Bias disparity in collaborative recommendation

Bias Disparity in Collaborative Recommendation: Algorithmic Evaluation and Comparison*

Masoud Mansoury†
Eindhoven University of Technology
Eindhoven, the Netherlands
m.mansoury@tue.nl

Bamshad Mobasher
DePaul University
Chicago, USA
mobasher@cs.depaul.edu

Robin Burke
University of Colorado Boulder
Boulder, USA
robin.burke@colorado.edu

Mykola Pechenizkiy
Eindhoven University of Technology
Eindhoven, the Netherlands
m.pechenizkiy@tue.nl

ABSTRACT

Research on fairness in machine learning has been recently extended to recommender systems. One of the factors that may impact fairness is bias disparity, the degree to which a group's preferences on various item categories fail to be reflected in the recommendations they receive. In some cases biases in the original data may be amplified or reversed by the underlying recommendation algorithm. In this paper, we explore how different recommendation algorithms reflect the tradeoff between ranking quality and bias disparity. Our experiments include neighborhood-based, model-based, and trust-aware recommendation algorithms.

KEYWORDS

Recommender systems, Trust ratings, Fairness, Bias disparity

1 INTRODUCTION

Recommender systems are powerful tools in extracting users preferences and suggesting desired items. These systems, while accurate, may suffer from a lack of fairness to specific groups of users. Research in fairness-aware recommender systems have shown that the outputs of recommendation algorithms are, in some cases, biased against protected groups [7]. As a result, this discrimination among users will degrade users' satisfaction, loyalty, and effectiveness of recommender systems, and at worst, it can lead to or perpetuate undesirable social dynamics.

Discrimination in recommendation output can originate from different sources. It may stem from the underlying biases in the input data [4, 25] used for training. On the other hand, the discriminative behavior may be the result of recommendation algorithms [13, 27, 28].

In this paper, we examine the effectiveness of recommendation algorithms in capturing different groups' interests across item categories. We compare different recommendation algorithms in terms of how they capture the categorical preferences of users and reflect them in the recommendation delivered.

It is important to note that in this paper, although we do not directly measure the fairness of recommendation algorithms, we study bias disparity of recommendation algorithms as an important factor that affects fairness. The benefit of studying bias disparity in recommender systems is that, depending on the domain, knowing which algorithms produce more or less disparity from users' stated preferences can allow system designers to better control the recommendation output. In our analysis of bias disparity, we also take into account item coverage in recommended lists. A recommendation algorithm with higher item coverage signifies that majority of item providers in the system will have equal chance to be shown to users.

Our analysis includes a variety of recommendation algorithms: neighborhood models, factorization models, and trust-aware recommendation algorithms. In particular we investigate the performance of trust-aware recommendation algorithms. In these algorithms, besides items ratings, explicit trust ratings are used as side information to enhance the quality of input values for recommender systems. It has been shown that using explicit trust ratings will provide advantages for recommender systems [20]. First, since trust ratings can be propagated, they can help overcome cold-start issue in recommender systems. Secondly, trust-aware methods are robust against shilling attacks in recommender systems [16]. In this paper, we also analyze the performance of these algorithms in addressing bias disparity in recommender systems.

The motivation behind this research is analyzing the performance of recommendation algorithms in preference deviation across item categories for a specific group of users (e.g., male vs. female). Given protected and unprotected groups, we aim to compare the ability of recommendation algorithms to generate recommendations equally well for each group based on their preferences in training data. Therefore, no matter what the context of the dataset is, given protected/unprotected groups and item categories, we are interested in comparing recommendation algorithms for their ability to recommend preferred item categories to these groups of users.

For experiments, we prepared a sample of publicly-available Yelp dataset for research on fairness-aware recommender systems. Our experiments are performed on multiple recommendation algorithms and the results are evaluated in terms of *bias disparity* and *average disparity* along with ranking quality and item coverage.

Presented at the RMSE workshop held in conjunction with the 13th ACM Conference on Recommender Systems (RecSys), 2019, in Copenhagen, Denmark.
†This author also has affiliation in School of Computing, DePaul University, Chicago, USA, mmansou4@depaul.edu.

