Balancing between Fairness and Accuracy in Fairness-Aware Recommender Systems

NASIM SONBOLI

The increasing role of recommender systems in many aspects of the society makes it essential to consider how such systems may impact social good. Traditionally, recommender system's focus has been on optimizing for accuracy. However, the research in this field has shifted its focus to beyond-accuracy and socially sensitive properties of fairness. Much of the previous work on fairness has been on classification problems, work that is not necessarily extendable to recommendation problems, due to the main characteristics of recommender systems: personalization and being a multi-stakeholder setting. One of the key challenges is the tension between personalization and fairness goals. As an example, in a crowd-sourced loan recommendation scenario, the fairness goal of the system might be to increase the visibility of under-represented loans, while personalization is trying to tailor the recommendation for individual users. Some users might not like to lend money to certain loans. Therefore these two goals might be competing at times and we would need to balance between them.

Another challenge is that recommender systems facilitate transactions between different parties such as borrowers, lenders, etc.: recommendation fairness is not one problem. Multiple stakeholders might be involved and they might all seek fairness, while their goal of fairness might be different (due to different definitions of fairness), competing or conflicting. This work has primarily focused on developing recommendation approaches for multiple-stakeholders and through designing methods in which fairness metrics are jointly optimized along with recommendation accuracy. In this proposal, I present my five main contributions to the fairness-aware recommendation field, working to achieve reasonable fairness improvements with minimal accuracy loss: (1) balanced neighborhood SLIM method, (2) fairness-aware recommendation re-ranking (FAR) and personalized fairness-aware re-ranking methods (PFAR), (3) opportunistic fairness-aware re-ranking (OFAiR), (4) SCRUF, a framework for assessing and achieving fairness in dynamic environments. In addition, I present my contributions to reproducibility in the fairness-aware recommendation field through enhancements to the Librec-auto experimentation platform.

ACM Reference Format:

Nasim Sonboli. 2021. Balancing between Fairness and Accuracy in Fairness-Aware Recommender Systems. 1, 1 (March 2021), 64 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

After a period of substantially fast development in different aspects of digital systems, and with more and more decisions being delegated to algorithms, the society has begun to realize these systems that were intended to assist people in different tasks have ethical issues and can cause harm to individuals and the society.

Take as a real-world example the following study: [6] studied the distribution of ads on Facebook to understand potentially discriminatory impact in the visibility of different kinds of ads. They found that even when an advertiser wishes to have fair distribution of their ad, for example to ensure that an ad for a job opening is seen by people of all

Author's address: Nasim Sonboli.

Manuscript submitted to ACM

genders, the combination of relevance optimization and market dynamics results in disparate distribution of ads across racial and gender lines.

Discrimination caused by algorithms that are trained on biased data or by lack of a good design, propagation of bias [11], marginalization of minority groups in the society, inflation in the polarization of the society that can be caused by tight filter bubbles due to massive filter bubbles, and etc. are among the harms that algorithms can potentially cause. These problems have gained the attention of a multidisciplinary community from computer scientists, social scientists and legal scholars. Thus as a response, recent research has shifted from design of algorithms that pursue purely optimal outcomes with respect to an objective function into ones that also consider social impacts such as fairness.

1.1 Fairness in Machine Learning

Fairness, bias and discrimination are topics of considerable research interest in the recent years [20, 46, 108].

Much of the work in algorithmic fairness has been focused on classification methods with a myriad of definitions that has been proposed. The key definition are explained in [102].

One of the main divisions in fairness definitions comes from the way we assess and evaluate fairness and whether this evaluation is individual based or group based.

1.1.1 Group Fairness. In group fairness based definitions, a model's treatment of two or more groups with respect to a sensitive attribute (e.g. gender, race, ethnicity, etc.) is compared. In this method, the protected group(s) is designated with respect to a previously defined sensitive attribute and should be protected against discrimination. The sensitive attribute definition is usually rooted in anti-discrimination laws[11].This notion of fairness tries to ensure that algorithms don't impact the members of the protected group more adversely and disproportionately. Group-based methods of fairness has helped build the most prevalent structures to achieve and assess fairness [73, 75, 146, 147].

Statistical Parity. The notion of Statistical or Demographic Parity requires that two groups with a different sensitive group have the same chance of getting a positive result. This notion has been discussed under different names of avoiding disparate impact [56], independence [10] and anti-classification [42]. This measure is used to ensure a "fair" representation of different groups in different tasks such as ranking[124, 139, 145], and recommendation[51, 101].

Performance Parity. Performance Parity is another category of group fairness that requires equal error rates for different groups. Equality of opportunity or equality of true positive rates [64] that requires the positive classification rates to be independent of the protected attribute given the true label, equality of both true positive and false positive rates which is known as equalized odds [64], equality of mis-classification rates (e.g. equality of false negative rates aka lack of disparate mistreatment[144]) and equality of positive predictive values aka calibration, belong to this category. Performance Parity has also been studied as error parity in recommendation algorithms[49, 141].

1.1.2 Individual Fairness. Dwork et al. [47] observed that the demographic parity requirements can be met when qualified candidates from one groups and random candidates from the other groups were chosen. Thus, satisfying certain group based fairness notions might degrade fairness for individuals in a group. Therefore fairness might be a requirement on an individual level. Dwork et al. [47] introduces the concept of individual fairness [47] which posits that similar individuals with respect to the task at hand should be treated similarly or in other words should have similar probabilities of positive classification outcomes. One of the limitations of this method is the choice of similarity metric to compare individuals and whether this metric is unbiased. This method also doesn't place any requirement on the treatment of dissimilar individuals. Another individual fairness definition was proposed by [82]

for the problem of candidate set selection from diverse incomparable source sets. Choosing candidates for a research position from a diverse research communities with uncomparable research metrics (e.g. citation rates are different in different research communities) is an example of this problem. Meritocratic fairness requires that less qualified candidates are probabilistically almost never chosen over the more qualified candidates.

1.1.3 Harm. Independent to group and individual fairness, Crawford [43] defines two new fairness definitions which are connected to the harm caused by unfairness: (a) distributional harm that is cause by an inequitable distribution of a resource or opportunities, and (b) representational harm, where a system doesn't have an accurate representation of the society or where it systematically misrepresents certain groups.

1.1.4 Anti-classification & Anti-subordination. The motivation of fairness constructs also categorizes fairness definitions into two groups: anti-classification and anti-subordination. U.S. anti-discrimination law is rooted in anti-classification ideas which requires that the influence of a protected group should not play a role in the decision making process. This notion can also be called as disparate treatment. The main idea of anti-subordination is to actively work to reverse the effects of historical discriminatory patterns in the decision making processes [11].

Although these motivations are often very clear, their proper application is still vague [138]. It is also worth mentioning that it has been shown that satisfying different fairness notions at the same time is mathematically impossible and infeasible [41, 86]. Due to the competing and sometimes conflicting goals of different fairness definitions, different needs of various stakeholders involved, etc. we cannot have a system that is universally "fair". Thus, it is essential to pick a fairness notion that serves best in the specific context of the target application.

1.2 Fairness in Recommender Systems

Recommender systems are one of the most pervasive applications of machine learning in industry. They play a pivotal role in connecting users to relevant items or content throughout the web while not only users rely heavily on them but also content producers, sellers or information providers.

Consider a recommender system suggesting job opportunities to job seekers. Discriminatory recommendations in this system could mean that men and women with similar qualifications don't get recommendations of jobs with similar rank and salary. Or when women get similar recommendations just because of their demographic information not because of their qualifications. The system would therefore need to defend against biases in recommendation output, even biases that arise due to behavioral differences: for example, male users might be more likely to click optimistically on high-paying jobs.

Traditionally the focus of recommendation algorithms have been on accuracy and it was known that it is tied strongly with user satisfaction. Later on, the focus of these systems changed to *beyond accuracy* methods such as diversity, coverage, novelty, serendipity. This change of focus was supported by the literature that showed these properties in recommendation lists of users increase their overall satisfaction. In recent years, aligned with the change of focus on these beyond-accuracy and socially sensitive properties in machine learning, the social aspects of these algorithms has come to the fore.

Therefore achieving fairness in recommendation algorithms has gotten more attention. However, the goal of fairness isn't completely new in the recommender system literature. Alleviating the problem of popularity bias in recommendations [31] and ensuring equality or equity in long-tail recommendations [57] can be thought of achieving fairness for content providers in the systems. Group recommendation also tries to recommend items to users while considering and treating all the members of the group fairly [79]. However, the goal of fairness in recommendation goes

beyond one stakeholder and is not bounded to the previously mentioned problems. Rather it focuses on the aspects that are socially sensitive such as discrimination against sensitive-groups, under-representation of sensitive groups and preventing biases from creeping into these systems.

1.2.1 Challenges of Fairness in Machine Learning. Recommender systems have their unique challenges for investigating the fairness concepts and the methods that have been developed in other machine learning literature is not fully applicable in recommender systems. The role of personalization and multistakeholder nature of recommender systems add major additional complications to the problem of fairness in recommendations.

Personalization. There is a tension between the goals of personalization and fairness [103]. On the one hand, the goal of personalization is to find the best item(s) for each user while this list could be different for the users. On the other hand, fairness for providers means to give them equal visibility. So, one could simply divide the recommendation opportunities equally among providers. In this case personalization for consumers and fairness for providers are conflicting goals and achieving one means sacrificing the other one.

Additionally, recommendations become homogeneous over time in iterative environments [36] causing multiple issues like deteriorating popularity bias in a system. This issue both causes less visibility for less popular or marginalized item providers (provider unfairness), and can be unfair to marginalized consumers as popularity bias in the input data can cause the preference of 95% the users be overshadowed by only the preference of 5% of users who have rated mostly popular items (consumer unfairness) [55].

Multistakeholder aspects. Recommender systems exist to facilitate transactions between consumers, content providers. Thus, many recommendation applications involve multiple stakeholders and therefore may give rise to fairness issues for more than one group of participants [24]. As an example, Uber Eats is a multistakeholder setting, where consumers are users who order food, providers are restaurants who provide the food, Uber itself is the system, and drivers are the other stakeholders involved as well. Fairness concerns of any or all of these entities are important and necessary. Consumer fairness, is concerned with fair and equitable treatment of all the users in the system regardless of their membership to any protected group. For example ensuring that all the subgroups of users are receiving quality recommendations not only certain groups. Provider fairness is concerned with a fair treatment of content providers or content creators. For example, by making sure they are represented in the recommendations fairly and have equal opportunities to benefit from the system. Subject fairness is concerned with fair treatment of the content, people or entities in a system. For example, ensuring that recommendations do not systematically under-represented specific segments of the society or certain content.

Therefore, recommendation fairness is not one problem. Multiple stakeholders might be involved and although they might all seek fairness, their goal of fairness might be different (due to different definitions of fairness), competing or conflicting.

For the previous two challenges, take the job recommendation scenario as an example. Since a recommender system is often in the position of facilitating a transaction between parties, such as job seeker and prospective employer. Defending biases towards both parties may be important. For example, at the same time that a job recommender system is ensuring that male and female users to get recommendations with similar salary distributions, it might also need to ensure that jobs at minority-owned businesses are being recommended to the most desirable job candidates at the same rate as jobs at majority-owned businesses such as white-owned businesses.

Due to issues as such, researchers in recommender systems have begun to seek ways to ensure fairness in the results that such systems produce for multiple parties (stakeholders).

Defeating such biases becomes more challenging when a system's main goal is personalization. Even in the example before, some users might prefer a somewhat lower-paying job if it had other advantages: such as a shorter commute time, or better benefits. Personalization is so important that if a job seeker does not find the system's recommendations valuable, he or she may ignore the fair aspect of the system and may migrate to a competing platform. The same is true of job providers; a company may choose other platforms on which to promote its job openings if a given site does not present its job ads as recommendations or does not deliver acceptable candidates. Therefore the fairness goal has some tensions with the personalization goal.

Other Challenges. As mentioned previously, we cannot achieve a universally fair system, therefore for each problem, it's essential that we consider the target stakeholders, the definition of harm or unfairness and the specific metrics for measuring harm or integrating in the system to avoid harm. These elements are important in order to define, integrate and assess fairness in recommendation algorithms.

Lack of appropriate data to study fairness goals is another challenge with which we have to deal. We might need sensitive attributes (e.g. gender, race, etc.) for the stakeholder entities but sharing and using this data might have privacy, legal or ethical concerns. Some of solutions that were used for this problem were using crowd sourcing[17], or professional annotation, integrating different datasets, using inference methods to impute demographic information, generating synthetic datasets [27], or training algorithms without demographics[72]. But, all of the previous solutions bring their specific limitations to the method.

One of the other key challenges in this area is the domain-specificity of recommendation environments. The utilities that are delivered to each class of stakeholder are highly dependent on the type of item being recommended, the social function of the platform, and the interactions that it enables. It is therefore difficult to find appropriate data sets for experimentation and challenging to generalize across recommendation scenarios.

1.3 Fairness for Multi-stakeholders

Fairness notions can be defined, assessed and integrated to the algorithms for different stakeholders. For each stakeholder here we investigate the prior work and categorize it based on previous fairness notions such as group fairness, individual fairness, etc.

