Can separate optimization be equal? Joint vs multi-stage optimization in recommender systems fairness

Shreyas Kadekodi, Amanda Aird, Trevor Ward, Robin Burke
{shreyas.kadekodi,amanda.aird,trevor.ward,robin.burke}@colorado.edu
University of Colorado, Boulder
Boulder, Colorado, USA

ABSTRACT

Fair recommender systems help to prevent the exclusion of marginalized communities that biased systems create. Two approaches have been considered to build recommender systems with fair output. One method is to consider fairness in the algorithm that creates the recommendation. The more common approach, because of its practical advantages, is to rerank the results after optimizing the recommendations for accuracy. When reranking is used, generally it is applied to the output of an existing algorithm that has been tuned to maximize its accuracy. In this paper, we explore whether better fairness / accuracy trade-offs are available through joint optimization of a recommendation algorithm and its reranker. We show that for some applications, the difference between joint optimization and reranking after optimization is negligible, with neither method showing a clear advantage.

KEYWORDS

recommender systems, fairness, bias, optimization

ACM Reference Format:
Shreyas Kadekodi, Amanda Aird, Trevor Ward, Robin Burke. 2022. Can separate optimization be equal? Joint vs multi-stage optimization in recommender systems fairness. In *Proceedings of ACM Conference on Knowledge Discovery in Databases (KDD '22).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Recommendation systems are increasingly ubiquitous with almost no facet of modern society left untouched by the decisions these algorithms make. As these systems become more prevalent, their use has broadened beyond areas of consumer taste to areas of greater impact such as housing, employment, and financial services.

This expansion has increased scrutiny of the social impacts from recommender systems. In addition, the scope of recommender system applications includes areas where non-discrimination is legally mandated, to ensure that people are not denied certain benefits on the basis of protected characteristics.

In order to achieve fairer outcomes in recommender systems, a wide variety of techniques have been proposed. These fall into two main categories: modifying or creating new algorithms that take into account fairness during the recommendation process, or using existing algorithms and *reranking* their results to improve fairness. See [6] for a survey of issues and techniques in recommender systems fairness.

In this paper, we explore this second option, where a *base algorithm* produces results, and a *reranker* re-sorts and filters them to produce the final output that the user sees. General practice in reranking is to treat the two steps as independent, letting the base algorithm produce the best recommendations it can based on accuracy criteria, and then applying the reranker to whatever is produced. There are practical advantages to reranking, as it is independent of the recommendation algorithm used and can be tuned to achieve different fairness / accuracy trade-off (even dynamically as in [15]).

There is another option, however, and that is to treat the base algorithm plus reranker as a single system for the purposes of optimization. This option has not been explored in the literature and it is the one that we investigate here.

2 OPTIMIZING FOR FAIRNESS

Like other machine learning algorithms, recommendation algorithms have a number of hyperparameters that control the learning process which must be set by the experimenter before training can take place. Typical examples of hyperparameters are the learning rate, the weight associated with a regularization term(s), or stopping criteria. In matrix factorization algorithms, the number of latent factors is a key hyperparameter. In neural networks, hyperparameters can include the number of layers, their sizes, and activation functions.

Determining the performance of a recommendation algorithm requires not just training it and evaluating the resulting model, but also selecting appropriate hyperparameters, often through extensive experimentation. Hyperparameters can be chosen through systematic exploration of the entire space (grid search) or in one of several algorithmic methods. In this paper, we concentrate on Bayesian black-box optimization BBO [18]. The basic concept of BBO is to treat the results of experiments as samples conditioned on the hyperparameters and to use Bayesian statistics to locate the parameters that supply the best experimental result with highest probability. For our experiments, we used the Tree of Parzen Estimators (TPE) method implemented in the Optuna library [3] as integrated into the librec-auto recommender systems experimentation platform [14, 17]. [1]

We can think of BBO as a function that applies to a parameterized algorithm $f(D, \theta)$, where D is some training data and θ

[1] https://github.com/that-recsys-lab/librec-auto

KDD '22, August 14–18, 2022, Washington, DC

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

is a vector representing the hyperparameters of the algorithm. A BBO algorithm then is a function $BBO(f, D, e) \rightarrow \theta^*$ that discovers the optimal parameters θ^* for f as applied to D relative to some criterion e.