2 BACKGROUND

The problem of unfair outputs in machine learning applications is well studied [3, 6, 12] and also it has been extended to recommender systems. Various studies have considered fairness in recommendation results [4].

One research direction in fairness-aware recommender systems is providing fair recommendations for consumers. Burke et. al. in [4] have shown that inclusion of a balanced neighborhood regularization to SLIM algorithm can improve the equity of recommendations for protected and unprotected groups. Based on their definition for protected and unprotected groups, their solution takes into account the group fairness of recommendation outputs. Analogously, Yao and Huang in [27] improved the equity of recommendation results by adding fairness terms to objective function in model-based recommendation algorithms. They proposed four fairness metrics that capture the degree of unfairness in recommendation outputs and added these metrics to learning objective function to further optimize it for fair results.

Zhu et al. in [29] proposed a fairness-aware tensor-based recommender systems to improve the equity of recommendations while maintaining the recommendation quality. The idea in their paper is isolating sensitive information from latent factor matrices of the tensor model and then using this information to generate fairness-aware recommendations.

Besides consumer fairness, provider fairness is another research direction in fairness-aware recommender systems. Provider fairness refers to the fact that items belong to each provider have equal chance to be shown in the recommended lists. This is known as *popularity bias* and usually measured by *item coverage*.

Abdollahpouri et al., [2] addressed popularity bias in learning-to-rank algorithms by inclusion of fairness-aware regularization term into objective function. They showed that the fairness-aware regularization term controls the recommendations being toward popular items.

Jannach et al., [11] conducted a comprehensive set of analysis on popularity bias of several recommendation algorithms. They analyzed recommended items by different recommendation algorithms in terms of their average ratings and their popularity. While it is very dependent to the characteristics of the data sets, they found that some algorithms (e.g., `SlopeOne`, KNN techniques, and ALS-variant of factorization models) focus mostly on high-rated items which bias them toward a small sets of items (low coverage). Also, they found that some algorithms (e.g., ALS-variants of factorization model) tend to recommend popular items, while some other algorithms (e.g., `UserKNN` and `SlopeOne`) tend to recommend less-popular items.

Multi-stakeholder recommender systems simultaneously take into account the fairness of all stakeholders or entities in a multi-sided platform. The main goal of multi-stakeholder recommendations is maximizing the fairness of all stakeholders. Consumers and providers are the major stakeholders in most multi-sided platforms [1, 5].

Surer et al. in [30] proposed a multi-stakeholder optimization model that works as a post-processing approach for standard recommendation algorithms. In this model, a set of constraints for providers are considered when generating recommendation lists for end users. Also, Liu and Burke in [17] proposed a fairness-aware re-ranking approach that iteratively balances the ranking quality and provider fairness. In this post-processing approach, users' tolerance for diversity list is also considered to find trade-off between accuracy and provider fairness.

3 FAIRNESS METRICS

In this paper, we compare the performance of state-of-the-art recommendation algorithms in terms of bias disparity in recommended lists. We also consider ranking quality and item coverage of recommendation algorithms as two important additional metrics.

We use two metrics to measure changes in bias for groups of users given item categories: *bias disparity* and *average disparity*.

Bias disparity measures how much an individual's recommendation list deviates from his or her original preferences in the training set [25]. Given a group of users, G, and an item category, C, bias disparity is defined as follow:

$$BD(G,C) = \frac{B_R(G,C) - B_T(G,C)}{B_T(G,C)} \tag{1}$$

where B_T (B_R) is the *bias* value of group G on category C in training data (recommendation list). B_T is defined by:

$$B_T(G,C) = \frac{PR_T(G,C)}{P(C)} \tag{2}$$

where $P(C)$ is the fraction of item category C in the dataset defined as $|C|\backslash|m|$. PR_T is the preference ratio of group G on category C calculated as:

$$PR_T(G,C) = \frac{\sum_{u \in G} \sum_{i \in C} T(u,i)}{\sum_{u \in G} \sum_{i \in I} T(u,i)} \tag{3}$$

where T is the binarized user-item matrix. If user u has rated item i, then $T(u,i) = 1$, otherwise $T(u,i) = 0$.

