Bias Mitigation in Recommender Systems to Improve Diversity

Du Cheng, Doruk Kilitçioğlu and Serdar Kadıoğlu

Abstract

Evaluating and mitigating bias in recommendation systems is of great practical interest in many real-world applications. This motivates the community to call for a more rounded evaluation of recommendation solutions that not only measures performance based on standard success metrics such as hit rate and ranking but also the quality across different user groups. To this end, we propose integrating post-processing techniques to mitigate bias in recommendations and measure the effectiveness of our approach in the CIKM 2022 EvalRS Challenge.

Keywords

Recommender Systems, CIKM 2022 EvalRS Challenge, Algorithmic Fairness, Equalized Odds

1. Introduction

Recommendation systems are ubiquitous in several applications, and their success is often measured by point-wise engagement metrics. Unfortunately, this might not only hinder important information when evaluating model performance but might also suffer from unwanted bias across different user and item groups.

In response to the CIKM 2022 EvalRS Challenge [1], we propose[1] an approach that adds both activity-based averaging, and post-processing steps to a collaborative filtering baseline model for bias mitigation. Specifically, we use equalized odds calibration [2] to perturb decisions of the recommender conditioned on protected classes to enhance fairness.

We tested this approach using the public Last.fm dataset by applying bias mitigation to improve diversity metrics. We then measure the difference between the performance over the entire test set versus the "protected" classes, such as song popularity and short user history. In the following, we provide further details on the approach taken for the EvalRS Challenge.

2. Our Methodology

Our work revolves around a standard approach as a baseline commonly known as Collaborative Filtering [3]. While sophisticated techniques, e.g., based on recent advances in Deep Neural Networks [4, 5] and Transformer models [6, 7] exist, our choice of a classical method is motivated by the desire to quantify the attribution of our post-processing approach for bias mitigation. We choose to build the model based on implicit feedback from the interaction data. We focus on implicit feedback since we do not have access to direct input from users, and item features are limited.

2.1. Alternating Least Squares (ALS)

The Alternating Least Squares [8] is a classical method that treats interaction data as indication of user preferences and takes into account the associated confidence levels. It computes the user factor $x_u \in \mathbb{R}^f$ and a vector $y_i \in \mathbb{R}^f$ for each item such that the user preference p_{ui} can be expressed as $x_u^T y_i$. Consequently, the following cost function is minimized:

$$\min_{x,y\star} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2) \tag{1}$$

where the second term serves to regularize the model, and r_{ui} denotes the confidence level of user u's preference towards item i. Here r_{ui} can be computed as $1 + \alpha r_{ui}$ where α is the weight given to positive feedback, and r_{ui} is the raw binarized interaction.

2.2. Beyond ALS

We propose two directions to extend the ALS model for better performance. The first direction is to train the model on the entire dataset as well as categorized subsets. In the Last.fm dataset, the users are categorized into three groups according to their user activity level. As such, an alternating least squares model specific to the user activity level is trained on each group. We take the top n_sum items from each model and recommend the $top{-}k$ items that score the highest in the average preference score from both the overall model and the categorized model in the final results.

du.cheng@fmr.com (D. Cheng); doruk.kilitcioglu@fmr.com (D. Kilitçioğlu); serdar.kadioglu@fmr.com (S. Kadıoğlu)

CEUR Workshop Proceedings (CEUR-WS.org)

[1] https://github.com/fidelity/jurity/tree/master/evalrs

The second direction is post-processing. The important insight is to view features such as gender, country, user activity, artist, and track popularity as criteria to create the "protected" groups. The problem can then be formulated as adjusting the model such that discrimination is mitigated from the recommendation results. One approach to achieve this is by optimizing Equalized Odds [9]. The equalized odds is defined as the independence of the predictor $\hat{Y}$ and protected membership A conditional on the prediction results. More formally;

$$Pr(\hat{Y} = 1|A = 0, Y = y) = Pr(\hat{Y} = 1|A = 1, Y = y) \quad (2)$$

given $y \in (0, 1)$.

Conceptually, the equalized odds method works as follows. First, we find the convex hull of the ROC curves of the contrasted groups such that any false-positive rate (FPR), true-positive rate (TPR) pair can be satisfied by either protected-group-conditional predictor. During training, we obtain four probabilities of flipping the likelihood of a positive prediction. Then during prediction, we randomize the indices of predictions and apply these learned mixing rates on the new data. The open-source Jurity library [10] offers an implementation of this method and helps us achieve our goal [2].