Let B be a base algorithm with hyperparameters θ_B and C be a reranking algorithm with hyperparameters θ_C. We can perform two separate optimizations $\theta_B = BBO(B, D, e_B)$ and then $\theta_C = BBO(B, B(D, \theta_B), e_C)$. Note that the input of the reranker is the output of the recommendation algorithm as learned from the data.) We assume the existence of different criteria for accuracy e_B (ignoring fairness) and fairness e_C. An alternative is to treat the entire process as a single joint optimization on the composed system. $(\theta_B, \theta_C) = BBO(C(B), D, e_J)$ where e_J is some joint criterion taking both accuracy and fairness.

The key question of this research is to compare these two approaches. What are the differences between the current practice (separate optimization) and the joint optimization method? We might expect the joint optimization to perform better since holistic methods usually do work better than greedy ones, but if so how much better, and is it worth the extra effort of combining the optimizations? Note that with the joint method, we are losing one key advantage of reranking, which is the independence of the reranking process from the base algorithm.

3 ALGORITHMS

We test our hypothesis on a range of different (base) recommendation algorithms and rerankers as discussed below.

3.1 Recommendation algorithms

There are dozens of recommendation algorithms to choose from in performing a study of this nature. Our aim was to select algorithms that covered a range of different underlying recommendation logics and with different fairness characteristics. The first, biased matrix factorization (BMF), is a well-known algorithm that represents the user-ratings matrix as a product of lower-dimensional matrices characterized by the interaction between latent factors [2]. This variation on the factorization technique is characterized by the isolation of separate item and user biases for each item and user, respectively, that are learned independently of the latent factors, yielding a prediction function of the following form:

$$\hat{r}_{ij} = o_i + p_j + \sum_{s=1}^{k} u_{is} * v_{js}$$

where o_i is the bias associated with user i and p_j is the bias associated with item j; u_i are the user latent factors and v_j are the item latent factors.

BMF uses rating prediction as its definition of loss and seeks to derive latent factors (and biases) that minimize the error on predictions across the dataset. An alternative type of loss function is one that is aimed at optimizating the ranking performance: the ability of the system to rank preferred items ahead of less-preferred ones. In this setting, we care less about the numerical prediction values and more about the system's discrimination ability. Bayesian Personalized Reranking (BPR) is a well-known algorithm for optimizing ranking accuracy [11]. It is based on the loss function

$$loss = \sum_{(u,i,j)\in D} ln(\sigma(\hat{y}_{ui} - \hat{y}_{uj})) - \lambda_\theta ||\Theta||^2 \tag{1}$$

where the i's are items liked by the user and j's are not. The last term serves to regularize the expression.

Finally, our third method is one that uses a neighborhood-based approach that is among the oldest in recommender systems. The Sparse Linear Method (SLIM) [9] is a generalization of kNN methods. It treats recommendation as a sparse regression problem over the user-item rating matrix.

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & \tfrac{1}{2}\|A - AW\|_F^2 + \tfrac{\beta}{2}\|W\|_F^2 + \lambda\|W\|_1 \\ \text{subject to} \quad & W \geq 0 \text{ and } \operatorname{diag}(W) = 0 \end{aligned}$$

The last two terms in the objective function represent the elastic-net regularizer, which combines L1 and L2 regularization, where $\|W\|$ represents the L2 norm and $\|W\|_1$ the L1 norm.

3.2 Reranking algorithms

Due to space limitations, we chose only to evaluate two reranking algorithms in our experiments. The first is FAR, introduced in [8]. FAR is an extension of EXplicit Query Aspect Diversification (xQuAD) algorithm from information retrieval [12]. The intent of xQuAD was to ensure that results delivered to the user accounted for all aspects of a query. FAR sought similar diversification but on behalf of protected group items. We use a proportional (rather than binary version of FAR) as described in [1]. FAR could be described a "score-based" approach to reranking in that it boosts the score of protected group items and then re-sorts the recommendation list with updated scores.

