To solve the first issue, we propose an evaluation process that is based on popularity-based negative sampling. In fact, the main issue with the standard LOO evaluation process is that some of the randomly sampled negative items to which we are comparing our test and validation items may be as relevant as, or possibly even more relevant than, the test and validation items. We propose to sample the negative items for every sequence based on

their popularity values, meaning the higher the popularity of an item, or in other words the more an item has been interacted with, the higher the probability that it will be sampled as a negative item for validation and testing. The idea is that more popular items have a higher propensity of being exposed, and hence have a higher likelihood that they have been exposed to the user and have not been interacted with because of their irrelevance to the user. The latter popularity-based negative sampling does not completely eliminate exposure bias in the evaluation. However, it is intended to mitigate it. Note that using popularity-based sampling to mitigate exposure bias was used in previous work [120] in the training phase. We are extending the approach to evaluation.

To solve the second issue, we extend the common procedure of estimating the exposure propensity by the item popularity [11, 121] by taking into consideration the temporal component. Thus, we estimate the temporal exposure propensity of an item to a user by the temporal popularity of the item such that:

$$\hat{\theta}_{S_s,I_i,t} = \frac{\sum_{j=1}^{|S|} Y_{S_j,I_i,t}}{\sum_{k=1}^{T}\sum_{l=1}^{|I|}\sum_{j=1}^{|S|} Y_{S_j,I_l,k}}. \tag{4.15}$$

Similarly, we estimate the static exposure propensity of an item in a sequence with the item's popularity, which corresponds to the sum of the estimated temporal exposure propensities of the item in the sequence over all the timesteps. Hence, the estimated static propensity is expressed as follows: $\hat{\theta}_{S_s,I_i} = \sum_{t=1}^{T} \hat{\theta}_{S_s,I_i,t}$.

Thus, we train the IPS-BERT4Rec and ITPS-BERT4Rec models, presented in section 4.3.1.3, using the estimated exposure propensities and estimated temporal exposure propensities, respectively.

4.3.2.3 Hyperparameter Tuning

For the beauty and ml-1m datasets, we perform the same hyperparameter tuning process described in Section 4.3.1.4 on the semi-synthetic dataset. However, for the ml-20m dataset, we increase the ranges of some of the hyperparameters given the relatively higher size and complexity of the dataset. In fact, more complex datasets require more complex

TABLE 4.5: Average Recall (R) and NDCG (N) results over 5 replicates on the three real interaction datasets that were described in Table 4. The best results are in **bold** and second to best results are underlined. A value with * is significantly higher than the next best value (p-value < 0.05).

Dataset	ml-1m				ml-20m				beauty			
Model	N@5	R@5	N@10	R@10	N@5	R@5	N@10	R@10	N@5	R@5	N@10	R@10
BERT4Rec	0.2820	0.4086	0.3262	0.5454	0.4205*	0.5583*	0.4624*	0.6876*	0.1056	0.1516	0.1260	0.2148
IPS-BERT4Rec	0.3416*	0.4751*	0.3801*	0.5940*	0.4004	0.5389	0.4434	0.6715	0.1053	0.1528	0.1268	0.2195
ITPS-BERT4Rec	**0.3451***	**0.4796**	**0.3844***	**0.6007***	**0.4295***	**0.5674***	**0.4709***	**0.6952***	**0.1197***	**0.1745***	**0.1444***	**0.2510***

models to fit them. Hence, the number of hidden units is tuned within {64, 128, 256}, the number of transformer blocks within {1, 2, 3}, the number of attention heads within {1, 2, 4, 8}, the batch size within {64, 128, 256}, and the dropout rate within {0, 0.01, 0.1, 0.2}.

4.3.2.4 RQ4: How well does the proposed ITPS estimator perform in terms of ranking accuracy?

To measure the ranking capabilities of the proposed approach, we train the models with their optimal hyperparameter configurations and evaluate them using the evaluation process presented in Section 4.3.2.2, which ensures that exposure bias is mitigated. Thus, the ranking accuracy results should provide a good approximation of how well the models capture the true relevance of the items to the users. We summarize the results on the three datasets in Table 4.5. Our proposed ITPS-BERT4Rec model was the best performer in all the settings, showing significantly superior performance than the vanilla BERT4Rec and the IPS-BERT4Rec models in all the metrics and on all the datasets. This validates the ability of the proposed ITPS debiasing framework to learn the true relevance of the items to the users, in addition to its applicability in real recommendation settings. Moreover, interestingly, the ranking performance was not consistent for the second to best model. In fact, IPS-BERT4Rec outperformed BERT4Rec overall in both the ml-1m and beauty datasets. However, BERT4Rec was the second to best model in the ml-20m dataset.

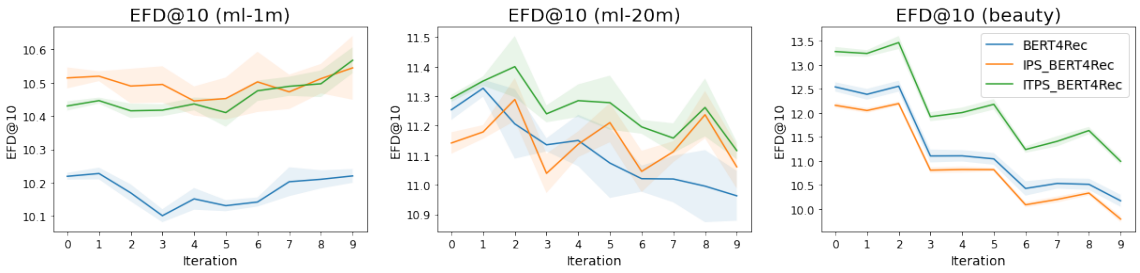
4.3.2.5 RQ5: How well does the proposed ITPS estimator help mitigate popularity bias in the short and long terms?