2.3. Beyond Binary Fairness Metrics

Notice, however, that there is still a gap between generating recommendations, which can be seen as multi-class, multi-label prediction, and binary mitigation techniques, such as equalized odds. To bridge this gap, we propose the following approach:

- Obtain the user-item-score matrix for user-item pairs and run softmax on it.
- Calculate a binary cutoff point per item based on the 80% quantile scores, as suggested in [**?**].
- Binarize the results item-wise and run equalized odds, using user activity as a protected class.
- In case the application of equalized odds change the binary label, go back to the softmax scores and use its complement, i.e., (1 - softmax).
- Re-order recommendations using the new scores.

The result of this process leads to a normalized set of scores where for each item, the decision of whether that item is recommended to a user is now unbiased between high activity and low activity users. As such, we would expect to see a lower difference in metrics when comparing high activity and low activity users, which is exactly what MRED_USER_ACTIVITY measures. This technique can be applied to user activity and item popularity.

[2]https://github.com/fidelity/jurity

alpha	Regularization	Factors	Score	Hit Rate
0.1	0.1	50	-21.82	0.046
1	0.05	50	-27.19	0.062
10	0.1	50	-27.41	0.075
20	0.2	50	-26.75	0.076
40	0.1	50	-23.54	0.075

Table 1
Hyperparameter tuning of the ALS algorithm. We list 5 out of 100 configurations for brevity.

In practice, binarization across all users (items) and equalizing odds per user (item) is costly. We need one mitigation model per user to mitigate differences in track popularity. Analogously, we need one mitigation model per item when balancing user activity. To simplify the process, we choose the $top{-}m$ items with the highest engagement the mitigation is focused on user activity.

3. Experiments

3.1. The Challenge Data

Our initial analysis is based on the transformed version of LFM-1b dataset. This dataset contains 119,555 distinct users, 820,998 tracks and 37,926,429 listening events. Our primary data source remains the implicit user feedback from the interaction data. We categorize and evaluate recommendation results across different groups. Based on MRED_USER_ACTIVITY, we separate users into the $[1, 100, 1000]$ user activity groups.

3.2. Additional Testing

We extend the testing suite provided by RecList [11] with a custom test aimed at fairness in recommender systems. More specifically, we look at the intersection between the user activity and track popularity, evaluating whether there is a material difference in the popularity of tracks that is recommended to users with differing activity.

To evaluate our new metric, inspired by MRED_USER_ACTIVITY, we first binarize users into high- and low-activity groups (using 1000 listens as a cut-off). We then bin tracks into [1, 10, 100, 1000] groups, similar to MRED_TRACK_POPULARITY. For each user, we look at which track popularity groups they are recommended and the activity group they belong to. We then utilize the multi-class statistical parity measure from Jurity [10] to measure fairness.

3.3. Numerical Results

We focus on optimizing the model performance in two aspects, i) the standard performance metrics and ii) the diversity metrics. The metrics are calculated using the RecList [11] library provided by the organizers.

Table 1 presents the set of hyperparameters considered when training the model on the whole dataset to find the configuration with the highest performance score.

Model	Score	Hit Rate	MRED_USER_ACTIVITY	Runtime
CBOW Baseline	-1.212	0.036	-0.022	N/A
ALS	-21.823	0.046	-0.007	3 min per fold
ALS + Averaging (with n_sum = 500)	-11.31	0.027	-0.0086	19 min per fold
ALS + Averaging (with n_sum = 1000)	-6.670	0.017	-0.005	25 min per fold
ALS + Post-processing	-18.761	0.042	-0.006	4 min per fold

Table 2
Comparison of post-processing with tuned ALS model. n_sum specifies the number of top items from each model that is used in the calculation. ALS + Averaging with n_sum = 500 is used in the final submitted code.

Table 2 summarizes our attempts that involve training and evaluating the averaged models and the post-processing algorithm for bias mitigation in an effort to balance between performance and diversity metrics.

4. Discussion

4.1. The Impact of Averaging

We compare our work with both the CBOW baseline provided by the challenge organizers and other solutions. When compared to the CBOW baseline, our overall score is lower while our hit-rate is lower but similar. Remember that our approach for averaging specifically targets the MRED_USER_ACTIVITY metric, hence, as expected, our score in this metric is better than the baseline.

One immediate observation from the challenge results presented in Leaderboard - II [3] is, due to the aggregated scoring scheme, it is non-trivial to compare algorithms. Different methods exhibit different strengths. Notice that scores in Leaderboard -II are calculated based on the previous statistics achieved in Leaderboard - I.

In terms of traditional performance metrics, it is worth noting that, our hit-rate performance is within the top-5 solutions. This is interesting given that we only utilized a standard recommendation algorithm. Without post-processing, our hit-rate would be even higher.