1.3.1 Consumer Side Fairness. Consumer fairness or (C-fairness) is concerned with the fair treatment of consumers (of recommendations) and the impact that recommendations have specifically on marginalized groups. Such objectives are sometimes required by law.

Individual Fairness. In collaborative filtering, the recommendations are build based on the similarity of users. And the goal is to use these similarities to recommend items to users in a personalized way. In other words, the recommendations of similar users are similar, thus they are treated similarly. This property might appear similar to the definition of individual fairness, although the metric based on which the similarity is calculated is different. As an example, collaborative filtering uses the rating behavior of users whereas individual fairness looks at the user demographic information to calculate similarity which is hard to get by. Another issue is that while similar users will be treated similarly, but since the data is not rich for minority groups, statistically, all of the users in that group might be treated unfairly when compared to the whole. As an example, if women in a job recommendation platform tend to click on lower

paying jobs, and since their rating behavior is similar, all of them get equally low paying jobs. Therefore, individual fairness and personalization might have similar goals, but the similarity metric and the information they use is different.

Group Fairness. To integrate group fairness in recommendation algorithms, we can imagine the utility that the consumers receive from recommendations and whether it is distributed equally or whether all the individuals are benefiting equally from it. Similar to the group fairness definition in machine learning, here also we have both categories of (a) statistical or demographic parity and (b) performance parity. The first category focuses on sub-group representation while the latter focuses on subgroup loss (or gain). Since the performance parity is the statistical parity in performance metrics, it has been called statistical parity in previous research.

we measure the distribution of the genders of the authors of books in user rating profiles and recommendation lists produced from this data.

[51] demonstrates that the distribution of the authors' genders in user rating profiles and recommendation lists produced from this data are very different while the collaborative filtering algorithms propagate this bias. As a conclusion, to achieve demographic parity of authors' genders in the recommendations, we need to ensure that group (authors' genders) proportions in the recommendation sets should be similar to group proportions in input ratings. Performance parity in recommendation is calculated based on the effectiveness of recommendations and whether different subgroups are experiencing the same accuracy or error. Although not all of the definitions of error in classification can be applied to recommendation settings. Similar to [77], Yao and Huang [141] have designed different error-based fairness metrics for collaborative filtering such as value unfairness, over-representation, under-representation, etc. They have compared the discrepancy between the actual and predicted ratings for protected and unprotected groups or inconsistencies between the predicted ratings for these groups. [49] has performed an off-line top-N evaluation of several collaborative filtering algorithms and has compared the results of different user demographics based on their NDCG. [127] introduces the concept of miscalibration in recommendations which has been used detected as consumer unfairness. Miscalibration happens when the item preferences of the users in their profile isn't covered in the recommendations they receive. [94] has discusses that usually smaller or niche subgroups receive more miscalibrated recommendations compared to bigger or more popular subgroups.

1.3.2 Provider Side Fairness. Provider-side fairness or (P-fairness) is concerned with treating the suppliers of information or items that are being recommended fairly. We can think of recommendation opportunities as a resource and the fair distribution of those opportunities among providers as provider fairness. Much of the research on diversity and decreasing popularity bias, contributes to provider fairness. Although, provider fairness derives from the goal of social justice and promotes the content from the underprivileged groups to provide more opportunities for them to be discovered. To achieve P-fairness, [101] tried to ensure that different sub-groups are similarly represented in the recommendations.

The utility defined in this context is mostly defined as exposure. A recommendation list is a short list that provides limited opportunities to expose items to consumers. According to the user attention pattern, the items that are on top of the list, receive more attention and this attention decreases as we go down in the list. Therefore, not all the positions in a list have equal utilities and only one item in the whole list can have the most valuable position and receive the most benefit [45]. Therefore provider utility is calculated over all the recommendation lists delivered to all the users, while to calculate each consumers utility we only need to look at each consumer's recommendation and nothing more. Most of the metrics introduced for provider utility are based on NDCG [18].

Individual Fairness. individual fairness for providers mean that similar items should receive similar utility from the recommender system. As an example, items that bring the same utility to a user (the user likes them both), are considered similar and should receive the same exposure in a recommendation list. [18] aggregates each item's attention and relevance over multiple rankings and assumes the providers are being treated fairly if the attention they have received from users are proportional to their relevance. The similarity of providers can be calculated in many other different ways and is an open research area.

Group Fairness. Group fairness for providers is concerned with a fair treatment of different provider sub-groups. This goal should be aligned with the goal of personalization which considers users preferences so it doesn't recommend provider groups to users that don't like them. [76] proposes that the recommendations outcomes should be statistically independent of a sub-group's protected attribute in order to have group fairness. In this case, the probability that an item shows up in a recommendation list is independent of its sensitive attribute. Another method is to ensure that there is a fair representation of providers in the recommendation lists. Unfairness in this case is when there is a big divergence between the distribution of the provider groups in the lists and the target distribution. [44, 139]. This divergence can be calculated using KL-divergence, difference in probabilities, odds ratio, etc. [18] calculates the provider groups fairness by aggregating expected exposure over multiple rankings. Fairness in this context happens when each provider group receives an appropriate level of exposure. [14] incorporates a fair construct (disparate mistreatment) in BPR[111] loss function. In this construct, a fair ranking happens when ranking a relevant item over an irrelevant item is independent of its group membership. Their pairwise fairness objective is defined in two way: once between groups and once within groups.

1.3.3 *Other stakeholders.*

Besides the consumers and providers, there might be other stakeholders in the system whose fairness matters. In the Uber Eats scenario, besides considering fairness for users (who place orders) and restaurants (providers), we might want to increase fairness for drivers as well. For example, we might not want to overbook one driver, while other drivers stay in the queue to take delivery orders. Subject fairness is another instance of this type, where subject is a stakeholder entity. And the fair goal might be having a fair distribution of the subjects of items being recommended. For example, in news platforms, to avoid polarization in the society, we might want to have a fair representation of different points of views, or giving a fair coverage to different topics and avoiding certain popular topics from monopolizing news feeds.

1.3.4 *Dynamic Fairness.*

To define, measure and incorporate fairness definitions in algorithms, we should take into account many different aspects of fairness as we mentioned above. Another important aspect to consider is measuring the dynamics of fairness. Recommendation engines, change over time, as they interact with their users, attract new users and lose other users in the process. Therefore achieving fairness in one iteration might not be enough and might overlook these temporal changes. As an example, in these systems, feedback loops might occur and this phenomena causes the system to pay more attention to popular/dominant subgroups of users [66] and therefore lose their under-represented sub-groups (either in consumers or providers). [150] analyzes the dynamics of fairness in sequential decision-making and tries to achieve a more balance performance which improves user retention. [36] studies how recommendations become more and more homogeneous in iterative environments which leads to inequity of exposure among items.

1.4 Achieving Algorithmic Fairness

To achieve algorithmic fairness, interventions can be made at different steps of the processing pipeline. [59] provides a broad overview of these methods.

1.4.1 Pre-processing. Pre-processing methods focus on compensating the existing biases in the dataset. [37] suggest different data collection enhancements to compensate for the biases that occur due to data imbalance. In order to do so, if there is an under-represented group in the data, by collecting more data on that group or imputing the fundamental features of that group, we can see improvements in the performance metrics automatically. [56] modifies the numerical attributes in the data to equalize their marginal distributions conditioned on the sensitive attribute. In this way, these distributions will be independent of the sensitive attribute and therefore the outcome of the machine learning models which are trained on this data will be independent of the group membership. [63] suggests modifying the values of the attributes and labels in the data such that unfair association rules cannot be mined from the dataset. Some other approaches create intermediary (lower-dimensional) representations of the data points so as to hide the information about sensitive attributes, while keeping the utility of the modified data for the required task [89, 146]. Despite all this work, except a few exceptions [49], there isn't much work in the recommenders systems field that focuses on de-biasing the data in recommender systems. However, many of these work in machine learning are applicable to the data that is appropriate for recommender systems. I recognize the existing gap in this topic in the field, however, my current work doesn't contribute to this section.

1.4.2 In-processing. In-processing approaches try to improve the fairness of results by modifying the algorithms and by integrating fairness notions in their loss functions. Therefore the problem will turn into a multi-objective optimization problem that seeks to simultaneously maximize utility and fairness. These types of algorithmic interventions, sometimes take the form of regularizers to control certain structural properties of the model in the optimization functions. Regularizers are usually used to control the complexity of the model and to prevent the model from overfitting, although they can capture unfairness of the model as well. For example, [144] proposes to add fairness constraints on top of the accuracy constraints in the optimization objective of a classifier. Fairness regularization has been used in other classification and regression problems such as [13, 75] and recommendation problems. For example, [76**?**] proposes to add an independence term to the loss function that penalizes any correlations between the sensitive attribute and the predicted ratings. This term can also be added to achieve consumer-side fairness[74]. They also propose multiple non-independence measures as well such as the difference in mean ratings between groups and the mutual information between the predicted ratings and the sensitive attributes. [141] also uses a regularization approach to minimize disparate rating predictions errors rather than recommendation errors. [15] adds a penalty term to their pairwise ranking loss function, to ensure that the difference between the ranking scores of the relevant and irrelevant items is uncorrelated with the relevant item's sensitive attribute. It is also possible to directly optimize a learning-to-rank such as [45] that uses such method to achieve equal expected exposure.

In section 2, I present Balanced Neighborhood method as an in-processing algorithm with the goal of balancing between fairness and accuracy term. We realize unfair recommendation in neighborhood based models, could be a cause of having a neighborhood that is too homogeneous. To achieve fairness in this approach, we added a regularizer that ensures a diverse neighborhood for users. In this way, we prevent the algorithm to form unfairly homogeneous neighborhoods. Additionally, our goal is to reach a balance between fairness and accuracy.

Despite all the progress in this approach to improve fairness, the goal of accuracy and fairness can contradict sometimes. Since traditionally a lot of these algorithms' objective functions are designed to achieve accuracy, adding fairness notions as an extra constraint might prevent the objective functions to converge. Therefore, post-processing methods provide alternatives where in-processing modification of algorithms is not fruitful.

1.4.3 Post-processing. Post-processing approaches focus on modifying the outputs of the algorithms to satisfy a fairness criteria. In these methods, fairness constraints will not interfere with the goals of the objective function, rather they intervene after the output is produced. This approach can be applied both to classification and recommendation problems. In [58], the proposed method tries to shift the boundaries of the already trained classifiers to achieve statistical parity with minimal accuracy loss. [64] tries to balance the true positive rates of different groups by modifying the decision score thresholds of a trained classifier. [73] proposes a methodology that relabels the nodes of a decision tree classifier in order to ensure demographic parity.

Fairness can also be improved by re-ranking the output lists which were produced with the goal of achieving a high relevance. There are two main approaches of reranking: (a) those that treat the problem as a global optimization task and try to improve fairness with respect to the entire list of recommendations and (b) those methods that focus on the fairness of single lists one at a time. An example of the first approach is [131] that proposes a constrained optimization-based method to enhance fairness (item exposure) for multiple provider groups, avoid unfairness towards under-represented groups and ensures a minimum degree of diversity for consumers. These methods are useful for occasions when recommendations are generated and cached in advance. A more common approach is to re-rank individual lists as they are generated. Such approaches use methods like MMR (Maximal Marginal Relevance) [32] or xQuAD [121] that were presented in the information retrieval literature. These methods propose a greedy list expansion approach, where the re-reranked list is generated by adding new items to this list where it satisfies a fairness or diversity criteria. This approach also provides the benefit of controlling the balance between the accuracy and the fairness goal. [103] use a re-rank approach to enhance provider exposure while preserving relevance. [61] uses a greedy approach to produce rankings of job-candidates that have a fair distribution of their demographic attributes, simultaneously optimizing for fairness and relevance. [145] uses A-star algorithm for reranking to achieve fairness in a ranked list at depth K. The goal here is to re-arrange the ranked lists to meet a fair distribution of items from different protected groups while keeping the quality of ranked lists as high as possible.

Overall, re-ranking approaches offer a number of advantages. First, the trade-off between accuracy and fairness can be tuned without re-learning the recommendation model (as we have to do in in-processing approaches). Second, researchers have found that re-ranking can achieve better trade-offs versus accuracy with this type of model [3, 96]. Due to this advantages we choose to use the latter method to increase fairness. In section 4, I present three re-ranking algorithms: Fairness-Aware re-Ranking, Personalized Fairness-Aware re-Ranking and Opportunistic Fairness-Aware re-Ranking. All of these methods are greedy approaches that focus on the fairness of single lists one at a time and they are based on information retrieval approaches that are intended to increase aggregate diversity.

1.5 Summary of Contributions

My work here has primarily focused on developing recommendation approaches in which fairness metrics are jointly optimized along with recommendation accuracy. I have structured the problem in a way that the balance between these two goals can be controlled and set using a hyper-parameter. However, my goal is to improve fairness, while preserving the accuracy as much as possible. Throughout the following work, I recognize all the stakeholders in a recommendation

setting and their fairness concerns. Although I considered to improve fairness for both consumers and providers in the Fairness through Balanced Neighborhoods and have demonstrated the fairness improvements for both parties, in the rest of my work, the stakeholder of interest is the provider. All of the following work's fairness definitions are group-based definitions not individual-based fairness definitions. In the following methods, I have used performance parity metrics to assess fairness of the results where these metrics where both accuracy-based and exposure-based.