To assess the short and long term popularity debiasing effects of our proposed ITPS framework, we implement a feedback loop which simulates a real recommendation environment. The feedback loop consists of consecutive recommendation iterations where at each iteration, the recommender system is re-trained and generates top 10 recommendations to every user in the dataset. Each user then interacts with one of the recommended items and the interactions are added to the dataset for training future iterations. We simulate the user's choice for one of the recommended items with a uniform distribution, meaning that the interacted item is chosen at random from the recommendation list. Moreover, the choice of re-training the model at each iteration is related to the nature of our training datasets. In fact, we assume that an iteration corresponds to one day and that users interact with at most one movie or beauty product per day. This setting could be extended to other types of recommendation datasets in the future. Finally, we assume that all the users interact with one item at every iteration. As was discussed in [122], this assumption is meant to speed-up the feedback loop process and should not alter the general characteristics of the emerging phenomena. Thus, no conclusions will be altered. We evaluate the popularity debiasing capabilities by looking at the novelty of the top 10 recommendations. The novelty is assessed using the Expected Free Discovery (EFD) [113], which is a measure of the ability of a system to recommend relevant long-tail items [113] and is calculated as follows

$$EFD@\mathcal{K}(TopK) = -\frac{1}{|S|}\sum_{s=1}^{|S|}\frac{1}{\mathcal{K}}\sum_{i\in TopK(S_s)} log_2\hat{\theta}_{S_s,i} \tag{4.16}$$

where $TopK$ is the top $\mathcal{K}$ recommendation matrix in which every row represents the Top $\mathcal{K}$ recommendations in a sequence.

We summarize the evolution of $EFD@10$ for 10 feedback loop iterations on the three datasets in Figure 4.3. On both the ml-20m and beauty datasets, our proposed ITPS-BERT4Rec model showed the best results on all iterations. The difference in performance compared to the other two models was significant in all the iterations in the beauty dataset



(a) (b) (c)

Figure 4.3: Evolution of EFD@10 with respect to feedback loop iterations on the (a) ml-1m, (b) ml-20m, and (c) beauty datasets. All values are averages over 5 replicates and 90% confidence intervals are highlighted. ITPS-BERT4Rec showed the best short and long-term popularity debiasing capabilities on the ml-20m and beauty datasets.

and in most iterations in the ml-20m dataset. However, we notice a change in trend in the ml-1m dataset where IPS-BERT4Rec and ITPS-BERT4Rec showed a relatively similar popularity debiasing performance, that still outperformed the vanilla BERT4Rec model. We believe that the difference in trend in the ml-1m dataset is due to the relatively low number of items and low sparsity of the dataset making the popularity bias problem less prominent compared to the other datasets. Moreover and interestingly, the vanilla BERT4Rec outperformed IPS-BERT4Rec in terms of EFD@10 on the beauty dataset. The overall superior performance of our proposed ITPS-BERT4Rec model shows the impact of exposure debiasing on popularity debiasing, where modeling the true preferences of the user results in more diverse and novel recommendations yielding a higher item discovery by the user. Moreover, the ml-20m and beauty datasets showed decreasing trends for EFD with respect to the feedback loop iterations for all the models overall. This means that the issue of popularity bias tends to worsen with time. However, the relatively low slope of ITPS-BERT4Rec demonstrates the importance of mitigating exposure bias to mitigate long-term popularity bias.

4.3.2.6 RQ6: How well does our proposed ITPS estimator help capture the temporal dependencies between items?

As was discussed in Section 4.1.4 and illustrated in Figure 4.1, there exist temporal dependencies within the sequential recommendation data that connect the items. Thus, the fitness of a sequential recommender system can be evaluated through its capacity to capture those temporal dependencies, which consequently reflects on the quality of the sequential recommendations generated by the model. Given that the temporal propensities, that are used in our proposed ITPS debiasing framework, take into consideration the interaction timesteps of the items within the sessions, we can suspect that our proposed ITPS framework helps better capture the relative temporal dependencies between the items.

Hence, we rely on Temporal Association Rule Mining (TARM) [123] and propose the following evaluation process to assess the capacity of the different models in capturing the temporal dependencies between the items in the sequential recommendation data:

(1) We rely on the ml-1m dataset and normalize the sequences of interactions to the last 100 timesteps.

(2) We mine temporal association rules consisting of two items, in the form "$A \rightarrow B$", from the ml-1m dataset which have a minimum support of 0.5% and then sort them based on their lift value. The support of a rule is defined as the frequency of transactions (in our case sequences) that contain the rule. On the other hand, the lift of a rule "$A \rightarrow B$" is computed as the support of the rule "$A \rightarrow B$" divided by the product of the supports of the sub-rules "A" and "B". The mathematical formulations of support and lift are presented below:

$$Support(A) = \frac{|\{s \in [1, S] | A \in S_s\}|}{|S|} \tag{4.17}$$

$$Support(A \rightarrow B) = \frac{|\{s \in [1, S] | A \rightarrow B \subset S_s\}|}{|S|} \tag{4.18}$$

$$Lift(A \rightarrow B) = \frac{Support(A \rightarrow B)}{Support(A) \cdot Support(B)} \tag{4.19}$$

We rely on the T-Apriori algorithm [124], which is an adaptation of the Apriori [123]

TABLE 4.6: Top 10 temporal association rules extracted from the ml-1m dataset. The temporal association rules are sorted by their lift values and represent the temporal dependencies between the items within the interaction sequences.