The *bias* value of group G on category C in the recommendation list, B_R, is defined similarly.

On the other hand, *average disparity* measures how much preference disparity between training data and recommendation list for one group of users (e.g., unprotected groups) is different from that for another group of users (e.g., protected group). Inspired by *value unfairness* metric proposed by Yao and Huang [27], we introduce the average disparity as:

$$\overline{disparity} = \frac{1}{|C|} \sum_{i=0}^{|C|} |(N_R(G_U, C_i) - N_T(G_U, C_i)) - (N_R(G_P, C_i) - N_T(G_P, C_i))| \tag{4}$$

where G_U and G_P are unprotected and protected groups, respectively. $N_R(G,C)$ and $N_T(G,C)$ return number of items from category C in recommendation lists and training data, respectively, that are rated by users in group G.

As part of our analysis, we also measure item coverage of recommended lists which is an important consideration in provider-side fairness. Given the whole set of items in the system, I, and whole recommendation lists for all users, R_{all}, item coverage measures what percentage of items in the system appeared in recommendation lists and can be calculated as:

Table 1: Parameter configuration

parameter	values
#neighbors	{10,20,30,40,50,70,100,200}
shrinkage	{10,30,50,100,200}
similarity	{pcc,cos}
user regularization	{0.0001,0.001,0.005,0.01}
item regularization	{0.0001,0.001,0.005,0.01}
bias regularization	{0.0001,0.001,0.005,0.01}
implicit regularization	{0.0001,0.001,0.005,0.01}
learning rate	{0.0001,0.001,0.005,0.01}
#iterations	{10,30,50,100}
#factors	{10,30,50,100,150,200,300}
ℓ_1-norm	{0.005,0.05,0.5,2,5}
ℓ_2-norm	{0.005,0.05,0.5,2,5}

$$coverage = 100.\frac{|\{i, i \in (R_{all} \cap I)\}|}{|I|} \tag{5}$$

4 EXPERIMENTS

4.1 Experimental setup

For comparing the effects of recommendation algorithms on bias and on item coverage, we performed an extensive experiments on state-of-the-art recommendation algorithms. Experiments are performed on model-based, neighborhood-based, and trust-aware recommendation algorithms.

Our experiments on neighborhood-based recommendation algorithms include user-based collaborative filtering (`UserKNN`) [22] and item-based collaborative filtering (`ItemKNN`) [23]. Also, our experiments on model-based recommendation algorithms include biased matrix factorization (`BiasedMF`) [15], combined explicit and implicit model (`SVD++`) [14], list-wise matrix factorization (`ListRankMF`) [24], and the sparse linear method (`SLIM`) [21]. Finally, our experiments on trust-aware recommendation algorithms include trust-aware neighborhood model (`TrustKNN`) [20], trust-based singular value decomposition (`TrustSVD`) [9], social regularization-based method (`SoReg`) [18], trust-based matrix factorization (`TrustMF`) [26], and social matrix factorization (`SocialMF`) [10]. Besides above well-known recommendation algorithms, we also performed experiments on two naive algorithms: random and most popular.

For sensitivity analysis, we performed extensive experiments with different parameter configurations for each algorithm. Table 1 shows the parameter configurations we used for our experiments.

We performed 5-fold cross validation, and in the test condition, generated recommendation lists of size 10 for each user. Then, we evaluated nDCG, item coverage, bias disparity, and average disparity at list size 10. Results were averaged over all users and then over all folds. We used *librec-auto* and LibRec 2.0 for all experiments [8, 19].

4.2 Yelp dataset

For our experiments, we use a subset of Yelp dataset from round 12 of Yelp Challenge[1]. In this sample, each user has rated at least 40 businesses and each business is rated by at least 40 users. Thus, there are 1,355 users who provided 100,409 ratings on 1,272 businesses. The range of ratings is 1 (not preferred) to 5 (preferred). The density of rating matrix is 5.826.

This Yelp dataset also has information about users friendship. Each user has selected a set of other users as her friends. We interpret this relationships as a trust network. When user A selects user B as a friend, it means that user A trusts user B with respect to the corresponding domain or category. In this dataset, 919 users have expressed their trustworthiness to 1,172 users and there are 26,453 trust relationships between users. With regard to the number of users, the density of trust matrix is 2.456.