I present my five main contributions to the fairness-aware recommendation field. After a thorough literature review of fairness in machine learning, I present

- **(1) Fairness through Balanced Neighborhoods**: Here, I present an in-processing method with the goal of improving fairness while preserving as much accuracy as possible. By adding a regularizer to the objective function, for each user a diverse neighborhood is generated. These user neighborhoods play an important role in creating the recommendations of users. A diverse neighborhood for each user ensures a non-biased (or less biased) set of recommendations. Here our, goal is to reach a balance between fairness and accuracy. Additionally, I define the concept of multi-sided fairness and demonstrate improvements in fairness both for two sides of the recommendation setting: consumers and providers.
- **(2) Fairness-Aware Recommendation Re-ranking (FAR) and personalized fairness-aware re-ranking methods (PFAR)**: I present two greedy re-ranking approaches here which are both based on XQuAD (an information retrieval method to improve diversification). Both methods are designed to improve the fairness / accuracy tradeoff for the protected providers. Other contributions of this project is the definition of a group-fairness metric for providers and the adaptation of fairness concepts to micro-finance systems. Therefore, the designed methods here are purposed for a loan recommendation scenario although they can be adapted to other contexts.
- **(3) Opportunistic Fairness-Aware Re-ranking (OFAiR)**: Here, I introduce the concept of opportunistic fairness. Users are not willing to experience diversity in every aspect of their recommendation. We detect the areas in which the users show willingness to see diversity and we consider them as opportunities to increase fairness without sacrificing much accuracy. Additionally, I use a greedy re-ranking approach based on Maximal Marginal Relevance or MMR (an information retrieval method to improve diversification), with the goal of improving the accuracy / fairness trade-off for multiple provider groups at the same time. This post-processing approach is one of the few approaches that defines multi-aspect fairness and designs a method to improve different fairness goals of various providers in a simultaneous way. As an example, improving the visibility of impoverished loan borrowers with respect to different aspects such as: region of the world, their demographic information, loan amount, the economic sector, etc.
- **(4) SCRUF framework**: Here, I propose a novel framework for recommender systems called *Social Choice for Re-ranking Under Fairness* (SCRUF). This framework is appropriate for dynamic environments where multiple fairness concerns matter. In this method, we use group-based exposure-based fairness definitions for providers.
- **(5) Fair Librec-auto**: I present my contributions to `librec-auto`, an open-source Python package providing a wrapper for the well-known LibRec which provides implementation of various recommendation algorithms. My contributions to this project were the implementation and addition of in-processing and post-processing fairness algorithms and fairness metrics.

2 REGULARIZATION

In this section, we examine applications in which fairness with respect to consumers and to item providers is important. We focus on integrating a group fairness definition to the objective function of the well-known sparse linear method (SLIM) via adding a regularizer. And we show that variants of SLIM can be used to negotiate the tradeoff between fairness and accuracy.

2.1 Balanced Neighborhoods in Recommendation

In [146], the authors impose a fairness constraint on a classification by creating a *fair representation*, a set of prototypes to which instances are mapped. The prototypes each have an equal representations of users in the protected and unprotected class so that the association between an instance and a prototype carries no information about the protected attribute.

As noted above, the requirement for personalization in recommendation means that we have as many classification tasks as we have users. A direct application of the fair prototype idea would aggregate many users together and produce the same recommendations for all, greatly reducing the level of personalization and the recommendation accuracy. This idea must be adapted to apply to recommendation.

One of the fundamental ideas of collaborative recommendation is that of the *peer user*, a neighbor whose patterns of interest match those of the target user and whose ratings can be extrapolated to make recommendations for the target user. One place where bias may creep into collaborative recommendation may be through the formation of peer neighborhoods.

Consider the situation in Figure 1. The target user here is the solid square, a member of the protected class. The top of the figure shows a neighborhood for this user in which recommendation will be generated only from other square users, that is, other protected individuals. We can think of this as a kind of segregation of the recommendation space. If the peer neighborhoods have this kind of structure relative to the protected class, then this group of users will only get recommendations based on the behavior and experiences of users in their own group. For example, in the job recommendation example that was mentioned previously, women would only get recommendations of jobs that have interested other women applicants, potentially leading to very different recommendation experiences across genders.

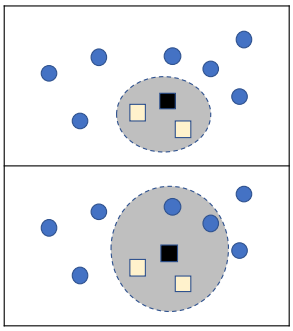


Fig. 1. Unbalanced (top) and balanced (bottom) neighborhoods

To enhance the degree of C-fairness in such a context, we introduce the notion of a *balanced neighborhood*. A balanced neighborhood is one in which recommendations for all users are generated from neighborhoods that are balanced with

respect to the protected and unprotected classes. This is shown in the bottom half of Figure 1. The target has an equal number of peers inside and outside of the protected class. In the case of job recommendation discussed before, this would mean that female job seekers get recommendations from some female and some male peers.

There are a variety of ways that balanced neighborhoods might be formed. The simplest way would be to create neighborhoods for each user that balance accuracy against group membership. However, this would be highly computationally inefficient, requiring the solution of a separate optimization problem for each user.

In this research, we explore an extension of the well-known Sparse Linear Method (SLIM) [105]. SLIM is well-known as a state-of-the-art technology for collaborative recommendation. It is a generalization of item-based recommendation in which a regression coefficient is learned for each $\langle user, item \rangle$ pair. It can be slower to optimize than factorization-based methods, but for our purposes, it has the important benefit that the learned coefficients are readily interpretable with regard to group membership. Our extension of SLIM uses regularization to control the way different neighbors are weighted, with the goal of achieving balance between protected and non-protected neighbors for each user.

2.1.1 *Sparse Linear Method*.

SLIM learns $\langle user, item \rangle$ regression weights through optimization, minimizing a regularized loss function. Although this is not proposed in the original SLIM paper, it is possible to create a user-based version of SLIM (labeled SLIM-U in [151]), which generalizes the user-based algorithm in the same way.

Assume that there are M users (a set U), N items (a set I), and let us denote the associated 2-dimensional rating matrix by R. SLIM is designed for item ranking and therefore R is typically binary. We will relax that requirement in this work, We use u_i to denote user i and t_j to denote the item j. An entry, r_{ij}, in matrix R represents the rating of u_i on t_j.

SLIM-U predicts the ranking score $\hat{s}$ for a given user, item pair $\langle u_i, t_j \rangle$ as a weighted sum:

$$\hat{s}_{ij} = \sum_{k \in U} w_{ik} r_{kj}, \tag{1}$$

where $w_{ii} = 0$ and $w_{ik} >= 0$.

Alternatively, this can be expressed as a matrix operation yielding the entire prediction matrix $\hat{S}$:

$$\hat{S} = WR, \tag{2}$$

where W is an MxM matrix of user-user weights. For efficiency, it is very important that this matrix be sparse.

The optimal weights for SLIM-U can be derived by solving the following minimization problem:

$$\min_W \frac{1}{2} \|R - WR\|^2 + \lambda_1 \|W\|^1 + \frac{\lambda_2}{2} \|W\|^2, \tag{3}$$

subject to $W > 0$ and $diag(W) = 0$.

The $\|W\|^2$ term represents the ℓ_2 norm of the W matrix and $\|W\|^1$ represents the ℓ_1 norm. These regularization terms are present to constrain the optimization to prefer sparse sets of weights. Typically, coordinate descent is used for optimization. Refer to [105] for additional details.

Neighborhood Balance. Recall that our aim in fair recommendation is to eliminate segregated recommendation neighborhoods where protected class users only receive recommendations from other users in the same class. Such neighborhoods would tend to magnify any biases present in the system. If users in the protected class only are recommended certain items, then they will be more likely to click on those items and thus increase the likelihood that the collaborative system will make these items the ones that others in the protected group see.

To reduce the probability that such neighborhoods will form, we use the SLIM-U formalization of the recommendation problem, but we add another regularization term to the loss function, which we call the *neighborhood balance* term. To describe this term, we will enrich our notation further by indicating U^+ to be the subset of U containing users in the protected class with the remaining users in the class U^-. Let W_i^+ be the set of weights for users in U^+ and W_i^- be the corresponding set of weights for the non-protected class. Then the neighborhood balance term b_i for a given user i is the squared difference between the weights assigned to peers in the protected class versus the unprotected class.

$$b_i = (\sum_{w^+ \in W_i^+} w^+ - \sum_{w^- \in W_i^-} w^-)^2 \tag{4}$$

A low value for the neighborhood balance term means that the user's predictions will be generated by weighting protected and unprotected users on a relatively equal basis.[1]

Another way to express this idea is to create a vector p of dimension M. If u_i is in U^+, then $p_i = 1$; if u_i is in U^-, then $p_i = -1$. Then, the sum expressed above can be rewritten as $b_i = (p^T \cdot w_i)^2$. By adding up this term for all users and adding it to the loss function, we can allow the optimization process to derive weights with neighborhood balance in mind. This adapted version of SLIM-U we will call *Balanced Neighborhood SLIM-U* or BN-SLIM-U.

As in the case of the original SLIM implementation, we can apply the method of coordinate descent to optimize the objective. The full loss function is as follows:

$$L = \frac{1}{2} \|R - WR\|^2 + \lambda_1 \|W\|^1 + \frac{\lambda_2}{2} \|W\|^2 + \frac{\lambda_3}{2} \sum_{i \in U} \left(\sum_{k \in U} p_i w_{ik} \right)^2, \tag{5}$$

where $w_{ii} = 0$ and $w_{ik} >= 0$ and where λ_3 is a parameter controlling the influence of the neighborhood balance calculation on the overall optimization

This loss function retains the property of the original SLIM algorithm in that the rows of the weight matrix are independent, and the weights in each row (those for each user) can be optimized independently. The algorithm chooses one w_{ik} weight and solves the optimization problem for that weight, repeating over all the weights until convergence is reached. If we take the derivative of L with respect to a single weight w_{ik}, we obtain

$$\frac{\partial L_i}{\partial w_{ik}} = \sum_{j \in I} (r_{ij} - \sum_{l \in U'} w_{il} r_{lj}) + w_{ik} \sum_{j \in I} r_{kj}^2 + \lambda_1 + \lambda_2 w_{ik} + \lambda_3 p_k \sum_{l \in U'} p_l w_{il} \tag{6}$$

where $U' = U - \{u_i, u_k\}$.

We then set this derivative to zero and solve for the value of w_{ik} that produces this minimum. This becomes the coordinate descent update step.

[1] Note that this is a class-blind optimization that tries to build balanced neighborhoods for both the protected and unprotected users. It is also possible to formulate the objective such that it only impacts the protected class and we leave this option for future work.

$$w_{ik} \leftarrow \frac{S\left(X_{ik}, \lambda_1\right)_+}{\sum_{j \in I} r_{kj}^2 + \lambda_2 + \lambda_3}$$
$$X_{ik} = \sum_{j \in I} \left(r_{ij} - \sum_{l \in U'} w_{il} r_{lj}\right) + \lambda_3 p_k \sum_{l \in U'} p_l w_{il} \tag{7}$$

where $S()_+$ is the soft threshold operator defined in [60].

Item-based neighborhoods. As noted above, some applications may require P-fairness: making the recommendation outcomes fair relative to the items being recommended. In our micro-finance example, the operators of this site have the goal of providing equal exposure to loans from different geographic regions. To address the P-fairness case, we can use an analogous approach using item neighborhoods and item weights, ensuring that items in a protected group are in neighborhoods that have balanced membership of items from the unprotected group. The derivation of the loss function is exactly analogous, yielding another variant of the SLIM algorithm that we refer to as *Balanced Neighborhood SLIM* or BN-SLIM.

2.1.2 *Methodology.*

In order to evaluate our balanced neighborhood approach, we conducted separate sets of experiments in both consumer- and provider-fairness. It is very difficult to find datasets that contain the kind of features that would be necessary to evaluate fairness-aware recommendation algorithms, especially related to user demographics in sensitive application areas such as employment.

For the purposes of this paper, we are using the well-known MovieLens 1M dataset [65], which contains gender information for each user, as well as ratings of 4,000 movies by 6,000 users. Movie recommendation is, of course, a domain of pure individual taste and therefore not an obvious candidate for fairness-aware recommendation. Following the example of [140], our approach to construct an artificial equity scenario within this data for expository purposes only, with the understanding that real scenarios can be approached with a similar methodology.

Our consumer-fairness scenario centers on movie genres. It can be seen in this data that there is a minority of female users (1709 out of the total of 6040). Certain genres display a discrepancy in recommendation delivery to male and female users. For example, in the "Crime" genre, female users rate a very similar number of movies (average of 0.048% of female profiles vs 0.049% of male profiles) and rate them similarly: an average rating of 3.7 for both female and male users. However, our baseline unmodified SLIM-U algorithm recommends in the top 10 an average of 1.10 "Crime" movies per female user as opposed to 1.18 such movies to male users. We are still exploring the cause of this discrepancy, but it seems likely that there are influential female users with a lower opinion of this genre.