Temporal Association Rule	Support	Lift
Friday the 13th Part VII: The New Blood (1988) → Friday the 13th Part VIII: Jason Takes Manhattan (1989)	0.0067	83.5492
Friday the 13th Part V: A New Beginning (1985) → Friday the 13th Part VI: Jason Lives (1986)	0.0074	70.7075
Child's Play 2 (1990) → Child's Play 3 (1992)	0.0069	48.5140
Superman III (1983) → Superman IV: The Quest for Peace (1987)	0.0054	44.4117
Halloween II (1981) → Halloween III: Season of the Witch (1983)	0.0076	44.0107
Three Colors: Blue (1993) → Three Colors: White (1994)	0.0074	38.5204
Poltergeist II: The Other Side (1986) → Poltergeist III (1988)	0.0074	38.2709
Friday the 13th: The Final Chapter (1984) → Friday the 13th Part V: A New Beginning (1985)	0.0062	37.0193
Friday the 13th Part 3: 3D (1982) → Friday the 13th: The Final Chapter (1984)	0.0117	36.3423
Halloween 4: The Return of Michael Myers (1988) → Halloween 5: The Revenge of Michael Myers (1989)	0.0059	34.8908

algorithm that takes into account the causal relationships between the items within the rules. More specifically, in this adaptation of the Apriori algorithm, the rules "$A \rightarrow B$" and "$B \rightarrow A$" are different. Hence, the generated temporal association rules capture the temporal dependencies between the movies in the dataset similarly to what was depicted in Figure 4.1. In this experiment, we focus on the short-term temporal dependencies between the items and only consider association rules of items that are consecutive in the sessions. Thus, we extract 2,157 temporal association rules from the ml-1m dataset. We present the top 10 extracted rules, sorted by their lift values in Table 4.6. Similarly to what was observed in Figure 4.1, the top 10 association rules all consist of movies followed by their sequels. Our goal in this research question is to evaluate the capacity of the different models to capture the temporal dependencies represented by these extracted temporal association rules.

(3) We rely on the feedback loop process which was presented in section 4.3.2.5 and perform 100 feedback loop iterations using each of the three models, BERT4Rec, IPS-BERT4Rec, and ITPS-BER4Rec, which result into new interaction sequences of 100 timesteps. Each generated dataset consists of 6,040 sequences of 100 timesteps, corresponding to each of the 6,040 uses in the ml-1m dataset, and simulates consecutive interactions with item recommendations generated using the corresponding model.

(4) We apply the T-Apriori algorithm with the same configuration as in step (1) on the three generated datasets from step (3) to extract temporal association rules sorted by

their lift values.

(5) We evaluate the capacity of a model to capture the temporal dependencies between items within the sequences by comparing the extracted temporal association rules from the generated datasets to the temporal association rules that were extracted from the ml-1m dataset in step (2). To do so, we rely on the three evaluation metrics that measure the rules' Precision at K ($rule\_P@K$), Average Precision at K ($rule\_AP@K$), and Normalized Discounted Cumulative Gain at K ($rule\_N@K$). The metrics are given below:

$$rule\_P@K(Rules_{data}, Rules_{model}) = \frac{|\{Rules_{data}\} \cap \{TopK(Rules_{model})\}|}{K} \tag{4.20}$$

$$rule\_AP@K(Rules_{data}, Rules_{model}) = \frac{1}{K} \sum_{k=1}^{K} rule\_P@k \cdot \mathbb{1}_{TopK(Rules_{model})[k] \in Rules_{data}} \tag{4.21}$$

$$rule\_N@K(Rules_{data}, Rules_{model}) = \frac{rule\_DCG@K(Rules_{data}, Rules_{model})}{rule\_IDCG@K(Rules_{data}, Rules_{model})} \tag{4.22}$$

where $rule\_DCG@K(Rules_{data}, Rules_{model})$ is the Discounted Cumulative Gain, which is divided by the Ideal $rule\_DCG$ ($rule\_IDCG@K(Rules_{data}, Rules_{model})$) for normalization purposes such that:

$$rule\_DCG@K(Rules_{data}, Rules_{model}) = \sum_{k=1}^{K} \frac{\mathbb{1}_{TopK(Rules_{model})[k] \in Rules_{data}}}{log_2(1+k)} \tag{4.23}$$

$$rule\_IDCG@K(Rules_{data}, Rules_{model}) = \sum_{k=1}^{K} \frac{1}{log_2(1+k)} \tag{4.24}$$

Where $Rules_{data}$ and $Rules_{model}$ are respectively the lists of temporal association rules extracted from the ml-1m dataset and the generated dataset using the feedback loop process with the corresponding model. The rules in $Rules_{data}$ and $Rules_{model}$ are sorted by their lift values. $TopK : Rules \mapsto TopK(Rules)$ is a function that filters the top K items from a list of rules $Rules$.

TABLE 4.7: Precision (rule_P), Average Precision (rule_AP), and Normalized Discounted Cumulative Gain (rule_N) results over 10 replicates for various cutoffs between the association rules extracted with the feedback loop process using the three models and the association rules extracted from the ml-1m dataset. The best results are in **bold** and second to best results are underlined. A value with * is significantly higher than the next best value (p-value < 0.05).

Cutoff	100			250			450			500		
Model	rule_P	rule_AP (10^{-4})	rule_N	rule_P	rule_AP (10^{-4})	rule_N	rule_P	rule_AP (10^{-4})	rule_N	rule_P	rule_AP (10^{-4})	rule_N
BERT4Rec	0	0	0	0	0	0	0.0006	0.0178	0.0005	0.0006	0.0160	0.0004
IPS-BERT4Rec	0	0	0	0	0	0	0	0	0	0.0002	0.0040	0.0001
ITPS-BERT4Rec	**0.002**	**0.3252**	**0.0016**	**0.0012***	**0.1571***	**0.0011***	**0.0011***	**0.1017***	**0.0011***	**0.001**	**0.0915***	**0.0010***

The above metrics are popular evaluation metrics for recommendation with explicit feedback [40] where the comparison is between a true list and a predicted list of rated items for a user. To avoid confusion with the task of recommendation, we add a "rule" to the name of each of these metrics because in this experiment, we are comparing lists of temporal association rules.