In order to evaluate the recommendation outputs in terms of bias disparity and average disparity, specific information about users and items is needed. First, we need to define users group based on users demographic information and item category based on item contents. In Yelp dataset, there is no useful information about user to define users' group. To overcome this issue, we prepared the dataset by extracting users' gender from users' name. To do this, we use an existing online tool[2] to extract users' gender. In this tool, for each user name as input, it will return the predicted gender, number of samples used for prediction, and prediction accuracy. Hence, it enables us to increase the reliability of extracted genders by taking outputs with high accuracy and fair amount of samples.

Moreover, information about items' category is provided in the dataset. Each business in Yelp dataset is assigned multiple relevant categories.

Overall, the prepared dataset has four separate sets:

1. The rating data that each user provided to businesses.
2. Explicit trust data that each user has selected trusted (friends) users.
3. Users information that consists of users' gender.
4. Items category that consists of several category for each business.

By using this dataset, we define the set $G =< male, female >$ and set C as categories assigned to each business. The dataset is available at https://github.com/masoudmansoury/yelp_core40.

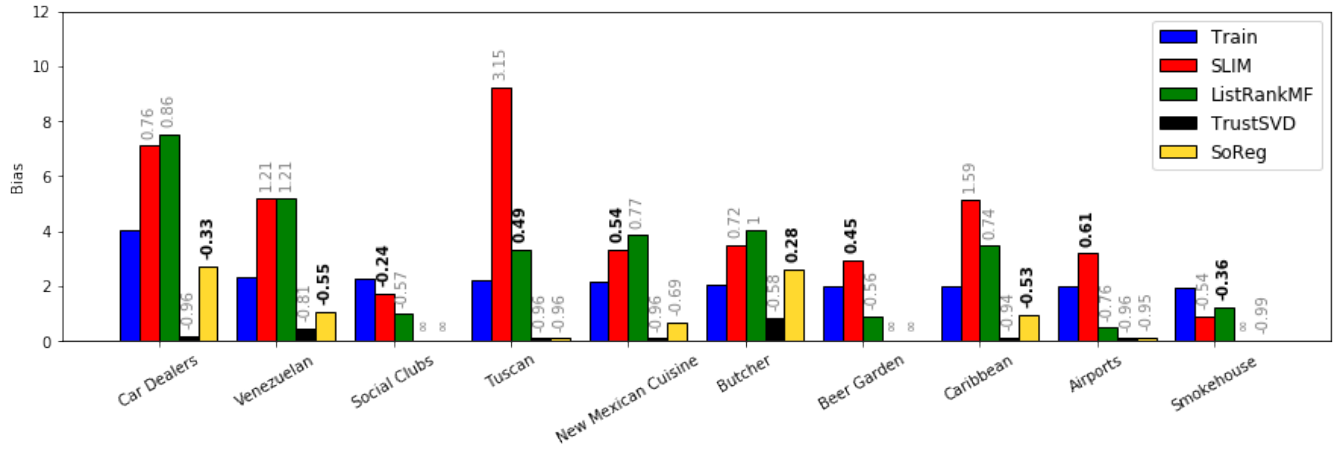
4.3 Experimental results

In this section, we compare the performance of recommendation algorithms across the different metrics discussed earlier. First, we show the bias disparity of recommendations results on top 10 most preferred item categories. Second, we show average disparity for each algorithm on all categories. For sensible comparison, we also take into account the ranking quality and item coverage.