Given that the rating profiles are similar but the recommendation outcomes are different, we can therefore conclude that the female users experience a deprivation of "Crime" movies compared to their male counter-parts. Similar losses can be observed for other genres. We are not asserting that there is any harm associated with this outcome. It is sufficient that these differences allow us to validate the properties of the BN-SLIM-U algorithm.

Our goal, then, is to reduce or eliminate genre discrepancies with minimal accuracy loss by constructing balanced neighborhoods for the MovieLens users. The p vector in Equation 7 therefore will have a 1 for female users and a -1 for male users. In the experiments below, we compare the user-based SLIM algorithm in its unmodified form and the balanced neighborhood version BN-SLIM-U.

In evaluating fairness of outcome, we use a variant of what is known in statistics as *risk ratio* or *relative risk* (RR)[116]. We measure what is effectively *relative opportunity*. In other words, we measure the observed probability of protected class items being recommended divided by the probability of unprotected class items being recommended. In our MovieLens experiments, we measure the number of movies in protected and unprotected genres included in recommendation lists as the measure of outcome quality. We construct a consumer-side equity score, $E_c@k$ for recommendation lists of k items, as the ratio between the outcomes for the different groups. Let $P_i@k = \rho_1, \rho_2, ..., \rho_k$ be the top k recommendation list for user i, and let $\gamma()$ be a function $\rho \rightarrow \{0, 1\}$ that maps to 1 if the recommended movie is in a protected genre. Then:

$$E_c@k = \frac{\sum_{i \in U^+} \sum_{\rho \in P_i@k} \gamma(\rho)/|U^+|}{\sum_{i \in U^-} \sum_{\rho \in P_i@k} \gamma(\rho)/|U^-|} \tag{8}$$

$E_c@k$ will be less than 1 when the protected group is, on average, recommended fewer movies of the desired genre. It may be unrealistic to imagine that this value should approach 1: the metric does not correct for other factors that might influence this score – for example, female users may rate a particular genre significantly lower and an equality of outcome should not be expected. While the absolute value of the metric may be difficult to interpret, it is still useful for comparing algorithms. The one with the higher $E_c@k$ is providing more movies in the given genre to the protected group. Note that this is an additive, utilitarian measure of outcome equity and does not take into account variations in user experience. More nuanced measures of distributional equity, including Pareto improvement, we leave for future work.

As in any multi-criteria setting, we must be concerned about any loss of accuracy that results from taking additional criteria into consideration. Therefore, we also evaluate the ranking accuracy of our algorithms in the results below. The measure that we use is normalized discounted cumulative gain (NDCG) measured at a specific list length. In this measure, an item appearing on a recommendation list accrues "gain" according to its position on the list – thus the discount. The measure is normalized by comparing the algorithm's performance to the best ranking that could have been achieved.

Let $P_i@10$ be a list of retrieved list of length 10 and let τ be an indicator function that is 1 for movies that the user liked and 0 for others. Then, DCG@10 is computed as

$$DCG@10 = \sum_{k=1}^{10} \frac{\tau(\rho_k)}{log_2(k+1)} \tag{9}$$

NDCG@10 is this DCG@10 value divided by the optimal DCG, which occurs when all of the movies liked by the user and appearing the test set are ranked at the top of the list in their order of preference.

Provider fairness. To evaluate our approach for provider fairness, we are using a dataset extracted from the Kiva.org microlending site using the site's API[2]. Again, we have constructed our own scenario using this data, focusing on geographic region. In our dataset, we find that there are some geographic regions with a higher than average number of unfunded loans. In these regions, borrowers have a lower probability of getting the desired capital. See Table 1.

For the purposes of our experiments, we will assume that one of the goals of a microlending site is to equalize access to capital across geographic regions. Kiva does not currently offer personalized recommendation of loans to its

[2]http://build.kiva.org/

Category	Region	Unfunded %
Unprotected	North America	1.73
	Eastern Europe	0.99
	South America	4.33
	Asia	6.70
Protected	Africa	10.57
	Middle East	13.23
	Central America	8.81

Table 1. Percentage of unfunded loans by region

users, but if it did, a fairness-aware recommendation approach could be used to promote the loans of borrowers in the underserved regions.

We will therefore treat the under-represented regions collectively as the protected group and the other regions as the unprotected group. This enables us to use our item-based neighborhood balance algorithm described above. A more fine-grained approach to geographic equity that tries to balance across all regions would require additional algorithmic development and is left for future work.

Again, we will represent fairness as a ratio of outcomes. It is simpler to compute in this case, as we are not dividing the recommendations by genre. The provider-side equity score, $E_p@k$, is defined on recommendation lists of k items. Let L^+ be the set of loans in the test set that are from the protected regions, and L^- be the corresponding set from the unprotected regions. Also, let $\pi^+()$ be an indicator function $\rho \rightarrow \{0, 1\}$ that maps to 1 if the recommended loan is from a protected region and π^- is a similar function for the unprotected regions. Then:

$$E_p@k = \frac{\sum_{i \in U} \sum_{\rho \in P_i@k} \pi^+(\rho)/|L^+|}{\sum_{i \in U} \sum_{\rho \in P_i@k} \pi^-(\rho)/|L^-|} \tag{10}$$

$E_p@k$ will be less than 1 when loans from the protected regions are appearing less often on recommendation lists. As with E_c, this is a utilitarian measure, summing over all borrower regions, and does not speak to the the distribution across individual borrowers. Like E_c, it does not take the rank of recommended items into account.

2.1.3 ***Results***.

We implemented the SLIM-U, BN-SLIM, and BN-SLIM-U algorithms using LibRec 2.0 [62], and used its existing implementation of SLIM. We used 5-fold cross-validation as implemented within the library.

Consumer fairness: MovieLens. Within the MovieLens 1M dataset, we selected the five genres on which the SLIM-U algorithm produced the lowest equity scores: "Film-Noir", "Mystery", "Horror", "Documentary", and "Crime". The parameters were set as follows: $\lambda_1 = 0.1$, $\lambda_2 = 0.001$, and (for BN-SLIM-U) $\lambda_3 = 25$[3].

Figure 2 shows the results of the experiment in terms of the equity scores for each genre. Perfect equity (1.0) is marked with the dashed line. As we can see, in every case, the balanced neighborhood algorithm produced an equity score closer to 1.0 than the unmodified algorithm. The largest jump is seen in the "Horror" genre, about 0.09 in the equity score or around 10%.

In terms of accuracy, there was only a small loss of NDCG@10 between the two conditions. See Table 2. The difference amounts to approximately 2% loss in NDCG@10 for the balanced neighborhood version.

[3]Because the balance term measures the difference in weights, it tends to be much smaller than the terms that measure the sums of weights. Therefore, the regularization constant must be much higher for the balance term to have an impact on the optimization.

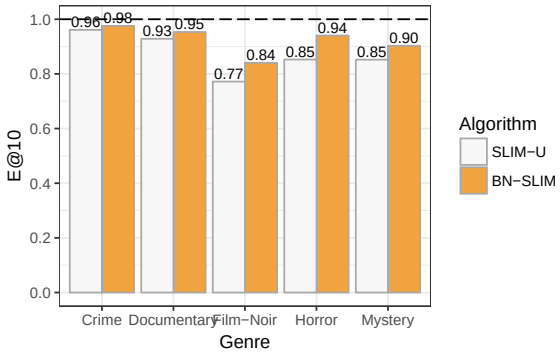


Fig. 2. Equity score for SLIM-U and BN-SLIM-U. Line indicates equal percentage across genders

Algorithm	NDCG@10
SLIM-U	0.053
BN-SLIM	0.052

Table 2. Ranking accuracy

Because the balanced neighborhood algorithm is applied across all users, it also has the effect of showing male users movie genres that occur more frequently for female users. To see this effect, we examined the five genres with the highest E_c@10 values: "Fantasy", "Animation", "War", "Romance", and "Western" using the same parameter values as above. The results appear in Figure 3 and show a similar result. "War" is something of an anomaly here, both because it is perhaps unexpected to see it as a one of the more female-recommended genres and because the genre-balance algorithm pushes it to become more skewed rather than less. We are investigating the cause of this phenomenon. Overall, the BN-SLIM-U algorithm produces a recommendation experience in which the occurrence of gender-specific genres is more closely equalized, with small loss in ranking accuracy.

Provider fairness: Kiva.org. Our dataset was extracted from Kiva's public API in September of 2016 and contains approximately 1 million loans funded by approximately 180,000 lenders. One challenge for collaborative recommendation in the microlending area is that loans are generally one-time endeavors. Unlike a movie that can be watched by an unrestricted number of viewers, a loan – once funded – disappears from Kiva.org and is not available for other lenders to view or support. Most loans are supported by from 1-330 lenders, by contrast, a popular movie in the MovieLens dataset might be rated by thousands of users. Thus, the lender-borrower relation is highly sparse, and loans have very small profiles.

To be able to apply the SLIM algorithm, we used a hybrid recommendation technique incorporating content data in the form of loan characteristics. We characterized each loan using five characteristics available from Kiva: borrower gender, borrower country, loan sector, loan purpose, and loan amount. Each of the original 1 million loan identifiers in the database was replaced with a psuedo-item identifier corresponding to the appropriate combination of loan characteristics. A 5-core transformation was then applied to the dataset, retaining only those users who had funded at least 5 psuedo-items and those psuedo-items with at least 5 funders. The retained dataset has 3,593 psuedo-items, 29,342 users and 393,035 ratings.

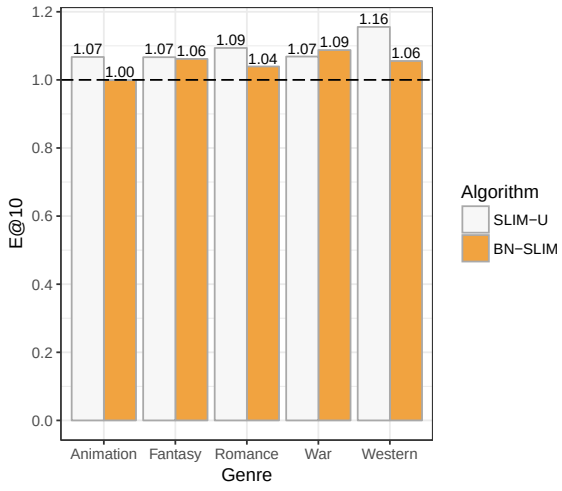


Fig. 3. Equity scores for female-preferred genres

Algorithm	NDCG@10	E_P@10
SLIM	0.046	0.90
BN-SLIM	0.049	1.05

Table 3. Comparison of algorithm performance

Kiva.org divides its borrowers into 9 geographic regions. As discussed above, for the purposes of this paper we are defining the protected group as those regions of the world where it appears to be more difficult to fund loans. (In Kiva.org, a loan that does not attract enough lenders over a 30 day period is marked as unfunded and dropped from the system.) As shown in Table 1, the regions of North America, Eastern Europe, South America, and Asia have proportionately more funded loans than the regions of Africa, Middle East, and Central America[4]. These regions where borrowers have lower funding percentages are treated as the protected group in our experiments.

With this transformation in place, it was possible to apply the SLIM algorithm and generate personalized recommendations. The regularization parameters were set as follows: $\lambda_1 = 0.01$ and $\lambda_2 = 0.001$. For BN-SLIM, λ_3 had a value of 0.9. Table 3 shows the performance of the these algorithms in the provider fairness condition. Interestingly, the ranking accuracy, as measured by NDCG@10, actually increases between the conditions, indicating that the balanced neighborhood condition actually yields better recommendation lists than the unmodified SLIM algorithm. In addition, the E_P@10 value, which is unbalanced at 0.90 for SLIM is improved to close to 1.0, the equity target that we were aiming for.

2.1.4 ***Conclusion and Future Work.*** Our BN-SLIM algorithm can be seen as an approach to building systems that target particular diversity-aware recommendation problems, where the providers and/or items can be divided into two disjoint categories. However, the approach is particularly suited to fairness-aware contexts because the objective function is optimized precisely when the protected and unprotected groups are weighted the same by the algorithm.

[4]Our data set had only a single loan request from Australia.

The most obvious precursor for this research is the work of Dwork et al. in the area of fair representation [46, 146]. The authors propose learning a mapping between the individual instances in the data to prototype instances with balanced membership such that protected group identities are not recoverable.

This paper extends this idea of fairness in classification to personalized recommendation. However, our application of this concept is different in that we are building on the standard nearest neighbor techniques in recommender systems and building balanced neighborhoods to ensure diversity among the peers from whom recommendations are generated.

A key aspect of this extension is to note the tension between a personalized view of recommendation delivery and a regulatory view that values particular outcomes. The regulatory view is somewhat foreign to research in personalization, but there are strong arguments that total obedience to user preference is not always risk-free or desirable [106, 130]. This paper also introduces the concept of multisided fairness, relevant in multisided platforms that serve a matchmaking function. We identify consumer- and provider- fairness as properties desirable in certain applications and demonstrate that the concept of balanced neighborhoods in conjunction with the well-known sparse linear method can be used to balance personalization with fairness considerations.