Our second reranking algorithm FA*IR [19] is quite different. The FA*IR algorithm divides item into two groups: one group of non-protected items and one group of protected items. Each group is ordered based on the preference computed by the recommender system. The algorithm then performs a search over the choices of possible inclusion of items from each group trying to maintain ranking accuracy while fairly distributing items over the ranking.

4 APPLICATION: FAIRNESS IN MICROLENDING

Kiva.org is an online micro-finance site designed to allow individuals to provide help to those in underserved regions. Kiva aggregates loan requests from field partners around the world who lend small amounts of money to entrepreneurs in their local communities. Loans are funded interest-free by Kiva's members, largely in the United States.

4.1 Data Set

We were able to obtain a proprietary dataset from Kiva.org, which contains all lending transactions for the year 2017. The original dataset contained roughly 1 million transactions involving around 120,000 loans and 200,000 Kiva users. However, as noted in [16], transformations were required to make the set usable for collaborative recommendation.

Unlike many common datasets, which contain items that can be used or viewed an indefinite number of times, a loan disappears from Kiva.org once it is funded and is not available for subsequent lenders to view or support, even if they would have liked to. The maximum profile for any loans in our dataset was 330 lenders, with a typical value around 10. In contrast, a popular movie in a movie dataset might be rated by thousands of users. As a result, borrower relation is highly sparse, and loans have very small profiles, making traditional collaborative filtering approaches ineffective.

To counter this problem, we created a version of the dataset in which individual loan items are replaced by pseudo-items, items representing clusters of items with similar properties. User profiles can then be expressed in terms of these pseudo-items instead of the original items, resulting in a denser dataset. Because this dataset is planned for public release, we also normalized the users' loan contributions around each user's mean loan contribution. Thus, it is not possible to associate a specific loan amount with a particular supported loan. We also applied a 10-core transform to the final dataset to ensure there was sufficient data about each user and each item. The final dataset has 2,673 pseudo-items, 4,005 lenders and 110,371 ratings / lending actions.[2]

4.2 Fairness

Recognizing that fairness is a complex concept, which is likely to be defined in different ways in different contexts, we follow [5], in defining a *fairness concern* as a specific type of fairness being sought, relative to a particular aspect of recommendation outcomes, evaluated in a particular way. In this work we concentrate on promoting (as is generally the case in fairness-aware recommendation) a single fairness concern at a time, although within the Kiva dataset there are multiple fairness concerns that could arise. Earlier research [16] identified country, economy sector, and loan size as dimensions along which fairness in lending may need to be sought. In this work, we report on results for loan size and country. Kiva's internal research has found that loans to larger groups (rather than individuals) are more effective at promoting economic development, and that such loans are less likely to receive lender attention. Therefore, in these experiments, we make use of the loans with the largest dollar value ($5,000 and up) as a protected group and seek to increase their representation in the recommendation results.

Kiva's mission is "global financial inclusion" and therefore ensuring equity in the geographic distribution of capital is another important fairness concern. Although most loans in Kiva's system do eventually get funded (around 85%), the length of time that a loan remains in the system is an important variable. If a loan is funded slowly, it takes longer for the borrower to get their funding and the loan occupies space in the Kiva system. Also, there is the chance that the loan will not be funded at all and lenders have to re-engage with the system in order to choose a different loan to support. For this reason, we calculate Percentage Funding Rate (PFR) for each loan:

$$PFR = \frac{100}{t_f - t_p} \tag{2}$$

where t_f is the time that the loan was funded (or the maximum $t_f + 1$ if it was not funded) and t_p is the time that the loan was posted.

A loan with a high PFR is one that was funded quickly. Since loans typically have only 30 days to receive funding, the lowest possible *PFR* is around 3.33. We identified the 16 countries whose loans have the lowest *PFR* scores and labeled these as the protected group for the purposes of geographic fairness. The aim would be to recommend loans from these countries more often to try to equalize *PFR* values across countries.