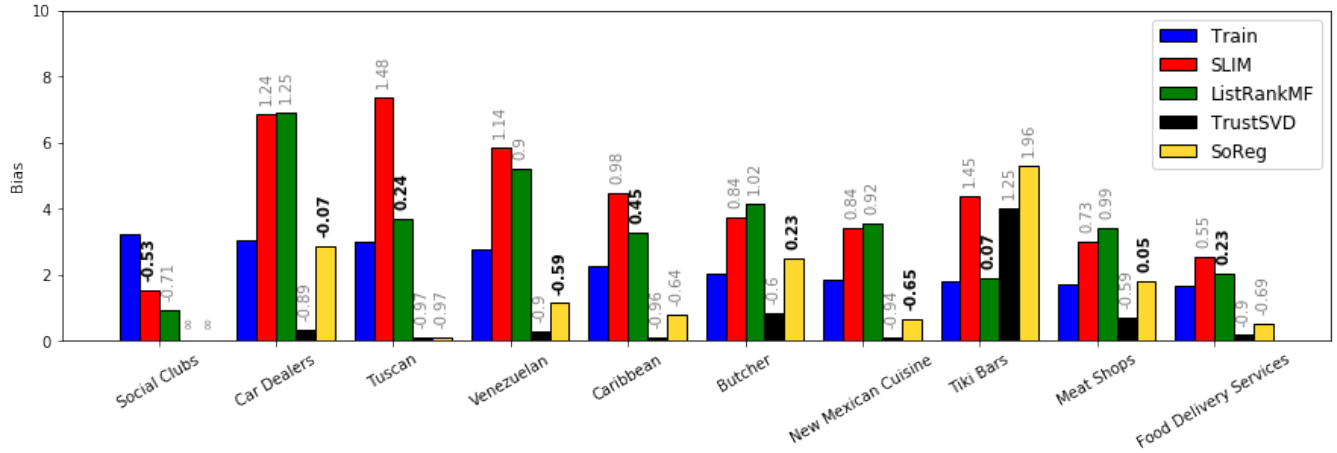
4.3.1 Bias disparity. Results on model-based recommendation algorithms on top 10 most preferred item categories for male and female are shown in Figure 1. Figure 1a shows the bias disparity for male individuals and Figure 1b shows the bias disparity for female individuals. Since there is always a trade-off between accuracy and non-accuracy metrics (e.g., nDCG vs. fairness), for comparison, the fairness analysis is conducted on recommendation outputs that give the same nDCG (highest possible) for all recommendation algorithms. For model-based recommendation algorithms, the nDCG value is set to 0.023 ± 0.001. This setting guarantees that the fairness

[1] https://www.yelp.com/dataset

[2] https://gender-api.com



(a) Male



(b) Female

Figure 1: Bias disparity for model-based recommendation algorithms. The x-axis is the top 10 most preferred categories for male and female on training data and y-axis is bias value computed by equation 2. The numbers on each bar shows the bias disparity computed by equation 1. Numbers in bold show the lowest bias disparity for each category.

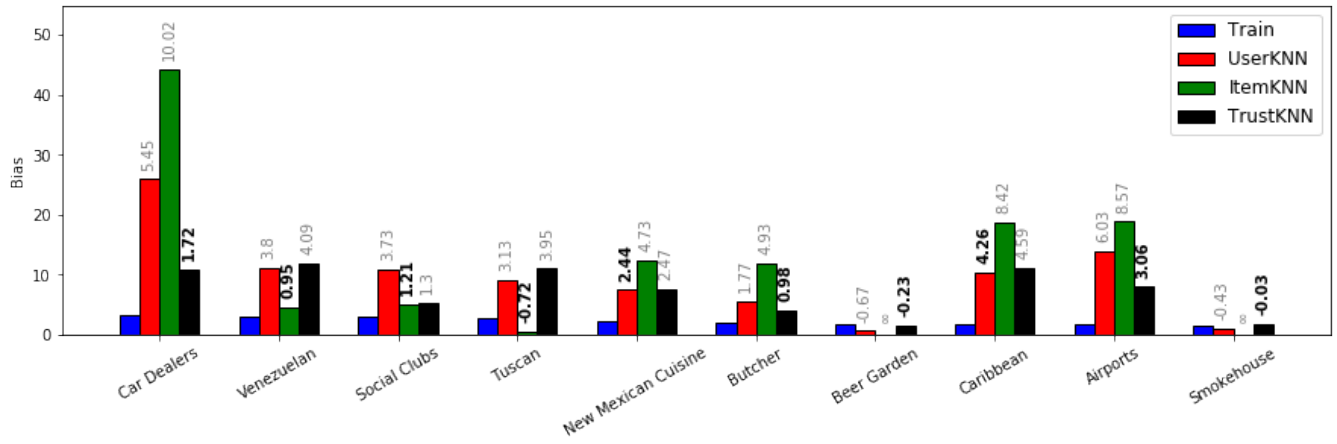
of recommendation algorithms is compared in same condition for all algorithms.