In our future work, we plan to extend these findings in several ways. It is possible that a multisided platform may require fairness be considered for both consumers and providers at the same time: a CP-fairness condition. For example, a rental property recommender may treat minority applicants as a protected class and wish to ensure that they are recommended properties similar to unprotected renters. At the same time, the recommender may wish to treat minority landlords as a protected class and ensure that highly-qualified tenants are referred to them at the same rate as to landlords who are not in the protected class. One important question for future research is how the outcomes for each stakeholder and the overall system performance are affected by combining consumer- and provider-fairness concerns. Another path to pursue is to have a more extensive experimentation of the fairness properties of the balanced neighborhood SLIM for both consumers and providers. We would like to test this idea on K-nearest neighbor method as well. Finally, we expect to publish a journal article of these thorough experiments in the Information and Management Journal.

3 RE-RANKING

In this section, we focus on achieving provider fairness using a re-ranking approach.

The problem of promoting provider fairness while maintaining recommendation accuracy can be generally characterized as a multi-objective optimization problem. If optimal fairness and optimal recommendation accuracy could be achieved simultaneously, there would be no need for research in this area. However, optimizing recommendation accuracy often comes at the expense of provider fairness, due to various biases present in recommender systems, including popularity bias [34, 90], and user-base composition [93, 140]. Research in provider fairness is therefore generally concerned with improving the tradeoff between fairness and accuracy, or in other words, increasing the amount of fairness that can be gained for a given degree of accuracy loss.

We motivate the problem in the context of loan recommendation where consumers are lenders and providers are borrowers. We propose two reranking methods: (1) Fairness-Aware Re-ranking (PFAR) and the personalized version of PFAR, and (2) Opportunistic Fairness-Aware Re-ranking (OFAiR).

The re-ranking criterion can be regarded as modelling *personalization* and *fairness*, respectively, with a hyper-parameter λ controlling the tradeoff between the two. We demonstrate that both methods achieve reasonable fairness / accuracy trade-offs and increase the exposure of the protected group(s) drastically. Although OFAiR achieves a better fairness / accuracy tradeoff compared to FAR/PFAR.

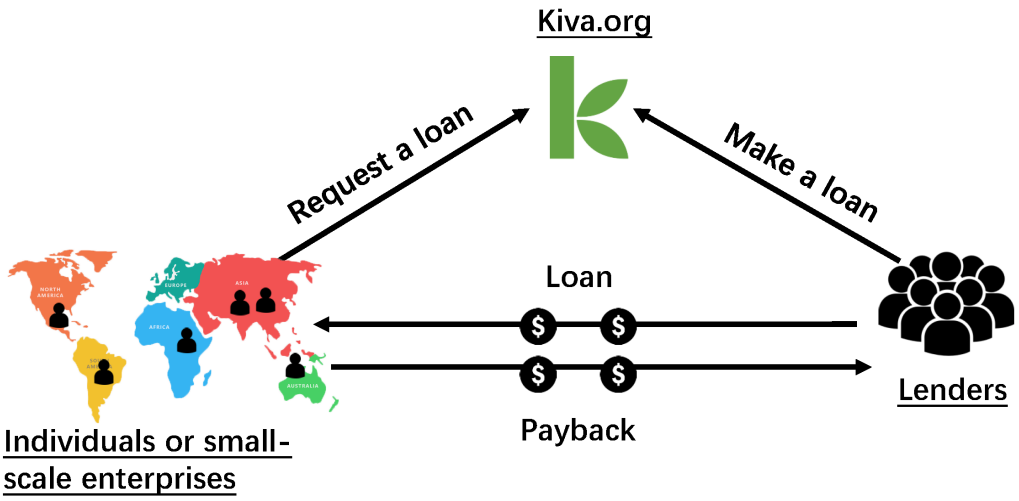


Fig. 4. Kiva.org provides an intermediary service for lenders and borrowers.

3.1 Fair-Aware Re-ranking/Personalized Fair-Aware Re-ranking

Microlending is the provision of small and low-interest loans (as little as $25) to low-income individuals or small-scale entrepreneurs from under-developed countries [143]. The extremely poor people in rural areas often lack collateral, steady employment, or a verifiable credit history, hence they cannot get access to financial services. Under such circumstances, microlending has attracted an increased attention in the last decade [38], providing impoverished entrepreneurs an opportunity to start their own businesses as well as avoid the vicious cycle of debt.

One of the leading international microlending organizations, Kiva Microfunds (Kiva.org), has crowd-funded about 3 million borrowers with $1.27 billion USD as of February 2019 [84]. Kiva does not collect interest but provides an intermediary service for lenders and borrowers, as illustrated in Figure 4. Borrowers from over 80 countries, divided into 8 regions by Kiva, post their applications for loans on the website for lenders to support. Lenders browse and crowd-fund the loans in the increments of $25 or more.

Loan recommender systems [39, 40] are designed to assist lenders in looking for promising borrowers. Such systems model lenders' historical behaviors and generate personalized recommendations to meet the lenders' interests or needs. While these recommender systems aim to provide efficient and personalized services, two issues are largely overlooked.

(i) *Unfair recommendation.* The existing recommender systems for microlending are all lender-centered. These recommender systems have been demonstrated to favor popular items [34, 90], resulting in extremely unbalanced recommendation results — majority groups are usually over-represented, thereby holding a higher proportion of opportunities and resources, while minority groups barely receive exposure. For example, it is observed that certain geographical regions such as Asia and Africa dominate the recommendation in a designed loan recommender system for Kiva [40]; whereas, others like Oceania and Eastern Europe barely receive recommendations, as shown in Figure 5. Less exposure means the borrowers from these regions are less likely to be funded.

(ii) *Lender's diversity tolerance.* Another noticeable phenomenon is that lenders' tolerance of diversity varies greatly. Thus, the diversity of recommended lists should be compatible with the level of each lender's interest in diverse recommendations. For instance, some lenders may highly prefer offering loans to certain regions (such as their home

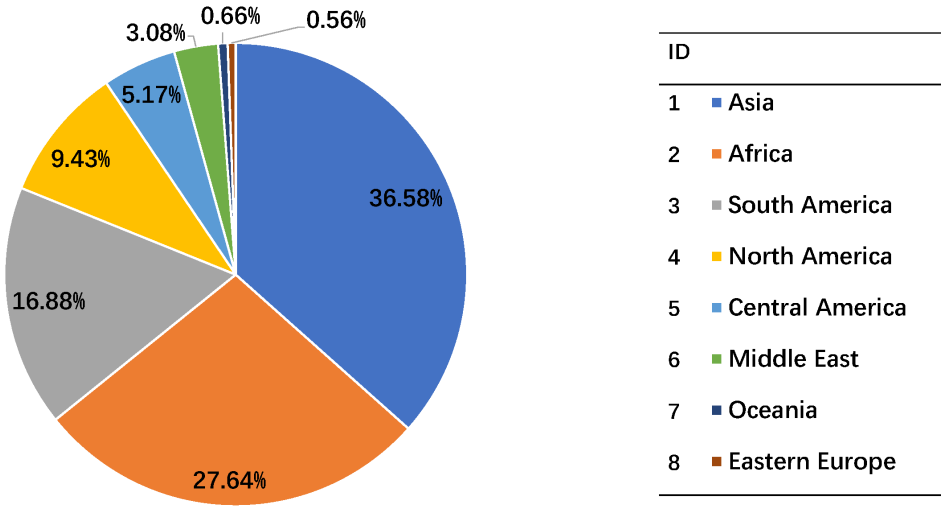


Fig. 5. The issue of fairness on regions in a designed loan recommender system [40] for Kiva: the recommendation percentage for each region.

countries), while others may be open to diverse regions. In such scenarios, assuming lenders' diversity tolerance is constant and increasing diversity uniformly for all lenders will result in poor recommendations [53]. Therefore, a well-designed recommender system should be personalized by both lenders' interests and the degree of loan diversity in recommended lists.

We aim to design a fairness-aware re-ranking algorithm on top of the existing recommendation algorithms. Our algorithm achieves a balance between recommendation accuracy and borrower-side fairness, and also considers lenders' preferences for diversity. This post-processing step does not depend on any specific recommendation algorithm, and therefore can be widely applied.

In this section, we first propose to formulate this recommendation scenario as a Multi-sided Recommender System (MRS) [25, 26]. Then, we design a personalized re-ranking algorithm to achieve a fair recommendation for microlending.

3.1.1 ***Problem Formulation***.

Based on the MRS model, different stakeholders are involved in the loan recommendation setting including lenders, borrowers, and the system (Kiva) where each stakeholder might have a different goal or a different fairness concern. In this system, borrowers post their loan requests on Kiva.org and lenders receive loan recommendations from the system, as depicted in Figure **??**. Besides maximizing the lenders' interests (personalization), Kiva also aims to consider the allocation of borrower side recommendation opportunities. In other words, Kiva wants to ensure that every loan request (borrower) gets an equal chance of being funded.

To achieve the above goal, given a set of lenders $\mathcal{U} = \{1, \ldots, n_u\}$, a set of loans $\mathcal{V} = \{1, \ldots, n_v\}$, and an initial ranking list $R(u)$ for lender $u \in \mathcal{U}$, our task is to re-rank $R(u)$ and generate a list of K distinct loans $S(u)$ that is both accurate and fair. For a loan $v \in \mathcal{V}$, $C(v) \in \{1, \ldots, n_c\}$ is the corresponding categorical protected attribute, such as region, race, or gender. Let $\mathcal{V}_c = \{v|C(v) = c, v \in \mathcal{V}\}$ denote the *group* of loans with attribute c. For instance, if the

protected attribute is the geographical region and we use the ID specified in Figure 5, then $\mathcal{V}_c$ with $c = 1$ represents the set of all loans applied from Asia.

In some contexts, we may separate borrowers into protected and unprotected groups, where the borrowers from the protected group are the particular concern [155]. This can be viewed as a special case of our problem with $n_c = 2$. In this research, we are looking at more general fairness across all borrower groups and trying to ensure that each re-ranked list $S(u)$ for a specific lender u covers as many borrower groups as possible while considering the personalized constraints for lenders in terms of their willingness to receive a diverse result. However, we are not allowed to sacrifice too much accuracy to achieve an absolutely fair recommendation result. A tradeoff must be made since accuracy and fairness cannot be fully satisfied at the same time [30].

3.1.2 ***Algorithm***.

We aim at producing a fairness-aware re-ranking algorithm that can balance personalization and fairness. We assume for a given lender u, a ranked recommendation list $R(u)$ has already been generated by a base recommender, *e.g.,* collaborative filtering. The task of our algorithm is to produce a new re-ranked list $S(u)$ containing loans that can satisfy lenders' demands and simultaneously cover as many borrower groups as possible.

We first propose a fairness-aware re-ranking algorithm (FAR) and then incorporate a diversity tolerance term τ_u to produce the personalized fairness-aware re-ranking algorithm (PFAR).

Fairness-aware Re-ranking (FAR). Our proposed fairness-aware re-ranking (FAR) criterion is defined as Eq.(11), which is the combination of a personalization-induced term and a fairness-induced term, with a hyper-parameter $\lambda \in (0, 1)$ controlling the tradeoff between the two. For any $u \in \mathcal{U}$, we solve

$$\max_{v \in R(u)} \underbrace{(1-\lambda)P(v|u)}_{\text{personalization}} + \underbrace{\lambda \sum_{c} P(\mathcal{V}_c) \mathbb{1}_{\{v \in \mathcal{V}_c\}} \prod_{i \in S(u)} \mathbb{1}_{\{i \notin \mathcal{V}_c\}}}_{\text{fairness}}, \tag{11}$$

where $P(v|u)$ is the personalization score determined by the base recommender, indicating the probability of lender u being interested in loan v. The indicator function $\mathbb{1}_A$ has the value 1 if A is true, and 0 otherwise. The new output list is built iteratively in a greedy manner. At each step, the algorithm selects one loan with the highest re-ranking score from the candidate list $R(u)$ and moves it to the output list $S(u)$.

For borrower-side fairness, our idea is to promote the loans that belong to currently uncovered borrower groups. For a loan v that belongs to $\mathcal{V}_c$, we first compute the coverage of $\mathcal{V}_c$ for the current generated re-ranked list $S(u)$ as $\prod_{i \in S(u)} \mathbb{1}_{\{i \notin \mathcal{V}_c\}}$, which is equal to 1 if none of the items in $S(u)$ belong to $\mathcal{V}_c$, and 0 otherwise. If both $\mathbb{1}_{\{v \in \mathcal{V}_c\}}$ and $\prod_{i \in S(u)} \mathbb{1}_{\{i \notin \mathcal{V}_c\}}$ are 1, items that belong to $\mathcal{V}_c$ are promoted by being assigned a higher score, and thus get a larger chance of being selected. The above process is repeated for each borrower group $\mathcal{V}_c$, $c = 1, \ldots, n_c$, and the results are summed up. Since each item may belong to multiple groups, the loans belonging to multiple uncovered borrower groups are favored.

The normalization term $P(\mathcal{V}_c)$ is determined by the system and indicates the importance of $\mathcal{V}_c$. For example, if a borrower group is identified as a protected group and receives few recommendations, then the system can assign a higher $P(\mathcal{V}_c)$ to the corresponding group. For simplicity, we assume a uniform preference over borrower groups and assign an equal $P(\mathcal{V}_c)$ for all borrower groups.