Note that $rule\_N@K$ and $rule\_AP@K$ are ranking metrics which, as opposed to $rule\_P@K$, assess the quality of the ranking of the temporal association rules. Also note that we did not use some of the well-established distances between rankings such as the Spearman's footrule [125] and Kendall's tau [126] because they fail to take into consideration the relevance of the items in the ranked lists in addition to their positions within the lists [127]. Instead, we rely on ranking metrics that are usually used in search and recommendation given that they are more adequate to our case.

(6) We repeat the experiment 10 times and summarize the average results for various values of the cutoff K in Table 4.7.

Our results show that ITPS-BERT4Rec outperformed the other two models on all metrics and in all settings, and its superior performance was significant in most of the settings. This means that ITPS-BERT4Rec was the best in capturing the temporal dependencies between the items within the sessions. This is certainly due to the fact that ITPS-BERT4Rec relies on the temporal exposures of items to users which, additionally to mitigating exposure bias in the Cloze task, also help model the temporal relationships between the items within the sessions. This shows that our proposed ITPS debiasing frame-

work helps achieve unbiased recommendations which better match the user's preferences in addition to successions of consecutive item recommendations with consistent ordinal relationships. Moreover, the vanilla BERT4Rec outperformed IPS-BER4Rec in all the settings. This means that using the static exposure propensities of items to users hinders the capacity of the sequential model to capture the temporal relationships within the sessions. **Thus debiasing a sequential recommender system without taking into consideration the temporal component in the data can hurt the pattern modeling capabilities of the recommender system**.

Note that we started with a cutoff of 100 because there was no match between the temporal association rules extracted from the data and from the datasets generated by the different models for small cutoffs. Yet, ITPS-BERT4Rec showed a match between the two lists of association rules for the smallest cutoff of 70. This is probably due to the relatively high number of items in the dataset which is coupled with a high sparsity, resulting in a sparsity in the extracted association rules. This can be observed in the relatively low support values in Table 4.6. Also note that BERT4Rec and IPS-BERT4Rec only started to show a match between the temporal association rules at relatively high cutoffs of around 450 and 500 respectively, which further reflects the advantage of using the temporal propensities in capturing the temporal dependencies between the items within the sequential data.

4.4 Chapter Summary

In this chapter, we started by formulating the problem of sequential recommendation with bidirectional transformers and formally introduced the Cloze task. Then, we focused on exposure bias and started by defining the ideal Cloze task loss function that we aim to estimate. This led us to prove that the Cloze task loss is biased against the ideal loss function. Then, we studied the applicability of the Inverse Propensity Scoring (IPS) framework in debiasing sequential recommendation approaches and unveiled its limitations, concluding on its inability to eliminate exposure bias in this context. Thus, we proposed a novel exposure debiasing framework called Inverse Temporal Propensity Scoring (ITPS), which we

theoretically proved to eliminate exposure bias in sequential recommendation with bidirectional transformers. Finally, we conducted experiments which empirically demonstrated the advantages of our proposed approach in terms of mitigating exposure bias, robustness to increasing levels of exposure bias, mitigating popularity bias in the short and long terms, and capturing the temporal dependencies between items within the sequences of interactions.

In fact, our proposed ITPS-BERT4Rec approach has demonstrated a significant increase of 1% in terms of modeling the true preferences of the user in a semi-synthetic setting over the state-of-the-art BERT4Rec model. Similarly, ITPS-BERT4Rec showed an average increase of 8.7% over BERT4Rec in three real-world recommendation settings. Furthermore, empirical experiments demonstrated the robustness of our proposed ITPS-BERT4Rec model to increasing levels of exposure bias and its relatively low variance. Additionally, experiments on popularity debiasing showed a significant advantage for our proposed ITPS-BERT4Rec model in both the short and long terms. Finally, ITPS-BERT4Rec showed respective improvements of around 60%, 470%, and 150% over BERT4Rec in capturing the temporal dependencies between the items within the sequences of interactions for three different evaluation metrics. These results demonstrate the capabilities of our proposed unbiased estimator in ameliorating the user experience in the context of sequential recommendation by presenting them with more accurate and diverse recommendations that better match their true preferences and the sequential dependencies between the recommended items.

CHAPTER 5

CONCLUSION

We introduced novel approaches that aim to improve the user's experience with recommender systems. Our proposed work is centered around the three objectives of accuracy, explainability, and unbiasedness and spans two fundamental recommendation settings, namely recommendation with user profiles and sequential recommendation.

First, we proposed novel approaches that aim to promote explainability and mitigate exposure bias in recommendation with user profiles. We started by proposing a novel explainable pairwise ranking loss with a corresponding MF-based model called Explainable Bayesian Personalized Ranking. We theoretically quantified the additional exposure bias resulting from the explainability, and proposed an IPS-based unbiased estimator for the ideal loss. We tested our proposed approaches on three recommendation tasks and presented an extensive discussion about the advantages of the proposed explainability extension; as well as the impact of the debiasing, for varying data sparsities and varying neighborhood sizes. Then, we studied the popularity-debiasing properties of the proposed methods in terms of Novelty, Popularity and Diversity, and unveiled an inherent popularity debiasing stemming from the neighborhood interactions.

Our findings are informative and motivate further research. In fact, our EBPR model yielded a high prediction performance, characterized by an increase in ranking accuracy of about 4% over the baseline BPR model, with no significant trade-off between explainability and accuracy. In fact, EBPR also showed an improvement of about 7% in terms of explainability. Moreover, we showed how combining explainability and exposure debiaing yields powerful popularity debiasing through the proposed UEBPR loss, characterized by an improvement of over 10% overall in popularity debiasing metrics over the unbiased UBPR

model. Additionally, coupling explainability with exposure debiasing was also shown to help capture the true preferences of the user with a significant improvement of 1% over the baseline unbiased model UBPR.