As it is shown in Figure 1, in most cases, `SoReg` provides lower bias disparity on top 10 most preferred categories for male and female groups. For males in Figure 1a, `SoReg` and `SLIM` generated more stable outputs compared to other algorithms with the lowest bias disparity in 40% cases. On the other hand, for female, `SoReg` and `ListRankMF` generated recommendations with the lowest bias disparity of 50% and 40% cases, respectively, when compared to other recommendation algorithms.

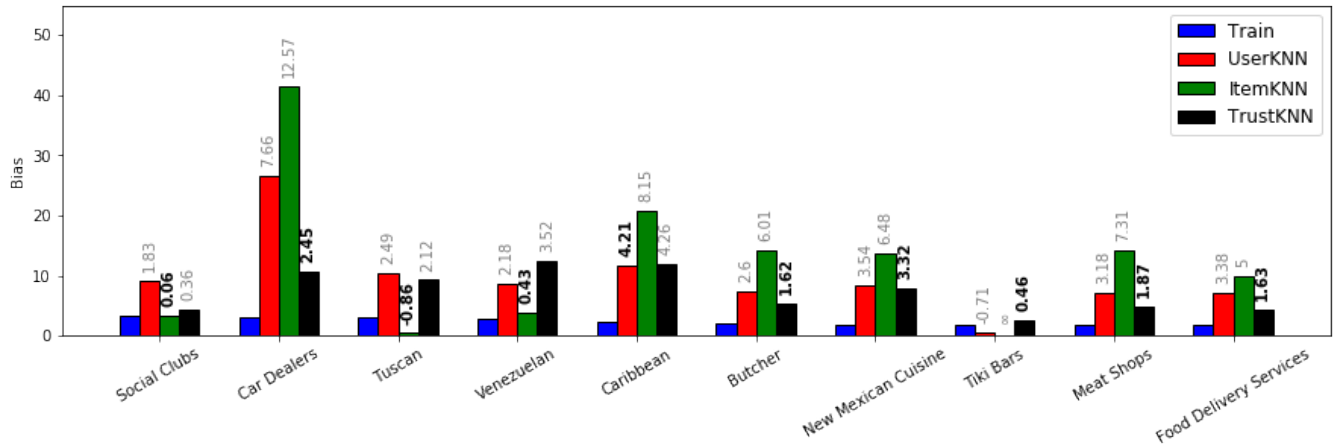
In Figure 1, we did not report the results for `BiasedMF`, `SVD++`, `SocialMF`, `TrustMF`, and random and most popular item recommendations because these algorithms either did not recommend any items from top 10 most preferred categories, or their ranking quality was lower than specified value for other algorithms.

Results on neighborhood-based recommendation algorithms for male and female groups are shown in Figure 2. The nDCG values for neighborhood algorithms are all set to 0.074 ± 0.01. Figure 2a shows the bias disparity of neighborhood models for male. `TrustKNN` generated more stable recommendations compared to other algorithms with 50% top 10 most categories. Also, for other categories, its output is very close to the best one. Moreover, a better output in terms of bias disparity can be observed in Figure 2b for female. On 60% of top 10 most preferred categories, `TrustKNN` worked better that other neighborhood algorithms.

4.3.2 Average disparity. Figure 3 compares the performance of recommendation algorithms with respect to two criteria: 1) how accurately recommendation algorithms generate stable (i.e. low disparity) recommendations for unprotected and protected groups, 2) how accurately recommendation algorithms are able to equally



(a) Male



(b) Female

Figure 2: Bias disparity for memory-based recommendation algorithms. The x-axis is the top 10 most preferred categories for male and female on training data and y-axis is bias value computed by equation 2. The numbers on each bar shows the bias disparity computed by equation 1. Numbers in bold show the lowest bias disparity for each category.

