Software Impacts 11 (2022) 100208

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

A framework for unbiased explainable pairwise ranking for recommendation

Khalil Damak *, Sami Khenissi, Olfa Nasraoui

Knowledge Discovery and Web Mining Lab, Department of Computer Science and Engineering, University of Louisville, United States of America

ARTICLE INFO

Keywords:
Recommender systems
Fairness in AI
Debiased machine learning
Pairwise ranking
Explainability
Exposure bias

ABSTRACT

Recent research in recommender systems has demonstrated the advantages of pairwise ranking in recommendation. In this work, we focus on the state-of-the-art pairwise ranking loss function, Bayesian Personalized Ranking (BPR), and aim to address two of its limitations, namely: (1) the lack of explainability and (2) exposure bias. We propose a recommendation framework that encompasses various loss functions that are based on BPR and which aim to mitigate the aforementioned limitations. Our open-source framework includes code to train and tune state-of-the-art pairwise ranking recommender systems on benchmark datasets and evaluate them based on the three criteria of ranking accuracy, explainability, and popularity debiasing.

Code metadata

Current code version	v1
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2021-158
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/7889543/tree/v1
Legal Code License	GNU General Public License v3.0
Code versioning system used	Git
Software code languages, tools, and services used	We use the Python machine learning framework PyTorch 1.7.1.
Compilation requirements, operating environments & dependencies	
If available Link to developer documentation/manual	https://github.com/KhalilDMK/EBPR/blob/main/README.md
Support email for questions	khalil.damak@louisville.edu

1. Introduction

Bayesian Personalized Ranking (BPR) is a pairwise ranking approach [1] that has recently received significant praise in the recommender systems community because of its capacity to rank implicit feedback data with high accuracy [2]. Aiming to rank relevant items higher than irrelevant items, pairwise ranking recommender systems often assume that all non-interacted items are irrelevant. The latter assumption engenders exposure bias, which is a notorious issue in recommendation from implicit feedback, and that is usually characterized by a bias against less popular items having a lower propensity of being observed [3].

Moreover, most state-of-the-art recommender systems, including BPR, are black boxes that do not justify why or how an item was recommended to a user. This might engender unfairness issues if particularly inappropriate content gets recommended to a user. In this case, knowing why an item was recommended might help diagnose the recommendation and mitigate the unfairness. Moreover, the lack of explainability may limit the capability of the user to make an informed decision when choosing to follow the recommendation. In fact, explanations bring more context based on which the user will make a decision, which was shown in earlier work to increase the user satisfaction [4,5].

In our previous work [6], we proposed novel loss functions for pairwise ranking recommendation, which aim to improve the explainability of BPR and mitigate exposure bias. In this article, we present our related open source framework which allows to train, evaluate and tune a Matrix Factorization [7] (MF) model with those proposed loss functions [6]. Our framework aims to facilitate incorporating explainability and exposure debiasing into pairwise ranking models for recommendation. We also make it easy to implement new models, in

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail address: khalil.damak@louisville.edu (K. Damak).

https://doi.org/10.1016/j.simpa.2021.100208
Received 7 November 2021; Received in revised form 13 December 2021; Accepted 18 December 2021

addition to matrix factorization, hence extending the scope of use of our framework. Finally, our framework provides an evaluation pipeline with metrics that assess the performance of the model in terms of ranking accuracy, explainability, and popularity debiasing. In the following sections, we will delve in more detail about the different characteristics and functionalities of our framework.

2. Description

In this section, we start by describing the loss functions and machine learning models included in our proposed framework. Then, we describe our framework's functionalities, namely the training and hyperparameter tuning of the different approaches.

2.1. Loss functions

Our proposed framework aims to train, tune, evaluate, and compare machine learning models for pairwise ranking recommendation with various degrees of explainability and exposure debiasing. The latter degrees of explainability and debiasing are related to the various loss functions that were discussed in our previous work [6], and that are implemented in our proposed recommendation framework. The following loss functions are implemented:

- **Bayesian Personalized Ranking (BPR)** [1]: This is the vanilla BPR loss that was proposed in [1]. This loss function aims to rank interacted items higher than non-interacted items for a given user.
- **Unbiased Bayesian Personalized Ranking (UBPR)** [8]: This is an unbiased version of the BPR loss function proposed in [8]. This approach relies on Inverse Propensity Scoring (IPS) [9] to theoretically eliminate the exposure bias in the BPR loss.
- **Explainable Bayesian Personalized Ranking (EBPR):** This is our proposed explainable BPR loss function [6]. This loss function is based on BPR and relies on neighborhood-based explainability [5,10,11] to rank relevant and explainable recommendations at the top of the recommendation list for a user. The explanations in this case are in the form "This item was recommended because you also liked these similar items.".
- **partially Unbiased Explainable Bayesian Personalized Ranking (pUEBPR):** This is a loss function that we proposed in [6] for partially unbiased and explainable BPR. In this loss function, we use IPS to eliminate the original BPR exposure bias similarly to UBPR. However, as was proven in [6], neighborhood-based explainability introduces some additional exposure bias. This additional exposure bias is what causes this loss function to be partially unbiased.
- **Unbiased Explainable Bayesian Personalized Ranking (UEBPR):** This is our proposed unbiased and explainable BPR loss function [6]. This loss function promotes ranking relevant and explainable items on the top of the recommendation list for a user and is, at the same time, theoretically free of exposure bias. We used a similar IPS-based approach to also eliminate the aforementioned additional exposure bias coming from the explainability as explained in [6].

2.2. Models

In our proposed framework, we implement the aforementioned loss functions with a Matrix Factorization (MF) [7] model. The MF model consists of two embedding matrices $P \epsilon \mathbb{R}^{n \times K}$ and $Q \epsilon \mathbb{R}^{m \times K}$ for the users and items respectively, each with K latent factors. In this case, n is the number of users and m is the number of items. Each row of the user (item) embedding matrix corresponds to a latent representation, or latent vector, of the user (item). Hence, the preference $\hat{y}_{u,i}$ of a user u to an item i is determined through a dot product of the user and item latent vectors, such that:

$$\hat{y}_{u,i} = P_u \cdot Q_i^T \quad (1)$$

This predicted preference is then fed into the corresponding loss function, from the previous subsection, to train the model to learn the "right" ranking of items for the users which fits the training data and optimizes the intended validation metrics [6].

Although the initial goal of our framework is to assess the effectiveness of the proposed explainability and debiasing components that were introduced into the BPR loss, which is why we only used MF, we also aim to ensure flexibility in choosing the machine learning model to train. For this reason, we made it easy to introduce a new model into the framework and train it using our proposed loss functions. In fact, the MF model's class is defined in the "Code/EBPR_model.py" file. Hence, replacing this class with the class of any other pairwise ranking model should be straight-forward and should allow for testing all of our framework's functionalities with the new model.

2.3. Training the models

The first functionality that our proposed framework offers is training the MF model with a specified loss function and evaluating it in terms of ranking accuracy, explainability, and popularity debiasing. The implemented evaluation metrics are summarized in Table 1. The "README.md" file in the repository explains how the training can be initiated and summarizes all of the hyperparameters that can be tuned as arguments in the command. It is worth noting that our framework comes ready to train on four benchmark datasets, being the "Movielens 100K", "Movielens 1M", "Yahoo! R3", and "Last.FM 2K" datasets. Also, switching between loss functions is as easy as updating the value of the "model" argument. To evaluate our models, we rely on the Leave One Out (LOO) evaluation process [12] where the last interaction of every user is left out for testing and the second to last interaction is left out for validation. The ranking accuracy metrics compare those test and validation instances to 100 randomly sampled negatives for every user. Finally, when training a model using our proposed framework, the model is trained on the training set, and evaluated on the test set in every epoch. The test performance on the best epoch is finally output.

Note that, as was mentioned in [6], the implemented loss functions differ from the proposed estimators in [6] in the following three aspects: First, as we do not have the true exposure propensities in our datasets, we estimate them with the relative item popularities. Second, as we cannot practically train on all possible (user, positive item, negative item) tuples and consider all non-interacted items as negatives, we use negative sampling and sample one negative interaction per positive interaction in the training, following the same methodology that was used in [1]. Finally, to ensure a fair comparison between all the models (unbiased and not unbiased) and truly assess the impact of every component in the loss, we train all the models on exactly the same training tuples.

2.4. Tuning the models

We provide the possibility to tune a given model on the validation set through a single command using random search, as explained in the "README.md" file. In this case, a set of hyperparameter configurations is sampled from a pool of hyperparameter values that are specified by the user. Then the model is trained with all of these hyperparameter configurations for a specified number of replicates. For every run, the best result on the validation set is saved. Finally, the results of all the models are aggregated in a table and saved as a Comma-Separated Values (CSV) file.

3. Impact

We proposed a fairness in recommendation framework that allows for training, tuning, and evaluating machine learning models for pairwise ranking recommendation with various novel loss functions that

Table 1
Evaluation metrics implemented.