Note that we concentrate in this work exclusively on *provider-side fairness* as defined in [4]. Borrowers are considered providers in this system since it is their requests for capital that are being delivered as recommendations to the users of Kiva's system. Consumer-side fairness (concerns directed towards these end users) have not arisen as a key issue in this application. In general, however, reranking can be used to increase consumer-side fairness as well and we intend to explore such applications in future work.

5 METHODOLOGY

Two experiments were conducted: one for each of the protected features that we considered. For each experiment, three cross-validation folds were used and the results averaged. All experiments were run using the open-source librec-auto recommendation platform[3]. For each combination of algorithm, reranker and protected feature, we examined the difference between separate optimization for accuracy and fairness and joint optimization of these characteristics. Each base algorithm produced 50 recommendations for each user and the rerankers produced 10 items as the final output list for evaluation.

5.1 Evaluation metrics

We evaluated recommender system performance for both ranking accuracy and fairness. For accuracy, we used Normalized Discounted Cumulative Gain, which values test items more highly if they appear at higher ranks. Only the top 10 list was evaluated, so values reported are NDCG@10 throughout.

There are many different ways to evaluate group fairness in a recommendation context. See [6] for a survey. We have chosen a very simple provider-side statistical parity (PSP) method looking at item exposure (also called *demographic parity* in [13]. We count the number of protected group items that appear in recommendation lists and subtract from it the count of unprotected group, then normalize by the total number of recommendations produced.

The formula for PSP can be represented as

$$PSP = \frac{p - u}{t}$$

where p is the number of protected items, u is the number of unprotected items, and t represents the total number of items. The numerical value of PSP will be between 1 and -1, where 0 means that protected and unprotected groups are equally represented in the recommendations, 1 represents only protected groups are in the output, and -1 means no protected groups are represented in the output.

[2] By the time of publication, we plan to have a public release of this transformed version of Kiva dataset for research use.

[3] https://github.com/that-recsys-lab/librec-auto

For the purposes of this study, we consider a somewhat-arbitrary 10% loss of NDCG to be an acceptable trade-off for increased fairness; this is obviously a very application-specific consideration. This means that when tuning our reranking algorithm we are seeking the best fairness (via PSP) that can be achieved without sacrificing more than 10% of the unreranked NDCG value. For joint optimization, we constructed a joint optimization function J that is insensitive to accuracy changes up to 10% but always sensitive to fairness.

$$J(a, f) = \max(0, (a - 0.9 * a_0)) + f$$

where a_0 is the accuracy of the baseline algorithm.

5.2 Optimization

As indicated above, we compared two different optimization approaches. For separate optimization, we optimized the algorithms first for accuracy using nDCG. We ran 50 iterations of the optimizer, which prior experiments had shown were sufficient to settle on optimal parameters. Then we performed grid search over the reranking parameter to identify the point of 10% accuracy loss. We used this method instead of a second round of black-box optimization because these rerankers only had a single parameter to tune and grid search was faster.

Our second approach involved the joint optimization of base and reranking algorithms as single system. Each iteration of the optimizer evaluated the whole recommendation pipeline from base algorithm through the reranker and the parameter of reranker was included among the optimization variables. The optimization function J, above, was used to ensure that within 10% loss the optimizer would only consider fairness. Note that it is one of the benefits of black-box optimization that the loss function can be arbitrary.

For BPR, the tuned hyperparameters were user regularization, item regulartization, the number of factors, and the learning rate. For BiasedMF, the tuned parameters were learning rate, user regularization, bias regularization, the number of factors, and the maximum learning rate.The tuned parameters for SLIM were L1 and L2 regularization.