Inline with the *rounded evaluation* objective of the competition, the top scoring solution does not strike a high hit-rate either. This is also the case for the two-tower deep neural networks, as evident in Leaderboard - I, where its performance trails behind the classical CBOW baseline.

In terms of extended metrics, our results on MRED_USER_ACTIVITY metric considerable improves over the CBOW baseline and is the 7th over 14, discarding solutions with -100 performance scores. Our relatively good performance on MRED_USER_ACTIVITY, even when our overal scores is not the best, is an empirical evidence for efficacy of the ensemble modeling approach. We observe that when targeting one specific metric, the overall results have suffered.

An area for future improvement is to focus on how to utilize averaging in such a way that benefits beyond a single protected class.

4.2. The Impact of Post-Processing

We compare our post-processing results with both the baseline CBOW model and our baseline ALS model. Compared to the CBOW baseline model, our overall score is lower while our hit-rate is higher. Our MRED_USER_ACTIVITY is again higher than this baseline, since our post-processing specifically targets this metric. Compared to our baseline ALS model, our overall score increases due to an increase in MRED_USER_ACTIVITY, and our hit-rate worsens. Further compared to our averaging model, the post-processing sacrifices less hit rate at the expense of achieving less improvement in MRED_USER_ACTIVITY.

Since the post-processing involves fitting an equalized odds model per item, we quickly hit high runtimes when using more than 1000 items. By utilizing the most popular items, we increase the impact we get from each trained equalized odds model. However, our results show that even though the post-processing accomplishes an improvement directionally, our solution based on averaging performs slightly better. As such, our final submission for the CIKM 2022 EvalRS Challenge is the averaging model with $n_sum = 500$.

5. Conclusion

In this work, we augmented the well-known collaborative filtering algorithm with ensembles and bias mitigation to strike a balance between performance and diversity. This carefully crafted CIKM Challenge goes beyond standard metrics, provides the easy-to-use RecList library, and raises awareness for a rounded evaluation. In the same spirit, we focused on mitigating bias on diversity metrics, leveraged the Jurity library, and demonstrated encouraging results. We showed how to use existing

[3] https://reclist.io/cikm2022-cup/leaderboard.html

algorithmic fairness metrics for recommendations and extended equalized odds beyond binary classification.

References

[1] J. Tagliabue, F. Bianchi, T. Schnabel, G. Attanasio, C. Greco, G. d. S. P. Moreira, P. J. Chia, Evalrs: a rounded evaluation of recommender systems, 2022. URL: https://arxiv.org/abs/2207.05772. doi:10.48550/ARXIV.2207.05772.

[2] M. Hardt, E. Price, N. Srebro, Equality of opportunity in supervised learning, 2016. URL: https://arxiv.org/abs/1610.02413. doi:10.48550/ARXIV.1610.02413.

[3] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, WWW '01, Association for Computing Machinery, New York, NY, USA, 2001, p. 285–295. URL: https://doi.org/10.1145/371920.372071. doi:10.1145/371920.372071.

[4] M. Naumov, D. Mudigere, Dlrm: An advanced, open source deep learning recommendation model, 2020.

[5] X. Yi, J. Yang, L. Hong, D. Z. Cheng, L. Heldt, A. Kumthekar, Z. Zhao, L. Wei, E. Chi, Sampling-bias-corrected neural modeling for large corpus item recommendations, in: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 269–277. URL: https://doi.org/10.1145/3298689.3346996. doi:10.1145/3298689.3346996.

[6] G. de Souza Pereira Moreira, S. Rabhi, J. M. Lee, R. Ak, E. Oldridge, Transformers4rec: Bridging the gap between nlp and sequential / session-based recommendation, in: Proceedings of the 15th ACM Conference on Recommender Systems, RecSys '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 143–153. URL: https://doi.org/10.1145/3460231.3474255. doi:10.1145/3460231.3474255.

[7] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang, Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, 2019, p. 1441–1450.

[8] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 263–272. doi:10.1109/ICDM.2008.22.

[9] M. Hardt, E. Price, E. Price, N. Srebro, Equality of opportunity in supervised learning, in: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 29, Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf.

[10] F. Michalský, S. Kadıoğlu, Surrogate ground truth generation to enhance binary fairness evaluation in uplift modeling, in: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), 2021, pp. 1654–1659. doi:10.1109/ICMLA52953.2021.00264.

[11] P. J. Chia, J. Tagliabue, F. Bianchi, C. He, B. Ko, Beyond ndcg: Behavioral testing of recommender systems with reclist, WWW '22 Companion, Association for Computing Machinery, New York, NY, USA, 2022, p. 99–104. URL: https://doi.org/10.1145/3487553.3524215. doi:10.1145/3487553.3524215.