Second, we studied the problem of exposure bias in sequential recommendation within the scope of bidirectional transformers trained to optimize the Cloze task, and proposed an ideal Cloze task loss that captures the true relevance. Then, we argued and proved that Inverse Propensity Scoring estimators do not extend to sequential recommendation. In addition, we proposed a theoretically unbiased estimator for the ideal Cloze task loss, and formulated a framework that allows for an unbiased training and evaluation of sequential recommender systems. Our experiments empirically validated our claims of exposure debiasing of the proposed ITPS-BERT4Rec estimator through experiments on semi-synthetic and real-world datasets which aimed to assess the capacity to capture the true preferences of the user. Moreover, our experiments demonstrated the robustness of our proposed ITPS-BERT4Rec model to increasing levels of exposure bias, along with its long term impact on popularity debiasing.

Our proposed ITPS debiasing framework was able to improve the capabilities of the recommender system to capture the true relevance of the items to the users. The latter contribution is characterized by a significant increase of 1% in terms of modeling the true preferences of the user in a semi-synthetic recommendation setting, and an average increase of 8.7% in three real-world recommendation settings over the state-of-the-art sequential recommendation model BERT4Rec. Moreover, ITPS-BERT4Rec was able to improve the item discovery in the recommendations in both the short and long terms; in addition to capturing the temporal dependencies between the items within the sequences of interactions, resulting in respective improvements of around 60%, 470%, and 150% over vanilla BERT4Rec in three different evaluation settings. Finally, empirical experiments demonstrated the robustness of our proposed ITPS-BERT4Rec model to increasing levels of exposure bias and its stability in terms of variance.

Despite their many advantages, our proposed approaches suffer a few limitations. For

instance, as was empirically demonstrated, our proposed explainability weighting technique cannot handle extreme sparseness in the data and leads to a vanishing gradient problem which results in a degradation in ranking performance. Furthermore, our proposed debiasing frameworks in both recommendation with user profiles and sequential recommendation rely on a few assumptions that need to be further justified. For instance, our proposed unbiased UEBPR estimator's unbiasedness relies on the assumption of conditional independence between exposure and relevance given the neighborhood. Also, the unbiasedness of our proposed ITPS-based Cloze task estimator is contingent upon the independence between exposure and relevance, in addition to the rationality of the users in terms of interacting with the most relevant item that is exposed to them.

In the future, we plan to build on our theoretical and experimental findings to:

1. Theoretically investigate the inherent debiasing properties of our proposed explainability weighting term in recommendation with user profiles.

2. Investigate how our proposed unbiased estimators, in recommendation with user profiles, approximate the ideal losses empirically, in a semi-synthetic setting, as was performed in [22] and [7], where the relevance of items to users is estimated using matrix completion algorithms.

3. Validate and challenge the assumptions on which our debiasing approaches are based in both recommendation with user profiles and sequential recommendation, and study their impact on exposure bias.

4. Evaluate the proposed models, both in recommendation with user profiles and sequential recommendation, in real deployments.

5. Further propose and implement new techniques that aim to mitigate different types of bias in recommendation from implicit feedback.

REFERENCES

[1] Yifan Hu, Yehuda Koren, and Chris Volinsky, "Collaborative filtering for implicit feedback datasets," in *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 2008, pp. 263–272.

[2] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[3] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[4] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[6] Ruining He and Julian McAuley, "Vbpr: visual bayesian personalized ranking from implicit feedback," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016, vol. 30.

[7] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims, "Recommendations as treatments: Debiasing learning and evaluation," *arXiv preprint arXiv:1602.05352*, 2016.

[8] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He, "Bias and debias in recommender system: A survey and future directions," *arXiv preprint arXiv:2010.03240*, 2020.

[9] Mustafa Bilgic and Raymond J Mooney, "Explaining recommendations: Satisfaction vs. promotion," in *Beyond Personalization Workshop, IUI*, 2005, vol. 5, p. 153.

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

[11] Yuta Saito, "Unbiased pairwise learning from implicit feedback," in *NeurIPS 2019 Workshop on Causal Machine Learning*, 2019.

[12] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Defu Lian, "A survey on session-based recommender systems," *arXiv preprint arXiv:1902.04864*, 2019.

[13] Zachary Chase Lipton, "A critical review of recurrent neural networks for sequence learning," *CoRR*, vol. abs/1506.00019, 2015.

[14] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1724–1734, Association for Computational Linguistics.

[15] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[16] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[17] Balázs Hidasi and Alexandros Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations," in *Proceedings of the 27th ACM international conference on information and knowledge management*, 2018, pp. 843–852.

[18] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio, "Object recognition with gradient-based learning," in *Shape, contour and grouping in computer vision*, pp. 319–345. Springer, 1999.

[19] Jiaxi Tang and Ke Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 565–573.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[21] Wang-Cheng Kang and Julian McAuley, "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 197–206.

[22] 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*, 2020, pp. 501–509.

[23] Wilson L Taylor, ""cloze procedure": A new tool for measuring readability," *Journalism quarterly*, vol. 30, no. 4, pp. 415–433, 1953.

[24] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[25] Yue Shi, Martha Larson, and Alan Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, pp. 1–45, 2014.

[26] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175–186.

[27] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 230–237.

[28] Jiyong Zhang and Pearl Pu, "A recursive prediction algorithm for collaborative filtering recommender systems," in *Proceedings of the 2007 ACM conference on Recommender systems*, 2007, pp. 57–64.

[29] Yue Shi, Martha Larson, and Alan Hanjalic, "Exploiting user similarity based on rated-item pools for improved user-based collaborative filtering," in *Proceedings of the third ACM conference on Recommender systems*, 2009, pp. 125–132.