Evaluation criterion	Metric	Description
Ranking accuracy	NDCG@k	Normalized Discounted Cumulative Gain at cutoff k. This metric assesses the ranking quality of the top k recommendations for every user with a higher emphasis on items on the top of the recommendation list.
	HR@k	Hit Ratio at cutoff k. This metric assesses the proportion of hits in the top k recommendations of every user. A hit corresponds to a relevant item that appeared in the top k recommendations.
Explainability	MEP@k	Mean Explainability Precision at cutoff k. This evaluation metric, proposed in [13], measures the proportion of explainable items that were recommended in the top k recommendation list of every user. An item i is considered explainable to a user u if the explainability value $E_{u,i}$ of the item to the user is strictly higher than 0, as explained in [6].
	WMEP@k	Weighted Mean Explainability Precision at cutoff k. This evaluation metric, proposed in [6], provides a smoother evaluation of the explainability of the top k recommended items, by weighting the items' contributions with their explainability values $E_{u,i}$, as explained in [6].
Popularity debiasing	EFD@k	Expected Free Discovery at cutoff k. This evaluation metric, proposed in [14], evaluates the model in terms of novelty, which is a measure of the ability of a system to recommend relevant long-tail items.
	Avg_Pop@k	Average Popularity at cutoff k. This evaluation metric evaluates the top k recommendations of every user in terms of the average popularity of the recommended items. The lower the average popularity, the better the popularity debiasing capabilities of the model.
	Div@k	Diversity at cutoff k. This evaluation metric computes the average pairwise similarity between the items in the top k recommendation list [14]. The lower the average pairwise similarity between the recommended items, the higher the diversity of the recommendation list.

have explainability and exposure debiasing capabilities. Our framework's impact on research in recommender systems can be summarized below:

- Our framework implements several state-of-the-art machine learning models for pairwise ranking-based recommendation [1,6,8].
- Our framework allows for training and tuning machine learning models for pairwise ranking-based recommendation using a single command. The hyperparameters of the models can be specified as arguments to the command which is convenient.
- Our framework is ready to use with the Matrix Factorization (MF) model and with five state-of-the-art loss functions for pairwise ranking from implicit feedback that allow for explainability and debiasing.
- Although our framework implements MF as a base model, it is fairly easy and straight-forward to implement any other pairwise ranking model and use it within our framework.
- Our framework is ready to use with four benchmark datasets that are commonly used in recent research papers.
- Our framework allows for evaluating recommender systems empirically in the three aspects of ranking accuracy, explainability, and popularity debiasing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by National Science Foundation grants IIS-1549981, DRL-2026584, and CNS-1828521.

References

[1] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, Lars Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, 2012, arXiv preprint arXiv:1205.2618.
[2] Ruining He, Julian McAuley, VBPR: visual bayesian personalized ranking from implicit feedback, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, No. 1, 2016.
[3] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, Xiangnan He, Bias and debias in recommender system: A survey and future directions, 2020, arXiv preprint arXiv:2010.03240.
[4] Mustafa Bilgic, Raymond J. Mooney, Explaining recommendations: Satisfaction vs. promotion, in: Beyond Personalization Workshop, IUI, Vol. 5, 2005, p. 153.
[5] Behnoush Abdollahi, Olfa Nasraoui, Using explainability for constrained matrix factorization, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, 2017, pp. 79–83.
[6] Khalil Damak, Sami Khenissi, Olfa Nasraoui, Debiased explainable pairwise ranking from implicit feedback, in: Fifteenth ACM Conference on Recommender Systems, in: RecSys '21, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450384582, 2021, pp. 321–331, http://dx.doi.org/10.1145/3460231.3474274.
[7] Yehuda Koren, Robert Bell, Chris Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37, http://dx.doi.org/10.1109/MC.2009.263.
[8] Yuta Saito, Unbiased pairwise learning from implicit feedback, in: NeurIPS 2019 Workshop on Causal Machine Learning, 2019.
[9] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, Thorsten Joachims, Recommendations as treatments: Debiasing learning and evaluation, 2016, arXiv preprint arXiv:1602.05352.
[10] Ludovik Coba, Panagiotis Symeonidis, Markus Zanker, Personalised novel and explainable matrix factorisation, Data Knowl. Eng. 122 (2019) 142–158.
[11] Shuo Wang, Hui Tian, Xuzhen Zhu, 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.
[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, Tat-Seng Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.
[13] Behnoush Abdollahi, Olfa Nasraoui, Explainable matrix factorization for collaborative filtering, in: Proceedings of the 25th International Conference Companion on World Wide Web, 2016, pp. 5–6.
[14] Saúl Vargas, 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.