Algorithm 1 (Personalized) Fairness-Aware Re-ranking (FAR/PFAR)

Input: $u, R(u), K, \lambda, \tau_u$
Output: $S(u)$
1: $S(u) \leftarrow \emptyset$
2: **while** $|S(u)| < K$ **do**
3: Select the optimal v^* by solving

$$\arg\max_{v \in R(u)} (1-\lambda)P(v|u) + \lambda\tau_u \sum_c P(\mathcal{V}_c)\mathbb{1}_{\{v \in \mathcal{V}_c\}} \prod_{i \in S(u)} \mathbb{1}_{\{i \notin \mathcal{V}_c\}}$$

4: $R(u) \leftarrow R(u) \setminus \{v^*\}$
5: $S(u) \leftarrow S(u) \cup \{v^*\}$
6: **end while**
7: **return** $S(u)$

Personalized Fairness-aware Re-ranking (PFAR). Note that Eq.(11) is designed for any lender $u \in \mathcal{U}$. If we simply follow this setting and treat each lender with an equal level of diversity tolerance, then the ranking quality is inevitably suppressed. Actually, lenders' propensity towards diversity varies and different levels of diversity should be considered.

To address this issue, we personalize the previous re-ranking criterion Eq.(11) by adding a personalized weight τ_u and derive our personalized fairness-aware re-ranking (PFAR) criterion Eq.(12). For any $u \in \mathcal{U}$, we solve

$$\max_{v \in R(u)} \underbrace{(1-\lambda)P(v|u)}_{\text{personalization}} + \underbrace{\lambda\tau_u \sum_c P(\mathcal{V}_c)\mathbb{1}_{\{v \in \mathcal{V}_c\}} \prod_{i \in S(u)} \mathbb{1}_{\{i \notin \mathcal{V}_c\}}}_{\text{personalized fairness}}, \tag{12}$$

where the second term considers personalized fairness. The diversity tolerance τ_u is incorporated to control the weight of the fairness score. If a lender has no special interest to a specific group, the algorithm will focus more on the personalization task. Otherwise, the fairness-induced term can be emphasized.

To calculate τ_u, we first compute a level of interest $P(\mathcal{V}_c|u)$ of lender u for each borrower group $\mathcal{V}_c$, $c = 1, \ldots, n_c$,

$$P(\mathcal{V}_c|u) \triangleq \frac{\sum_v r(u,v)\mathbb{1}_{\{v \in \mathcal{V}_c\}}}{\sum_{c'} \sum_v r(u,v)\mathbb{1}_{\{v \in \mathcal{V}_{c'}\}}}, \tag{13}$$

where $r(u,v)$ is the rating from lender u to loan v. We compute the ratio of summation over rated borrowers that belong to group c over summation over all the rated borrowers.

The preference $P(\mathcal{V}_c|u) \in [0,1]$ indicates the lender's taste over borrower groups where $\sum_c P(\mathcal{V}_c|u) = 1$. Some lenders may be highly interested in certain borrower groups, while some lenders may have equal preferences over all the borrower groups. To capture this characteristic, we use the information entropy [123] to identify the lender diversity tolerance, namely

$$\tau_u \triangleq -\sum_c P(\mathcal{V}_c|u) \log P(\mathcal{V}_c|u), \tag{14}$$

where a larger τ_u means that the lender is more open to a diverse set of borrower groups.

The algorithm FAR/PFAR is formally given in Algorithm 1. For a lender u, loans are generated iteratively from the initial ranking list. The loan with the highest score is selected from the candidate list $R(u)$ according to our re-ranking criterion Eq.(11) or Eq.(12). The process is repeated until $S(u)$ has reached the desired length. The proposed algorithm

automatically balances personalization and fairness by adding a bonus to the loans that belong to the uncovered borrower groups. The generated re-ranked list for each lender tends to cover each borrower group at least once while encouraging personalization.

3.1.3 ***Experiments***.

In this section, we test our proposed algorithms on a real-world dataset from Kiva.org. The performance of our proposed re-ranking algorithms on top of different base recommenders is evaluated in terms of accuracy and fairness. Our implementation is built upon the LibRec 2.0 [62], and all results are averages from five-fold cross-validation.

Dataset. Our algorithms are evaluated on a proprietary dataset obtained from Kiva.org, including all lending transactions over an 8-month period. Each loan is specified by features including borrower's name, gender, borrower's country, loan purpose, funded date, posted date, loan amount, loan sector, and geographical coordinates.

One important characteristic of this dataset, and the micro-finance domain in general, is there is rapid turn-over in recommendable items. Loans are only available to lenders for a short period until they are fully funded or dropped from the system. Subsequent visitors will not see or be able to support these loans, limiting the maximum item profile size significantly. For example, at a minimum loan amount of $25, a $200 loan can have a maximum of only 8 lenders. Contrasting with a consumer taste domain such as MovieLens [65], where a popular movie might be rated by hundreds or thousands of consumers, the Kiva.org dataset is extremely sparse (with a sparsity of 4.19×10^{-5}) and exhibits a significant item cold-start problem.

To generate a denser dataset with greater potential for user profile overlap, we apply a content-based technique [113], creating *pseudo-items* that represent large categories of items. In particular, all loans that share the same borrower gender, borrower country, loan purpose, loan amount (binned to 5 equal-sized buckets), and loan sector are combined into a single pseudo-item. Then we apply a 10-core transformation, selecting pseudo-items with at least 10 lenders who had funded at least 10 pseudo-items. The retained dataset has 11,085 pseudo-items, 9,597 lenders and 204,830 ratings.

Comparative Recommenders. As of this writing, Kiva.org does not offer recommendation functionality. In our experiments, we assume a context in which the site provides short lists of recommended loans to lenders for their review. We set the protected attribute as the geographical region, because part of Kiva.org's mission is to achieve equitable access to capital across regions. In order to set up the recommendation scenario for Kiva, we select four representative base recommenders to study their performance in accuracy and fairness, as well as how our proposed algorithms can influence the recommendation results: (a) RankSGD [70] uses stochastic gradient descent to optimize the ranking error; (b) UserKNN [113] is a memory-based collaborative algorithm that computes user similarity; (c) Weighted Regularized Matrix Factorization (WRMF) [69] creates a reduced-dimensionality factorization of the rating matrix; (d) Maximum-entropy distribution (Maxent) [40] is a loan recommender system specially designed for Kiva. Maxent models lending behaviors by estimating a maximum-entropy distribution based on a set of heterogeneous information regarding micro-financial transactions available at Kiva.

Evaluation Metrics. We propose to utilize Normalized Discounted Cumulative Gain (nDCG) [71] and Average Coverage Rate (ACR) to evaluate recommendation accuracy and borrower-side fairness, respectively. ACR is defined by the average number of borrower groups covered by the ranked list,

$$\text{ACR} = \frac{\sum_{u \in U_t} N_{S(u)}}{N_{\text{bg}} |U_t|}, \tag{15}$$

where U_t is the test lender set, $|U_t|$ is the number of lenders in the test set, N_{bg} is the total number of borrower groups and $N_{S(u)}$ is the number of borrower groups covered in the list $S(u)$. A larger ACR indicates a fairer system regarding borrower-side fairness.

We propose to use Normalized Discounted Cumulative Gain (nDCG) to evaluate the ranking accuracy and borrower-side fairness, respectively.

Discounted Proportional Fairness (DPF) We adopt a well-accepted and axiomatically justified metric of fairness, the proportional fairness [83]. Proportional fairness is a generalized Nash solution for multiple groups.

$$DPF = \sum_{i=1}^{n_c} \log\left(\frac{x_c}{\sum_{c'} x_{c'}}\right),$$

where x_c is the allocation utility of group $\mathcal{V}_c$. We define the utility of $\mathcal{V}_c$ by the cumulative gain that $\mathcal{V}_c$ received from all users,

$$x_c = \sum_{u \in \mathcal{U}} \sum_{c} \sum_{i=1}^{K} \frac{rel_i \mathbb{1}_{\{v \in \mathcal{V}_c\}}}{\log(i+1)}$$

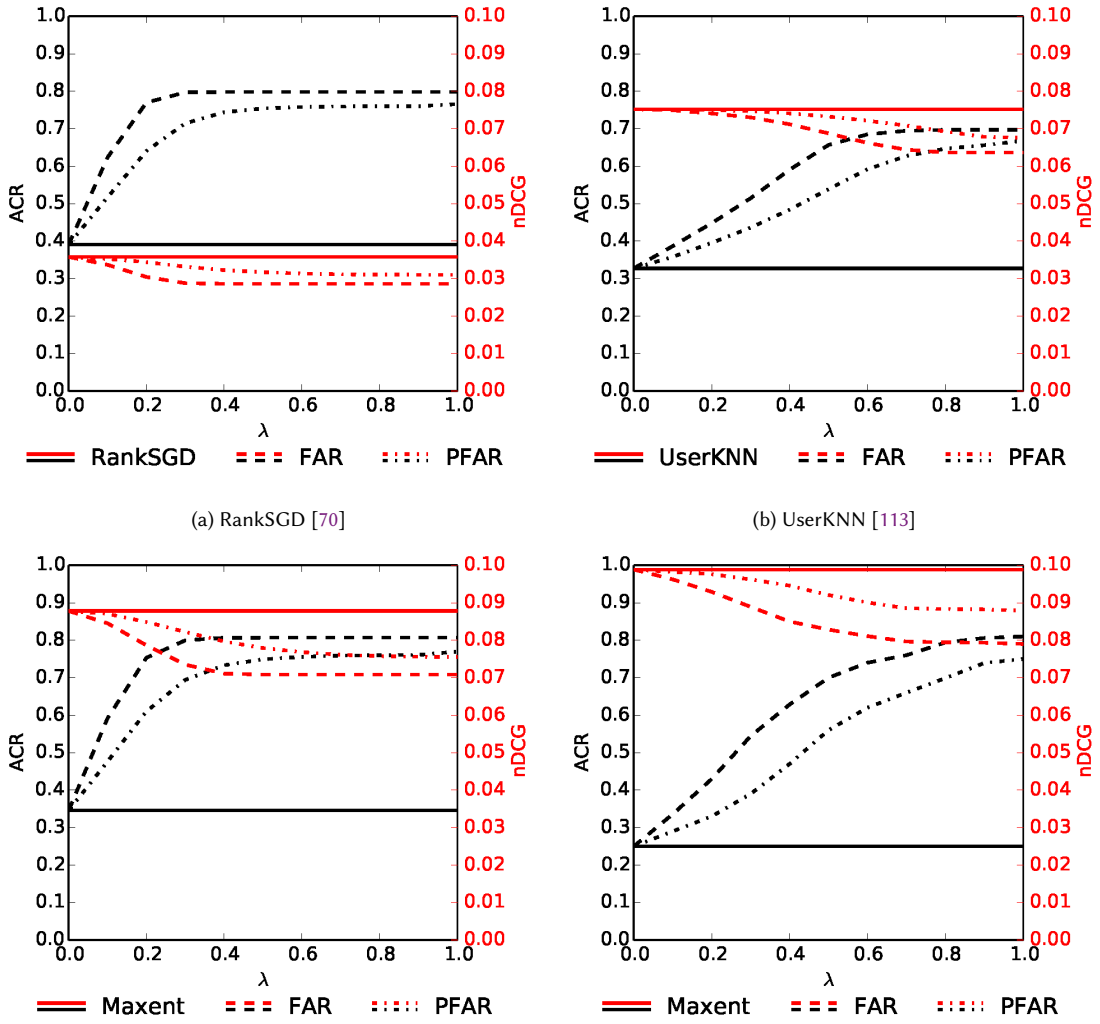
Since fairness-aware re-ranking aims to achieve a better tradeoff between fairness and accuracy, we expect that there will be a price to pay for obtaining a fairer system. To study the ACR gain under a certain accuracy budget, we calculate ACR@NDCG$_{5\%}$, the increased ACR value obtained when we allow a 5% decrease in nDCG. We believe that 5% nDCG loss may be a reasonable accuracy tradeoff if provider fairness can be enhanced. Indeed, it has been shown that user inconsistency in recommender systems generates a lower bound (the "magic barrier") within which accuracy measurements are meaningless, so small relaxations of nDCG may not actually represent a loss in performance [119].

Results and Analysis. We first study the performance of applying FAR and PFAR to the base recommenders. We vary the hyper-parameter λ from 0 to 1 in steps of 0.1 and record the corresponding ACR and nDCG, where a larger λ means the weight of fairness is larger. The results are shown in Figure 6.

Considering the performance of all base recommenders (the solid lines), rankSGD has the lowest accuracy (nDCG = 0.0358); UserKNN comes next with nDCG=0.0752 by finding the nearest neighbors; WRMF performs better than UserKNN by learning the latent factors of lenders and loans (nDCG=0.0878); Maxent obtains the highest nDCG of 0.0988 since Maxent is specially designed for loan recommendation and additional features of loans, *e.g.,* loan sector and geographical coordinates, are utilized. However, accurate recommenders tend to favor items from certain groups, thus resulting in fairness issues.