[30] Mukund Deshpande and George Karypis, "Item-based top-n recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, jan 2004.

[31] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[32] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, New York, NY, USA, 2001, WWW '01, p. 285–295, Association for Computing Machinery.

[33] Manh Cuong Pham, Yiwei Cao, Ralf Klamma, and Matthias Jarke, "A clustering approach for collaborative filtering recommendation using social network analysis.," *J. Univers. Comput. Sci.*, vol. 17, no. 4, pp. 583–604, 2011.

[34] Yehuda Koren, Robert Bell, and Chris Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[35] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[36] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua, "Outer product-based neural collaborative filtering," *arXiv preprint arXiv:1808.03912*, 2018.

[37] Hao Wang, Naiyan Wang, and Dit-Yan Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1235–1244.

[38] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson, "Neural collaborative filtering vs. matrix factorization revisited," in *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 240–248.

[39] Alexandros Karatzoglou, Linas Baltrunas, and Yue Shi, "Learning to rank for recommender systems," in *Proceedings of the 7th ACM Conference on Recommender Systems*, 2013, pp. 493–494.

[40] Khalil Damak, Olfa Nasraoui, and William Scott Sanders, "Sequence-based explainable hybrid song recommendation," *Frontiers in Big Data*, vol. 4, pp. 57, 2021.

[41] Hsiang-Fu Yu, Mikhail Bilenko, and Chih-Jen Lin, "Selection of negative samples for one-class matrix factorization," in *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 2017, pp. 363–371.

[42] Rong Pan and Martin Scholz, "Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 667–676.

[43] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang, "One-class collaborative filtering," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 502–511.

[44] Ulrich Paquet and Noam Koenigstein, "One-class collaborative filtering with random graphs," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 999–1008.

[45] Yue Shi, Martha Larson, and Alan Hanjalic, "List-wise learning to rank with matrix factorization for collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 269–272.

[46] Shanshan Huang, Shuaiqiang Wang, Tie-Yan Liu, Jun Ma, Zhumin Chen, and Jari Veijalainen, "Listwise collaborative filtering," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 343–352.

[47] Liwei Wu, Cho-Jui Hsieh, and James Sharpnack, "Sql-rank: A listwise approach to collaborative ranking," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5315–5324.

[48] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z Sheng, Mehmet A Orgun, and Defu Lian, "A survey on session-based recommender systems," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1–38, 2021.

[49] Jiawei Han, Jian Pei, and Yiwen Yin, "Mining frequent patterns without candidate generation," *ACM sigmod record*, vol. 29, no. 2, pp. 1–12, 2000.

[50] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa, "Effective personalization based on association rule discovery from web usage data," in *Proceedings of the 3rd international workshop on Web information and data management*, 2001, pp. 9–15.

[51] Shoujin Wang and Longbing Cao, "Inferring implicit rules by learning explicit and hidden item dependency," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 3, pp. 935–946, 2017.

[52] Rana Forsati, Mohammad Reza Meybodi, and A Ghari Neiat, "Web page personalization based on weighted association rules," in *2009 International Conference on Electronic Computer Technology*. IEEE, 2009, pp. 130–135.

[53] Liang Yan and Chunping Li, "Incorporating pageview weight into an association-rule-based web recommendation system," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2006, pp. 577–586.

[54] María N Moreno, Francisco J García, M José Polo, and Vivian F López, "Using association analysis of web data in recommender systems," in *International Conference on Electronic Commerce and Web Technologies*. Springer, 2004, pp. 11–20.

[55] Bo Shao, Dingding Wang, Tao Li, and Mitsunori Ogihara, "Music recommendation based on acoustic features and user access patterns," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 8, pp. 1602–1611, 2009.

[56] Wei Song and Kai Yang, "Personalized recommendation based on weighted sequence similarity," in *Practical Applications of Intelligent Systems*, pp. 657–666. Springer, 2014.

[57] Malte Ludewig and Dietmar Jannach, "Evaluation of session-based recommendation algorithms," *User Modeling and User-Adapted Interaction*, vol. 28, no. 4-5, pp. 331–390, 2018.

[58] Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff, "Sequence and time aware neighborhood for session-based recommendations: Stan," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 1069–1072.

[59] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang, "Modeling personalized item frequency information for next-basket recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1071–1080.

[60] Magdalini Eirinaki, Michalis Vazirgiannis, and Dimitris Kapogiannis, "Web path recommendations based on page ranking and markov models," in *Proceedings of the 7th annual ACM international workshop on Web information and data management*, 2005, pp. 2–9.

[61] Zhiyong Zhang and Olfa Nasraoui, "Efficient hybrid web recommendations based on markov clickstream models and implicit search," in *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*. IEEE, 2007, pp. 621–627.

[62] Duc-Trong Le, Yuan Fang, and Hady W Lauw, "Modeling sequential preferences with dynamic user and context factors," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 145–161.

[63] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th International Conference on World Wide Web*, New York, NY, USA, 2010, WWW '10, p. 811–820, Association for Computing Machinery.

[64] Xiang Wu, Qi Liu, Enhong Chen, Liang He, Jingsong Lv, Can Cao, and Guoping Hu, "Personalized next-song recommendation in online karaokes," in *Proceedings of the 7th ACM Conference on Recommender Systems*, New York, NY, USA, 2013, RecSys '13, p. 137–140, Association for Computing Machinery.

[65] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan, "Personalized ranking metric embedding for next new poi recommendation," 2015.

[66] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King, "Where you like to go next: Successive point-of-interest recommendation," in *Twenty-Third international joint conference on Artificial Intelligence*, 2013.

[67] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei, "Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 59–66.

[68] Defu Lian, Vincent W Zheng, and Xing Xie, "Collaborative filtering meets next check-in location prediction," in *Proceedings of the 22nd International Conference on World Wide Web*, 2013, pp. 231–232.