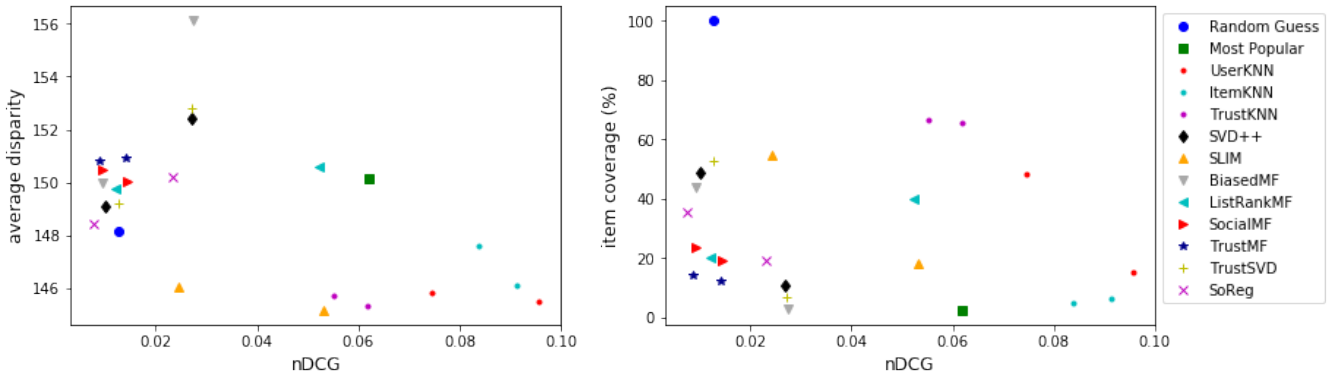
recommend the items belonging to all providers when generating recommendations (provider-side fairness).

For all experiments that we performed with different hyperparameters, the best and worst nDCG for each algorithm are reported in Figure 3.

Random guess algorithm is a naive approach that randomly recommends a list of items to each user. Although this algorithm has low accuracy, it has the highest item coverage and lower average disparity compared to other recommendation algorithms. This algorithm does not take any preferences into account and unlikely to provide good results for any user. Also, most popular item recommendation is another naive, non-personalized, algorithm that only recommends items with the highest number of ratings to each user. Although it has high ranking quality and average disparity similar to model-based recommendation algorithms, it has the lowest item coverage. These algorithms provide baselines that other algorithms should be expected to beat.

For neighborhood models, `TrustKNN` showed better performance. Although it has lower ranking quality than `UserKNN` and `ItemKNN`, it has significantly better item coverage and average disparity. One possible reason for low nDCG of `TrustKNN` can be high sparsity of trust matrix. Using a propagation model for reducing the sparsity of trust matrix may increase the ranking quality of `TrustKNN`. Overall, neighborhood algorithms worked better than model-based algorithms in terms of all metrics. This is due to the fact that the rating data for these experiments is very dense and all users are heavy raters.

For model-based algorithms, `SLIM` shows better performance compared to other algorithms. From Figure 3a, while showing high nDCG, it has the lowest average disparity and in terms of item coverage, it has comparable coverage to other model-based algorithms.



(a) nDCG vs. average disparity **(b) nDCG vs. item coverage**

Figure 3: Comparison of recommendation algorithms by ranking quality and item coverage/average disparity.

This result is also consistent with the definition of `SLIM` algorithm which is an extension of `ItemKNN` and analogous to neighborhood algorithms, it showed significant performance.

In addition, `ListRankMF` is another model-based algorithm that, although having high accuracy and item coverage, has average disparity is as high as other algorithms. Also, for model-based trust-aware recommendation algorithms, although `SoReg` showed significant reduction in bias disparity on the top 10 most preferred categories, it did not improve the average disparity on all categories.

5 CONCLUSION

In this paper, we examined the effectiveness of recommendation algorithms in generating outputs with lower bias disparity for different groups of users across item categories. We measured the performance of recommendation algorithms in terms of bias disparity on top 10 most preferred item categories, average disparity, ranking quality, and item coverage. A comprehensive sets of experiments showed that neighborhood models work significantly better than other algorithms, particularly trust-aware neighborhood model that outperformed other algorithms. Also, we observed that in most cases, having additional information along with rating data can enhance the performance of recommender systems.