		Separate		Joint	
	Baseline	FA*IR	FAR	FA*IR	FAR
BPR	0.0658	0.0596	0.0595	0.0593	0.0642
BMF	0.0325	0.0341	.0330	0.2920	0.0312
SLIM	0.0167	.0160	0.0161	0.0159	0.0150

Table 1: NDCG Results for Experiment 1: Protected feature = loan size

		Separate		Joint	
	Baseline	FA*IR	FAR	FA*IR	FAR
BPR	-0.828	-0.603	-0.328	-0.340	-0.512
BMF	-.628	0.106	0.107	-0.581	-0.269
SLIM	-0.773	-0.094	-0.094	-0.345	-0.043

Table 2: PSP Results for Experiment 1: Protected feature = loan size

		Separate		Joint	
	Baseline	FA*IR	FAR	FA*IR	FAR
BPR	0.0658	0.0596	0.0642	0.0598	0.0594
BMF	0.0325	0.0341	0.031	0.0292	0.0291
SLIM	0.0167	0.0168	0.0164	0.0159	0.0153

Table 3: NDCG Results for Experiment 2: Protected feature = Country

		Separate		Joint	
	Baseline	FA*IR	FAR	FA*IR	FAR
BPR	-0.965	-0.603	-0.512	-0.610	-0.640
BMF	-0.727	-0.306	-0.302	-0.595	-0.202
SLIM	-0.719	-0.208	-0.141	-0.345	-0.202

Table 4: PSP Results for Experiment 2: Protected feature = Country

6 RESULTS

The results for Experiment 1 using loan size as the protected feature are shown in Tables 1 and 2 and in Figure 1a. As we can see, the two optimization methods are quite comparable. (In some cases, they overlap and are indistinguishable on the plot.) BPR is clearly the dominant algorithm, with all of its variants sitting clearly on the Pareto frontier at the upper right. SLIM does relatively poorly on accuracy, although its results can be reranked for better fairness. BMF has results exceeding SLIM, but beneath BPR.

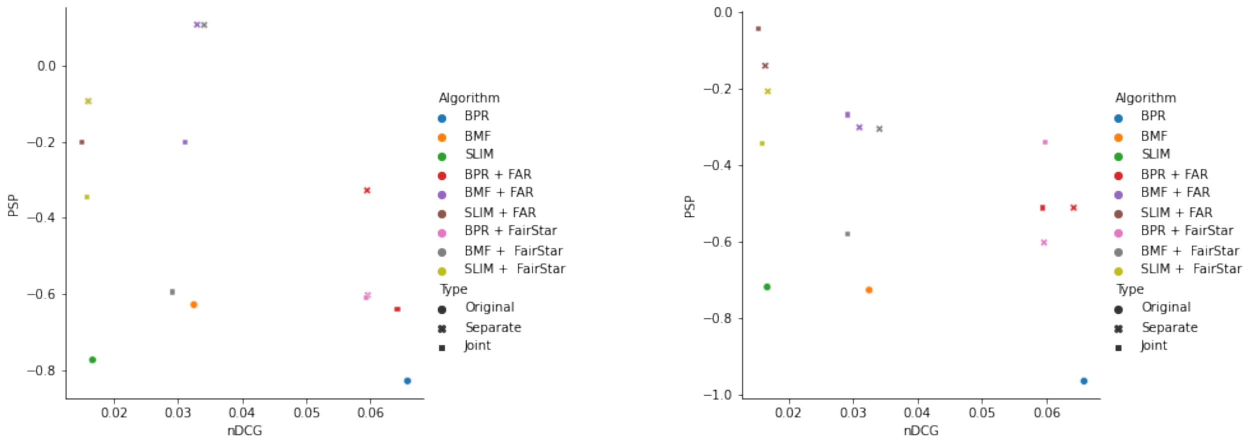
In Experiment 2, with country as the protected feature, BPR retains the best nDCG, but has the lowest fairness, even after reranking. There is no consistent pattern for any algorithm, with joint optimization and two-stage reranking once again performing similarly, without a clear dominant performer.

Table 6 characterizes the overall results of these experiments by averaging the fairness achieved at 10% NDCG loss over the different rerankers. It is not a consistent picture. Seperate optimization is generally better, but there are algorithm / dataset combinations in which it is worse. None of the differences are very large.