Effectiveness of the re-ranking algorithms. By applying our proposed algorithms (when $\lambda \in (0, 1)$), all recommenders tend to achieve fairer recommendation results by promoting the loans that belong to less-popular groups, showing the flexibility and effectiveness of our proposed algorithms. Our re-ranking algorithms can be deployed to any base recommender, with the weight of the fairness tunable. The accuracy slightly decreases as λ increases, as there is a price to pay for obtaining a fairer system. Take $\lambda = 0.1$ for instance, Maxent can obtain a gain of 25.1% in ACR with a loss of merely 1.6% in nDCG. Moreover, RankSGD and WRMF converge faster than UserKNN and Maxent with the increase of λ, indicating that the behavior of our proposed algorithms depends on the initial ranking list to some extent.

Note that RankSGD has limited ability in learning lenders' preferences, while the results are usually fairer. This is to be expected since accuracy and fairness are conflicting.



(a) RankSGD [70]

(b) UserKNN [113]

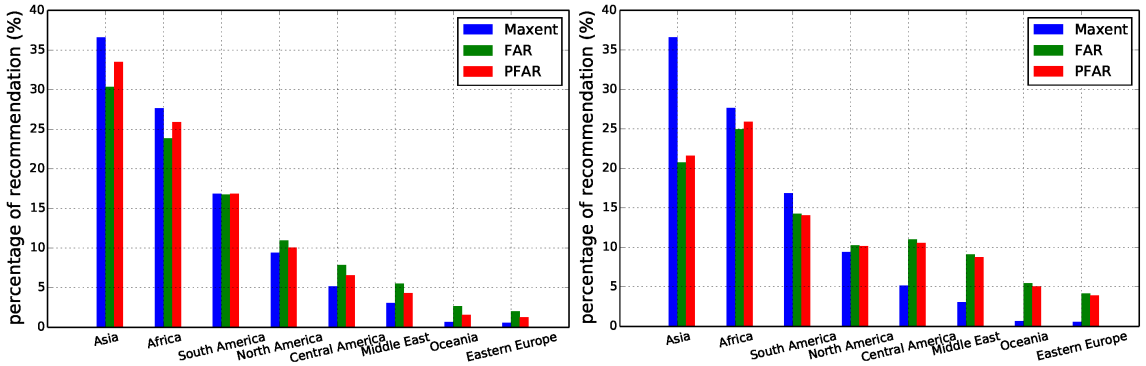
(c) WRMF [69]

(d) Maxent [40]

Fig. 6. Tendencies of ACR and nDCG with increasing λ.

Comparison between FAR and PFAR.. The recommendation accuracy of PFAR is higher than FAR, since PFAR limits the amount of loan diversity that the re-ranking imposes, based on the individual tolerance. We can also observe from Figure 6 that fairness of PFAR is lower, which is consistent with our previous discussion and demonstrates the tradeoff between accuracy and fairness.

Visualization of the re-ranking results. We compute the percentage of recommendations for each group with and without the proposed re-ranking algorithms, and study the corresponding allocation distribution. Due to the 4-page



(a) $\lambda = 0.1$, $\text{nDCG}_{\text{FAR}} = 0.0962$, $\text{nDCG}_{\text{PFAR}} = 0.0983$. (b) $\lambda = 0.99$, $\text{nDCG}_{\text{FAR}} = 0.0709$, $\text{nDCG}_{\text{PFAR}} = 0.0756$.

Fig. 7. Recommendation percentage of each region. The blue bars show the results of the base recommender ($\lambda = 0$). The green and red bars represent the results of FAR and PFAR, respectively.

limitation, we chose Maxent as an example, and the results are shown in Figure 7. Similar trends can be observed for other base recommenders.

(i) The blue bars show the distribution of the base recommender Maxent, *i.e.*, $\lambda = 0$. We observe that Maxent focuses on a few major borrower groups, namely Asia and Africa (making up 36.58% and 27.64% of total recommendations, respectively), while paying less attention to the others.

(ii) The ideal fairness is to give each group an equal chance of being recommended. However, accuracy will be significantly downgraded as lenders' preferences are not learned. As a compromise, we find a balance between the two and select $\lambda = 0.1$, where the growth rate of ACR per unit nDCG loss is the largest. As illustrated by the green and red bars in Figure 7a, nDCG still remains at a high level after the re-ranking (nDCG=0.0962 for FAR and nDCG=0.0983 for PFAR), while fairness of the recommendation is significantly improved, as loans belonging to less-popular groups are promoted.

(iii) In Figure 7b, a larger $\lambda = 0.99$ is applied, and the distribution can be even more balanced, while the accuracy is lower.

3.1.4 ***Conclusion and Future Work.*** In this work, we proposed a personalized fairness-aware re-ranking algorithm for microlending that can balance accuracy and fairness. We increase the coverage rate of borrowers' regions for Kiva.org to achieve borrower-side fairness, and we show that our algorithm can do so with minimal loss in ranking accuracy. In addition, our algorithm includes lender-specific weights that can be used to personalize the degree of loan diversity.

We note that, in real-world recommendation applications, managing the tradeoff between accuracy and coverage of provider groups is not a single-shot process. Rather it is an online process, where a current lack of coverage can be compensated for at a later time, and where results are evaluated temporally. This would require making the algorithm sensitive to historical patterns of coverage, rather than just the results obtained in the current list. I intend to explore this type of algorithm design and evaluation in our future work.

3.2 Opportunistic Multi-aspect Fairness through Personalized Re-ranking

In this section we focus on the *provider fairness* again where it concerns the impact of recommendation delivery on the providers of items being recommended and the questions of fair treatment that may arise[25].

Recent research has sought to alleviate this concern, including the re-ranking method in the previous section, using a variety of approaches. See, for example, [14, 28, 51, 77, 96, 140]. What all of these approaches share is that they focus on a *single dimension* over which fairness is sought: a single protected group among the providers, and except for [96], they do not take user preferences in item features into account.

Our work in this paper extends the FAR/PFAR approach discussed in the previous section which was seeking to improve the accuracy / fairness tradeoff through increased *personalization*. Namely, can we tailor the type and degree of optimization specific to each user's tastes and preferences and therefore improve accuracy? We label this approach *opportunistic* because we view each user as presenting a particular type of opportunity to increase recommendation fairness and try to make the most of each. In particular, we seek to identify the particular dimensions along which a user might be open to result diversification that improves fairness and thereby enable multiple fairness concerns to be addressed at once.

$user_1$	F_1:Region	F_2:Gender	F_3:Sector	F_4:Amount
$item_1$	Africa	Female	Agriculture	\$0-\$500
$item_2$	Africa	Female	Health	\$0-\$500
$item_3$	Africa	Female	Clothing	\$0-\$500

Table 4. Profile of $user_1$

As an example, in the context of loan recommendation, suppose user u prefers to lend her money to women in Kenya but she does not have a strong preference for a loan's purpose or economic sector. This user's profile might appear as in Table 8. While the user might not respond well to loans in other countries, we can consider her open-mindedness regarding the Sector feature as an opportunity to increase fairness in this area. For the sake of example, assume loans from the Education and Conflict Zones sectors are historically underfunded in Kenya, so the loans in these sectors are identified as protected. Consider the recommendation results in Table 5. The first two recommendations ($r_1 and r_2$) increase fairness across only the Sector feature by promoting items from underfunded sectors while honoring the user's preference to lend money to Kenyan women. On the other hand, loan r_3 might not be an effective recommendation for this user since it diversifies on the wrong dimensions, although it might still be promoting protected items. In other words, we want to promote fairness concerns when the user's profile indicates receptivity and be cautious otherwise.

$user_1$	F_1:Region	F_2:Gender	F_3:Sector	F_4:Amount
r_1	Africa	Female	Conflict Zones	\$0-\$500
r_2	Africa	Female	Education	\$0-\$500
r_3	Asia	Male	Livestock	\$500-\$700

Table 5. Recommendations for $user_1$

This paper addresses the following research questions:

RQ1: Do users exhibit different patterns of preference across fairness dimensions?

RQ2: Can these patterns be exploited to improve the recommendation fairness / accuracy tradeoff using re-ranking?

3.2.1 ***Background.*** This line of research has much in common with work that seeks to enhance diversity in recommendation [32, 53, 137, 148]. However, the key differences have to do with the concerns being addressed and, accordingly, the way in which success is measured. Usually when diversity is invoked as a desirable property of a recommender system, it is in the service of some user-oriented goal. Diverse recommendations can help a system cope with a diverse range of user intents and contexts. For example, a restaurant recommender might know that a user sometimes goes to family-style pizzerias 70% of the time and fancy French restaurants 30% of the time. Rather than present just pizzerias in a recommendation list, even though that is likely to be the right answer statistically, it might be better to include one or two fine dining establishments on the list, just in case the user is looking for a "date night" recommendation this time around.

Typical measures of diversity such as intra-list distance, for example [153], therefore measure the difference among items in each user's list, without regard to what items they are. Diversity as a fairness concern seeks varied outputs for a completely different reason, namely to increase the prevalence of items from under-represented providers, and measures outcomes relatively to those providers specifically. We will distinguish between these sense of diversity by using the term *list diversity* to refer to the user-centered objective and *fairness-promoting diversity* to the provider-centered objective, our main concern in this paper.

Another related definition of diversity is what is called *aggregate diversity* or catalog coverage. The question here is whether the recommender is presenting all of the available items in the catalog. This can be seen as a minimal form of fairness where the frequency of appearance is not considered, just that an item is recommended at least once, and we do not differentiate between different items or different providers [5].

As noted above, most work in recommendation fairness, and machine learning fairness more generally, simplifies the problem of fairness-enhancement by concentrating on a single (usually binary) distinction between a protected group and an unprotected group. This is an excellent starting point and admits of tractable mathematical formulations. However, this approach is not a good match to real-world applications, where there are likely to be multiple fairness concerns related to multiple dimensions of identity [81].

From the user perspective, the effect of these considerations is more or less the same: the recommender produces recommendations that are more diverse than would be found based on personalization considerations alone. It is only in looking at the recommendations in aggregate and considering the system's objectives that different diversity methods can be readily distinguished.

3.2.2 ***Problem formulation.***

Given a set of users $\mathcal{U} = \{u_1, \ldots, u_n\}$, a set of items $\mathcal{V} = \{v_1, \ldots, v_m\}$, and initial ranking lists $R(u)$ for users $u \in \mathcal{U}$, our task is to re-rank $R(u)$ and generate a list of k distinct items $S(u)$ that is both accurate and fair similar to [96]'s goal.

We will further assume that each item $v_i \in \mathcal{V}$ is represented by a d-dimensional feature vector $\vec{\phi}_i = \langle f_{i1}, \ldots, f_{id} \rangle$ over a set of categorical features $F = \{F_1, F_2, \ldots, F_d\}$. Each dimension F_j can be viewed as a set of categorical values or labels and so for an item v_i, its feature vector ϕ_i contains $f_{ij} \in F_j$ for each feature F_j. We will use the notation $c_j = |F_j|$ to refer to the cardinality of the feature F_j.

As an example, suppose that our set of items are loans and users are our potential lenders. Suppose that each loan is characterized by two features: geographical region and economic sector. Thus, $F = \{\text{Region}, \text{Sector}\}$, and $d = 2$. Suppose that we have 5 geographical regions and 7 economic sectors. For example: Region $= F_1 = \{\text{Africa}, \text{Asia}, \text{Americas}, \ldots\}$ and Sector $= F_2 = \{\text{Agriculture}, \text{Housing}, \text{Education}, \text{ConflictZones}, \ldots\}$. If a particular loan v_i is sought in the agriculture sector in Africa, we would say $\vec{\phi}_i = \langle \text{Africa}, \text{Agriculture} \rangle = \langle f_{i1}, f_{i2} \rangle$.

A protected class, within some F_j feature, consists of a set of values $F'_j \subset F_j$ that are considered protected and for which fairness is sought. There may be multiple fairness dimensions of concern, we define the protected dimensions F' as the subset of F that contain such protected values. For example, if Education and Conflict Zone loans are relatively underfunded, then in the Sector feature, these two specific values form the protected group F'.

Personalized diversity. Studies have shown that users generally prefer more recommendation results they perceive as diverse [68]. This suggests that the opportunity for fairness-enhancing diversification exists and may come at minimal cost in terms of user experience. However, users differ in the variety that they seek in recommendations [135]. Some recommendation research has sought to capitalize on these differences in improving diversity [52]. Here we aim to do the same in a more fine-grained way, consider each user's interest in diversity across multiple features.

Figure 8 gives a schematic depiction of this distinction. In this example, each item has a color and a shape feature. A user profile, shown at the top, consists of squares of different colors. Clearly, this user has a strong interest in squares and cares less about what color they are. A recommender that prioritizes triangles and circles as a protected group as well as greenish/yellowish hues might deliver recommendations as shown in the second row. These will likely not be accepted as they deviate too much from the characteristics preferred by the user. A better approach would be to diversify only in (the dimensions/values of) color, retaining the aspect of the items that the user apparently prefers.