For future work, we would like to investigate individual fairness by considering the performance of recommendation algorithms in capturing individual users' interest across different item categories. Also, we are interested to repeat the experiments in this paper on another sample of Yelp dataset with sparser rating data and denser trust data to see how recommendation algorithms are able to control bias disparity.

REFERENCES

[1] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Augusto Pizzato. 2019. Beyond Personalization: Research Directions in Multistakeholder Recommendation. *CoRR* abs/1905.01986 (2019). arXiv:1905.01986 http://arxiv.org/abs/1905.01986

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning-to-Rank Recommendation. In *RecSys '17 Proceedings of the Eleventh ACM Conference on Recommender Systems*. 42–46.

[3] Engin Bozdag. 2013. Bias in algorithmic filtering and personalization. *Ethics and information technology* 15, 3 (2013), 209–227.

[4] Robin Burke, Nasim Sonboli, Masoud Mansoury, and Aldo OrdoÃśez-Gauger. 2017. Balanced neighborhoods for fairness-aware collaborative recommendation. In *RecSys workshop on Fairness, Accountability and Transparency in Recommender Systems*.

[5] Robin D. Burke, Himan Abdollahpouri, Bamshad Mobasher, and Trinadh Gupta. 2016. Towards Multi-Stakeholder Utility Evaluation of Recommender Systems. In *In UMAP (Extended Proceedings)*.

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

[7] Michael D. Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D. Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. 2018. All The Cool Kids, How Do They Fit In?: Popularity and Demographic Biases in Recommender Evaluation and Effectiveness. In *In Conference on Fairness, Accountability and Transparency*. 172–186.

[8] Guibing Guo, Jie Zhang, Zhu Sun, and Neil Yorke-Smith. 2015. LibRec: A Java Library for Recommender Systems. In *UMAP Workshops*.

[9] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2015. TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[10] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *In Proceedings of the fourth ACM conference on Recommender systems*. 135–142.

[11] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. 2015. What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction* 25, 5 (2015), 427–491.

[12] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. 2010. Discrimination aware decision tree learning. In *In 2010 IEEE International Conference on Data Mining*. 869–874.

[13] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. 2011. Fairness-aware learning through regularization approach. In *In 11th International Conference on Data Mining Workshops*. 643–650.

[14] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.

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

[16] Shyong K. Lam and John Riedl. 2004. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*. ACM, 393–402.

[17] Weiwen Liu and Robin Burke. 2018. Personalizing Fairness-aware Re-ranking. *CoRR* abs/1809.02921 (2018). arXiv:1809.02921 http://arxiv.org/abs/1809.02921

[18] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 287–296.

[19] Masoud Mansoury, Robin Burke, Aldo Ordonez-Gauger, and Xavier Sepulveda. 2018. Automating recommender systems experimentation with librec-auto. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 500–501.

[20] Paolo Massa and Paolo Avesani. 2007. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*. ACM, 17–24.

[21] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 497–506.

[22] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. ACM, 175–186.

[23] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW'01 Proceedings of the 10th international conference on World Wide Web*. 285–295.

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

[25] Virginia Tsintzou, Evaggelia Pitoura, and Panayiotis Tsaparas. 2018. Bias Disparity in Recommendation Systems. *CoRR* abs/1811.01461 (2018). arXiv:1811.01461 http://arxiv.org/abs/1811.01461

[26] Bo Yang, Yu Lei, Jiming Liu, and Wenjie Li. 2017. Social collaborative filtering by trust. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 8 (2017), 1633–1647.

[27] Sirui Yao and Bert Huang. 2017. Beyond parity: Fairness objectives for collaborative filtering. In *In Advances in Neural Information Processing Systems*. 2921–2930.

[28] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *In International Conference on Machine Learning*. 325–333.

[29] Ziwei Zhu, Xia Hu, and James Caverlee. 2018. Fairness-aware tensor-based recommendation. In *In Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1153–1162.

[30] ÃŰzge SÃijrer, Robin Burke, and Edward C. Malthouse. 2018. Multistakeholder recommendation with provider constraints. In *In Proceedings of the 12th ACM Conference on Recommender Systems*. 54–62.