It should be noted that joint optimization was a considerably more time intensive process than the both processes independently. While exact computational times were not recorded, it was not uncommon for joint optimization to take as much as 3 or four times as long to execute relative to the reranking after-the-fact approach. Our results suggest that this extra time is probably not worthwhile.

7 RELATED WORK

Fairness in recommender systems has been the subject of considerable research attention over the past five years. A variety of models and reranking methods have been proposed and evaluated as detailed in [6]. This field has also intersected with research in information retrieval and fairness in ranking systems that used in other settings, for example, non-personalized lists of job candidates in algorithmic hiring. See the survey in [20].



(a) Experiment 1 **(b) Experiment 2**

Figure 1: Ranking accuracy vs fairness

	Loan Size		Country	
	Separate	Joint	Separate	Joint
BPR	-0.466	-0.426	-0.558	-0.635
BMF	0.107	-0.425	-0.304	-0.399
SLIM	-0.094	-0.194	-0.175	-0.274

Table 5: PSP Averages for Protected Features by Method

	Loan Size		Country	
	Separate	Joint	Separate	Joint
BMF	+ 0.361	+ 0.381	+0.407	+ 0.330
BPR	+ 0.834	+0.302	+0.423	+0.328
SLIM	+0.679	+0.679	+0.544	+0.445

Table 6: Average gain in fairness

Obviously, our work here only touches on a few of the many approaches to fairness-aware reranking that have been explored in the literature. We concentrate on FAR [8] and FA*IR [19] because these are algorithms with very different approaches to the reranking problem. FAR uses simple re-scoring of results to promoted protected group items; FA*IR handles protected and unprotected items separately using search to create lists that satisfy a more stringent fairness constraint.

We use as baseline algorithms three well-studied recommendation algorithms: BPR [11], Biased Matrix Factorization [2] and SLIM [9]. These approaches were chosen because they cover a range of different underlying algorithmic concepts. Matrix factorization optimizes for prediction error, BPR optimizes for ranking loss, and SLIM uses an instance-based approach. These approaches are known to have different characteristics with respect to popularity bias and recommendation diversity, and those characteristic emerge in their interaction with the rerankers, with each base algorithm yielding vastly different NDCG values and more limited variations in PSP.

The Optuna implementation used here is just one method for automated exploration of the hyperparameter space of learning algorithms. Another recent development is the emergence of the AutoML technique [7], which can be applied to a more general class of problems, including neural architecture search and search over different algorithm types.

8 CONCLUSIONS AND FUTURE WORK

In this work, we have found no major differences in accuracy or fairness between reranking after optimizing for accuracy and joint optimization. Neither approach had a consistent advantage, meaning that either approach is acceptable if the primary goal is to minimize the fairness/accuracy trade-off. However, there are distinct advantages to each approach: for instance, reranking alone preserves the independence of the base algorithm results while offering a substantial computational time benefit. joint optimization allows for results from a greater search space as well as the potential of the integrated reranker to enhance results.

The results presented here concentrate exclusively on the Kiva dataset. This is an important application of recommendation, but it has some characteristics that set it apart from other applications in which recommender systems are commonly deployed, including streaming media, e-commerce and social media. In our future work, we plan to extend our study to additional datasets where provider-side fairness concerns arise.

As noted above, our work here has addressed provider-side fairness exclusively, looking at two different aspects of borrowers: loan amount and geographic location. In other recommender systems applications, for example, recommending jobs to job seekers, consumer-side fairness may be important. Some types of reranking algorithms, for example [10], have been employed in such settings, and we intend to explore whether the findings here translate to consumer-side fairness as well.

With regards to the joint optimization problem, it may be worth exploring further combinations of combined objective functions. 10% is low accuracy cost, but for some applications a smaller value might be desireable. Some of our preliminary research with Kiva suggests that, for their application, the threshold might be higher. We are also interested in exploring alternative formulations of the joint objective, for example multiplicative ones.