[69] Xin Liu, Yong Liu, Karl Aberer, and Chunyan Miao, "Personalized point-of-interest recommendation by mining users' preference transition," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013, pp. 733–738.

[70] Shoujin Wang, Liang Hu, and Longbing Cao, "Perceiving the next choice with comprehensive transaction embeddings for online recommendation," in *Machine Learning and Knowledge Discovery in Databases*, Michelangelo Ceci, Jaakko Hollmén, Ljupčo Todorovski, Celine Vens, and Sašo Džeroski, Eds., Cham, 2017, pp. 285–302, Springer International Publishing.

[71] Flavian Vasile, Elena Smirnova, and Alexis Conneau, "Meta-prod2vec: Product embeddings using side-information for recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems*, New York, NY, USA, 2016, RecSys '16, p. 225–232, Association for Computing Machinery.

[72] Dongjing Wang, Shuiguang Deng, Xin Zhang, and Guandong Xu, "Learning music embedding with metadata for context aware recommendation," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, New York, NY, USA, 2016, ICMR '16, p. 249–253, Association for Computing Machinery.

[73] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao, "Atrank: An attention-based user behavior modeling framework for recommendation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[74] Shuai Zhang, Yi Tay, Lina Yao, and Aixin Sun, "Next item recommendation with self-attention," *arXiv preprint arXiv:1808.06414*, 2018.

[75] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge, "Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation," in *Fifteenth ACM Conference on Recommender Systems*, 2021, pp. 143–153.

[76] William Merrill, Yoav Goldberg, and Noah A Smith, "On the power of saturated transformers: A view from circuit complexity," *arXiv preprint arXiv:2106.16213*, 2021.

[77] Adrian M. P. Braşoveanu and Răzvan Andonie, "Visualizing transformers for nlp: A brief survey," in *2020 24th International Conference Information Visualisation (IV)*, 2020, pp. 270–279.

[78] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[79] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," in *International Conference on Learning Representations*, 2020.

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

[81] Yongfeng Zhang, "Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation," in *Proceedings of the eighth ACM international conference on web search and data mining*, 2015, pp. 435–440.

[82] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua, "Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention," in *Proceedings of the 40th International ACM SIGIR*

conference on Research and Development in Information Retrieval, 2017, pp. 335–344.

[83] Xu Chen, Yongfeng Zhang, Hongteng Xu, Yixin Cao, Zheng Qin, and Hongyuan Zha, "Visually explainable recommendation," *arXiv preprint arXiv:1801.10288*, 2018.

[84] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu, "Interpretable convolutional neural networks with dual local and global attention for review rating prediction," in *Proceedings of the eleventh ACM conference on recommender systems*, 2017, pp. 297–305.

[85] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam, "Neural rating regression with abstractive tips generation for recommendation," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017, pp. 345–354.

[86] Bashir Rastegarpanah, Mark Crovella, and Krishna P Gummadi, "Exploring explanations for matrix factorization recommender systems," 2017.

[87] Weiyu Cheng, Yanyan Shen, Linpeng Huang, and Yanmin Zhu, "Incorporating interpretability into latent factor models via fast influence analysis," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 885–893.

[88] 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*, 2000, pp. 241–250.

[89] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[90] Ludovik Coba, Panagiotis Symeonidis, and Markus Zanker, "Personalised novel and explainable matrix factorisation," *Data & Knowledge Engineering*, vol. 122, pp. 142–158, 2019.

[91] Shuo Wang, Hui Tian, Xuzhen Zhu, and Zhipeng Wu, "Explainable matrix factorization with constraints on neighborhood in the latent space," in *International Conference on Data Mining and Big Data*. Springer, 2018, pp. 102–113.

[92] Huafeng Liu, Jingxuan Wen, Liping Jing, Jian Yu, Xiangliang Zhang, and Min Zhang, "In2rec: Influence-based interpretable recommendation," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1803–1812.

[93] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin, "Unbiased offline recommender evaluation for missing-not-at-random implicit feedback," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 279–287.

[94] Robin Devooght, Nicolas Kourtellis, and Amin Mantrach, "Dynamic matrix factorization with priors on unknown values," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 189–198.

[95] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 549–558.

[96] Yanen Li, Jia Hu, ChengXiang Zhai, and Ye Chen, "Improving one-class collaborative filtering by incorporating rich user information," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, 2010, pp. 959–968.

[97] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Yan Feng, and Chun Chen, "Samwalker: Social recommendation with informative sampling strategy," in *The World Wide Web Conference*, 2019, pp. 228–239.

[98] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua, "Reinforced negative sampling over knowledge graph for recommendation," in *Proceedings of The Web Conference 2020*, 2020, pp. 99–109.

[99] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei, "Modeling user exposure in recommendation," in *Proceedings of the 25th international conference on World Wide Web*, 2016, pp. 951–961.

[100] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Jingbang Chen, Yan Feng, and Chun Chen, "Fast adaptively weighted matrix factorization for recommendation with implicit feedback.," in *AAAI*, 2020, pp. 3470–3477.

[101] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani, "Large-scale causal approaches to debiasing post-click conversion rate estimation with multi-task learning," in *Proceedings of The Web Conference 2020*, 2020, pp. 2775–2781.

[102] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai, "Entire space multi-task model: An effective approach for estimating post-click conversion rate," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 1137–1140.

[103] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang, "Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 2377–2386.

[104] Wentian Bao, Hong Wen, Sha Li, Xiao-Yang Liu, Quan Lin, and Keping Yang, "Gmcm: Graph-based micro-behavior conversion model for post-click conversion rate estimation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 2201–2210.

[105] Pengyu Zhao, Tianxiao Shui, Yuanxing Zhang, Kecheng Xiao, and Kaigui Bian, "Adversarial oracular seq2seq learning for sequential recommendation," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*, 2020, pp. 1905–1911.

[106] Ruiyang Ren, Zhaoyang Liu, Yaliang Li, Wayne Xin Zhao, Hui Wang, Bolin Ding, and Ji-Rong Wen, "Sequential recommendation with self-attentive multi-adversarial network," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 89–98.

[107] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba, "Sequence level training with recurrent neural networks," *arXiv preprint arXiv:1511.06732*, 2015.

[108] F Maxwell Harper and Joseph A Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[109] Yahoo!, "Yahoo! Webscope dataset ydata-ymusic-rating-study-v1.0-train," `http://research.yahoo.com/Academic_Relations`.

[110] Last.FM, "hetrec2011-lastfm-2k," `https://grouplens.org/datasets/hetrec-2011/`.

[111] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik, "2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011)," in *Proceedings of the 5th ACM conference on Recommender systems*, New York, NY, USA, 2011, RecSys 2011, ACM.

[112] Behnoush Abdollahi and Olfa Nasraoui, "Explainable matrix factorization for collaborative filtering," in *Proceedings of the 25th International Conference Companion on World Wide Web*, 2016, pp. 5–6.

[113] Saúl Vargas and Pablo Castells, "Rank and relevance in novelty and diversity metrics for recommender systems," in *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 109–116.

[114] Sunitha Basodi, Chunyan Ji, Haiping Zhang, and Yi Pan, "Gradient amplification: An efficient way to train deep neural networks," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 196–207, 2020.

[115] Shantanu Gupta, Hao Wang, Zachary Lipton, and Yuyang Wang, "Correcting exposure bias for link recommendation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 3953–3963.

[116] F. Maxwell Harper and Joseph A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, dec 2015.

[117] Jianli Zhao, Shangcheng Yang, Huan Huo, Qiuxia Sun, and Xijiao Geng, "Tbtf: an effective time-varying bias tensor factorization algorithm for recommender system," *Applied Intelligence*, pp. 1–12, 2021.

[118] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems (TOIS)*, vol. 23, no. 1, pp. 103–145, 2005.

[119] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 2015, pp. 43–52.

[120] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme, "Personalized ranking for non-uniformly sampled items," in *Proceedings of KDD Cup 2011*. PMLR, 2012, pp. 231–247.

[121] Khalil Damak, Sami Khenissi, and Olfa Nasraoui, "Debiased explainable pairwise ranking from implicit feedback," in *Fifteenth ACM Conference on Recommender Systems*, 2021, pp. 321–331.

[122] Andres Ferraro, Dietmar Jannach, and Xavier Serra, "Exploring longitudinal effects of session-based recommendations," in *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 474–479.

[123] Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh, "Algorithms for association rule mining—a general survey and comparison," *ACM sigkdd explorations newsletter*, vol. 2, no. 1, pp. 58–64, 2000.

[124] Zhai Liang, Tang Xinming, Li Lin, and Jiang Wenliang, "Temporal association rule mining based on t-apriori algorithm and its typical application," in *Proceedings of International Symposium on Spatio-temporal Modeling, Spatial Reasoning, Analysis, Data Mining and Data Fusion*. Citeseer, 2005.

[125] Persi Diaconis and R. L. Graham, "Spearman's footrule as a measure of disarray," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 2, pp. 262–268, 1977.

[126] Maurice G Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[127] Ravi Kumar and Sergei Vassilvitskii, "Generalized distances between rankings," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 571–580.

CURRICULUM VITAE

NAME: Khalil Damak

ADDRESS: Computer Science & Engineering Department

Speed School of Engineering

University of Louisville

Louisville, KY 40292

EDUCATION:

Ph.D., Computer Science

May 2022

University of Louisville, *Louisville, Kentucky*

M.S., Computer Science & Engineering

May 2019

University of Louisville, *Louisville, Kentucky*

B.Eng., Polytechnic Engineering

June 2017

Tunisia Polytechnic School, *Tunis, Tunisia*

RESEARCH & WORK EXPERIENCE

1. Gradiate Teaching Assistant, **University of Louisville**, August 2021 - May 2022
2. Applied Scientist Intern, **Amazon**, May 2021 - August 2021
3. Graduate Research Assistant, **University of Louisville**, April 2017 - May 2022

4. Software Engineer Intern, **Banque Internationale Arabe de Tunisie**, July 2016 - August 2016

PUBLICATIONS:

1. **Damak, K.**, Khenissi, S. and Nasraoui, O., 2022. A framework for unbiased explainable pairwise ranking for recommendation. Software Impacts, 11, p.100208.

2. **Damak, K.**, Khenissi, S. and Nasraoui, O., 2021, September. Debiased explainable pairwise ranking from implicit feedback. In Fifteenth ACM Conference on Recommender Systems (pp. 321-331).

3. **Damak, K.**, Nasraoui, O. and Sanders, W.S., 2021. Sequence-Based Explainable Hybrid Song Recommendation. Frontiers in big Data, p.57.

4. Boujelbene, M., **Damak, K.**, Sener, A.C.A., Hieb, J.L., Bego, C.R., Ralston, P.A. and Nasraoui, O., 2020, June. A Data-science Approach to Flagging Non-retention in Engineering Enrollment Data. In 2020 ASEE Virtual Annual Conference Content Access.

HONORS AND AWARDS:

1. Graduate Dean's Citation, April 2022

2. CECS Arthur M. Riehl Award, April 2019

3. Graduate Dean's Citation, April 2019

4. Second prize at SPEED STUDENT RESEARCH EXPOSITION, April 2019

5. Doctoral Fellowship, UofL, March 2019

6. Tunisian National Scholarship for Engineering Studies, September 2014

7. PHI KAPPA PHI Honor Society membership, March 2020