It is clear that our work here has surveyed only a small subset of the available algorithms, both for recommendation generation and for fairness-aware reranking. In our future work, we plan to expand the scope of our study to include additional algorithms in each category. Little work has been done on studying the fairness properties of recommendation algorithms using neural models and these would be obvious targets for future study.

We note also that reranking is not the only route to fairness-aware recommendation and that there is considerable literature on incorporating fairness objectives into recommendation models themselves. A natural extension of the research presented here is to explore whether such models can be optimized to provide a better combination of fairness and accuracy than a two-stage reranking pipeline. This question has not been thoroughly explored in the literature as most researchers have concentrated on one approach or the other. The combination of such fairness-aware recommendation models with reranking is yet another approach to consider. It is possible that joint optimization in such a pipeline might enable the placement of protected items into the initial round of results so that the reranker has better items to choose from.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under grant awards IIS-1911025 and IIS-2107577. Special thanks to Pradeep Ragothaman and Melissa Fabros of Kiva for their assistance, and to Nasim Sonboli for her work on the Kiva Recommendation Dataset.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing popularity bias in recommender systems with personalized re-ranking. In *The thirty-second international flairs conference*.
[2] Charu C Aggarwal. 2016. Model-based collaborative filtering. In *Recommender systems*. Springer, 71–138.
[3] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2623–2631.
[4] Robin Burke. 2017. Multisided Fairness for Recommendation. (July 2017). arXiv:1707.00093 [cs.CY] http://arxiv.org/abs/1707.00093
[5] Robin Burke, Nicholas Mattei, Vlasislav Grozin, Amy Voida, and Nasim Sonboli. 2022. Multi-agent Social Choice for Dynamic Fairness-aware Recommendation. In *Adjunct Publication of the 30th Conference on User Modeling, Adaptation and Personalization*. to appear.
[6] Michael Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. 2022. Fairness and Discrimination in Information Access. *Foundations and Trends in Information Retrieval* In press (2022).
[7] Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems* 212 (2021), 10–66.
[8] Weiwen Liu, Jun Guo, Nasim Sonboli, Robin Burke, and Shengyu Zhang. 2019. Personalized fairness-aware re-ranking for microlending. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 467–471.
[9] Xia Ning and George Karypis. 2011. Slim: Sparse linear methods for top-n recommender systems. In *2011 IEEE 11th international conference on data mining*. IEEE, 497–506.
[10] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. 2020. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of The Web Conference 2020*. 1194–1204.
[11] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
[12] Rodrygo LT Santos, Craig Macdonald, Iadh Ounis, et al. 2015. Search result diversification. *Foundations and Trends® in Information Retrieval* 9, 1 (2015), 1–90.
[13] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. ACM, New York, NY, USA, 2219–2228. https://doi.org/10.1145/3219819.3220088
[14] Nasim Sonboli, Robin Burke, Zijun Liu, and Masoud Mansoury. 2020. Fairness-aware Recommendation with librec-auto. In *Fourteenth ACM Conference on Recommender Systems*. 594–596.
[15] Nasim Sonboli, Robin Burke, Nicholas Mattei, Farzad Eskandanian, and Tian Gao. 2020. "And the Winner Is...": Dynamic Lotteries for Multi-group Fairness-Aware Recommendation. https://doi.org/10.48550/ARXIV.2009.02590
[16] Nasim Sonboli, Farzad Eskandanian, Robin Burke, Weiwen Liu, and Bamshad Mobasher. 2020. Opportunistic Multi-aspect Fairness through Personalized Re-ranking. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. 239–247.
[17] Nasim Sonboli, Masoud Mansoury, Ziyue Guo, Shreyas Kadekodi, Weiwen Liu, Zijun Liu, Andrew Schwartz, and Robin Burke. 2021. librec-auto: A Tool for Recommender Systems Experimentation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4584–4593.
[18] Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. 2021. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*. PMLR, 3–26.
[19] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1569–1578.
[20] Meike Zehlike, Ke Yang, and Julia Stoyanovich. 2022. Fairness in Ranking, Part I: Score-based Ranking. *ACM Computing Surveys (CSUR)* (2022).