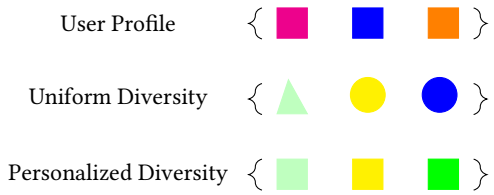


Fig. 8. Uniform vs Personalized Diversity

In the previous section, we introduced the concept of recommendation re-ranking using a quantity τ_u, a user-specific measure of interest in diversity, based on information entropy [95, 96]. Here we extend this definition to take into account multiple item features while seeking fairness within each feature's dimensions. Instead of a single user-specific τ_u, the $\vec{\tau}_u$ vector will represent the user's level of tolerance for diversity across the feature space (such as the user in the above example having more tolerance for diversity in the "color" feature and less in the "shape" feature). Specifically,

$$\vec{\tau}_u(F_j) \triangleq -\sum_{f \in F_j} P(f|u) \log P(f|u), \tag{16}$$

where $P(f|u)$ is computed as the fraction of items in the user's profile that have the feature value f. This can be interpreted as the user's likelihood of liking items with that value. The higher the entropy value is for a user on a feature, the higher her tolerance to see diversity within that feature. For example, the user in Table 8 would have low entropy for Region and Gender, but higher entropy for Sector.

This vector of values, therefore, quantifies the relative opportunities for providing diverse results to users. As we show in Section 5.4, these values vary widely across different features and different users, motivating a recommendation techniques that is sensitive to these individual differences.

Recommendation re-ranking. Re-ranking is a common technique for enhancing the non-accuracy properties of recommender systems output. It provides a relatively simple framework for augmenting an existing recommender

system with concerns that are not part of its design. Generally speaking, a re-ranker is a function that maps a ranked list $R(u)$ of size k (e.g., a ranked recommendation list) and produces a new list $S(u)$ of size k' where $k' <= k$ and where all items are drawn from the original list: $\forall i : i \in S(u)$ *iff* $i \in R(u)$. The loss of ranking accuracy in doing so is thereby limited by the size k; no item in $S(u)$ can be worse than what the original recommendation placed at rank k.

Re-ranking algorithms of this type were introduced in information retrieval for enhancing user-oriented diversity. The *Maximum Marginal Relevance* method as proposed in [32] measures for each user, the dissimilarity between a query and the items in her retrieved results. This method intends to combine query relevance and list diversity using a greedy list accumulation algorithm. The algorithm builds the output list S one item at a time.

At each point in time, it scores potential new items by a combination of their relevance (as computed in the initial retrieval step) and their differences from the current list (novelty), computed by identifying the item $j \in S$ that is most similar to the new item.

In our context, we will assume that we have some function *sim* that computes similarity between two items i, j and that our recommender system returns a relevance score of *rec(v,u)* for a user u and item v. We can then define the MMR-based scoring function:

$$OFAiR(u, v, R, S) \triangleq \arg\max_{v \in R \setminus S} [\lambda(rec(v, u) - (1 - \lambda) \sum_{v' \in S} sim(v, v')] \tag{17}$$

Effectively, the algorithm, at each point, finds the next item to include by incorporating the original ranking (as encapsulated in the recommendation score), but penalizes that score when the proposed item is highly similar to the items already added.

There is a subtle difference between the MMR formulation here and its original specification. When scoring a new item to decide whether to add it to the re-ranked list, MMR chooses the most similar item – this is the "marginal" part of the algorithm. Our formulation calculates the summation of similarities between the target item and all the other items in the re-ranked list. We can think of this as identifying the item with maximum aggregate difference from the existing list. We will explain later how this change is appropriate in a fairness context.

eXplicit Query Aspect Diversification method proposes another formulation to enhance diversity. Although, this method has a similar goal to MMR, it enhances diversity with respect to specific aspects of an item [120]. The diversity objective relative to a particular aspect (e.g., feature, topic, or category) is considered satisfied if one item containing that aspect is added to the result list. In context of recommendations, we can express this ranking score as follows:

$$xQuAD(u, v, R, S) \triangleq \arg\max_{v \in R \setminus S} [\lambda(rec(v, u) + (1 - \lambda) \max_{v' \in S} \mathbb{1}_{\vec{v} \cap \vec{v'} = \emptyset}], \tag{18}$$

where x_v represents the set of aspects present in item v. In effect, this algorithm boosts the rank of items that, when added to the list so far, bring in new aspects – features that have not yet appeared in the list.

In FAR/PFAR algorithms in the previous section [95, 96], we proposed two extensions to xQuAD. The first *FAR* (fairness-aware re-ranking) applied the formalism using aspects of an item defined over a fairness-relevant feature. In this configuration, the algorithm boosts the scores of items from protected groups when no such item has yet been added to the list. Once the group is represented, the boosting disappears. This can be seen as an implementation of the "Rooney rule" [85] that ensures minimum representation for protected groups. The second variant *PFAR* adds personalization to this process. Using the τ_u information entropy measure described above, the fairness-boosting term is

modulated so that users with more diverse profiles (who have a high diversity tolerance/higher entropy) are presented with results containing more fairness-enhancing diversity.

In particular, the scoring function of PFAR is composed of a personalization score $rec(v, u)$ and a personalized fairness score. PFAR simply assumes only one sensitive feature need to be considered. Suppose the given sensitive feature dimension is F_a, then the scoring function is defined by

$$PFAR(u, v, R, S) \triangleq \arg\max_{v \in R \setminus S} [\lambda rec(v, u) + (1 - \lambda)\tau_u \min_{v' \in S} \mathbb{1}_{v_a \neq v'_a}], \tag{19}$$

where v_a is the a-th element of the feature vector $\vec{v}$. Note that PFAR inherits the limitation of xQuAD that it assumes binary inclusion as a sufficient definition of fairness and it is therefore difficult to tune it to improve the representation of protected groups in a proportional way.

3.2.3 ***Opportunistic fairness***.

We are now ready to describe *OFAiR* (**O**pportunistic **F**airness-**A**ware **R**eranking), which incorporates personalization at the feature level into the re-ranking process and also allows fine-grained control of protected group promotion by using per-feature weights.

As discussed above, we can represent the variation in a user's profile across all features through the vector $\vec{\tau}_u$, calculated using information entropy. However, because these weights are feature-specific, we cannot incorporate them as a single multiplier as found in PFAR. Also, because we are interested in fine-grained control over the proportions of protected group items in recommendation lists, the xQuAD formula with its binary inclusion metric is not appropriate. So, our alternative in OFAiR applies the MMR approach by penalizing item similarity, but we build the feature significance into the similarity metric itself. We want to add items to the recommendation list if they add to the representation of protected groups in the recommendation list and if they differ from the items on the list in areas of high diversity tolerance for the user. To achieve this effect, we multiply together the user-specific tolerance weight for each feature and a weight associated with a feature's protected / unprotected class.

We use weighted cosine similarity to allow the similarity between two items to be controlled by weights associated with each dimension. Because the weights actually vary by value, not just by dimension, and we can only pass a single weight vector to the weighted cosine similarity function, we convert the feature vector $\vec{\phi}$ to a smoothed binary vector of dummy variables b_i with one dimension for each possible feature value. The smoothing operation means that instead of missing values being represented by zero, they have a small value $\epsilon = 2.2e^{-16}$. The user tolerance weights are correspondingly expanded in dimension to match: $\vec{\tau}_u \rightarrow \vec{\gamma}_u$.

Let $\vec{a} \circ \vec{b}$ represent the element-wise (Hadamard) product between two vectors a and b. Let $W(f')$ be a function that returns the weight of a particular binary feature value f'. This value will be small for unprotected values and larger for protected values as described below. For all items, we derive a weight vector $\vec{w}$ where the elements $w_j = W(f'_j)$. Let $\vec{z_u}$ be the product, which combines the two types of weights.

$$\vec{z}_u = \vec{\gamma}_u \circ W(F') \tag{20}$$

The entries z_{uj} represent the weight assigned to user u for the jth dummy (smoothed binary) feature, combining both individual diversity tolerance and the system's fairness objective.

The weighted cosine metric applies weights to the terms of the cosine computation:

$$wcos(\vec{b}, \vec{b}', z_u) \triangleq \sum_{j}^{|F|} z_{uj} b_j \times b'_j \frac{1}{\sqrt{\sum_j z_{uj} b_j^2} \times \sqrt{\sum_j z_{uj} b_j'^2}} \tag{21}$$

Two items are similar under this calculation if their values on many dimensions are the same and those dimensions are ones where the user profile has high entropy / variation and where their associated weight is high.

Recall that the similarity calculation in MMR is used to penalize items that would be redundant with what is already in the recommendation list. So, the higher the similarities are, the higher the penalty. Therefore, we will want a weighting scheme where protected items are weighted high: their similarity is more important to the system.

This weighting scheme interacts with our aggregate difference alteration of the MMR algorithm noted above. By definition, protected items will be a small subset of the recommended items. Therefore, protected items will always differ from the list in aggregate. Also, the features in the recommendation list are likely to reproduce the consistencies in the user profile that represent lower tolerance for diversity. Weighting the protected features more highly helps promote diversity on those dimensions while keeping the other dimensions less diverse.

Various schemes for the weighting function were considered in our experimentation. In this paper, we report on a simple scheme where protected features receive a fixed high weight α and unprotected features a fixed low weight $\alpha/100$. In our experiments, the results were not sensitive to the magnitude of these values as long as protected features have a lower weighting. Additional exploration of feature weighting will be considered in future work.

3.2.4 ***Experiments***.

Evaluation Metrics. The accuracy of the following methods was evaluated based on Precision, Recall, normalized discounted cumulative gain (nDCG), and to calculate their feature-based diversity both intra-list distance (ILD) and entropy of the recommendation lists were used. The fairness of lists was evaluated based on protected group exposure, which measures the fraction of the recommendation list that consists of protected group items. This value is related to the fairness concept of "statistical parity," measured relative to items' level of promotion within the recommender system. Because list lengths are fixed (10 in our case), the exposure of unprotected items is just one minus the protected group exposure.

Dataset. We test our model on two datasets. The first is The Movies Dataset, which was obtained from the Kaggle website and contains the metadata of 45,000 movies listed in the Full MovieLens Dataset [5] which were released on or before July 2017. Although movies are not a domain to which important fairness concerns are typically applied, we use this dataset as a well-known example with a rich set of provider-side features. The dataset contains 26 million ratings from 270,000 users for all 45,000 movies. Ratings are on a scale of 1-5. Each movie contains a set of features from which the following were used in this project: genres, original language, release date, revenue, run-time, popularity, production countries and spoken language. A sample of this dataset was extracted which contained the 559,070 ratings from 6,000 users on 14,623 items (density of 0.63%).

All the features were transformed into categorical variables. If the movie's popularity is greater than the average popularity, we tag the movie as popular and unpopular otherwise. We transform the revenue and run-time in the same way as well. The release date is bucketed into old and new if the movie's release date is before or after 1990 [77]. All the categorical features were transformed into dummy variables, resulting in a total of 323 binary features.

[5] https://grouplens.org/datasets/movielens

For the purposes of exposition, we selected two features in each dataset along which to identify protected features, although the OFAiR algorithm supports any number of sensitive features. In the Movies Dataset, we identified the following protected classes within each feature: "unpopular" (popularity), "lower revenue" (revenue) , "longer" (running time), "before 1990" (release date), some genres and movies the were produced in some non-US countries. More specifically, in our experiments, within genre and production country features we chose "Horror", "Music", "Mystery", "History" (genres) and "CA", "ES", "DE", "HK" (countries) to be the protected group. These feature values were chosen because they represented a minority within each feature, and so are good exemplars for demonstrating the capabilities of our algorithm.

Our algorithms are also evaluated on a proprietary dataset obtained from Kiva.org, including all lending transactions over an 12-month period. Initially, there were 1,084,521 transactions involving 122,464 loans and 207,875 Kiva users. Of these loans, we found that 116,650 were funded, that is they received their full funding amount from Kiva users by the 30-day deadline imposed by the site. We selected only the funded loans for analysis. Each loan is specified by features including borrower's name/id, gender, borrower's country, loan purpose, funded date, posted date, loan amount, loan sector, and geographical coordinates. To reduce the feature space, and to solve the multicollinearity problem, highly correlated features were removed. The percentage funding rate (PFR) was added as a new feature, computed as follows:

$$PFR = \frac{1}{\#\ days\ to\ fund} * 100 \tag{22}$$

The percentage funding rate captures the speed at which a loan goes from being introduced in the system to being fully funded.[6] For example, a loan with PFR of 25% is accumulating a quarter of its needed capital each day. After preparing the data, the final features for each loan reduced to borrower's gender, borrower's country, loan purpose, loan amount (binned to 10 equal-sized buckets), and loan's percentage funding rate. We found that this dataset was highly sparse (density = $4.2e^{-5}$) and could not support effective collaborative recommendation, because a loan can only attract a limited amount of support (up to that needed for its funding). There are no "blockbuster" loans with thousands of lenders.

To generate a denser dataset with greater potential for user profile overlap, we applied a content-based technique creating *pseudo-items* that represent groups of items with shared features. We applied agglomerative hierarchical clustering [115] using the features of borrower gender, borrower country, loan purpose, loan amount (binned to 10 equal-sized buckets), and percentage funding rate (4 equal-sized buckets). We chose the cluster with the highest Silhouette Coefficient [118] of around 0.69 which indicates a reasonable cohesion of the clusters. Then we applied a 10-core transformation, selecting pseudo-items with at least 10 lenders who had funded at least 10 pseudo-items. The retained dataset has 2,673 pseudo-items, 4,005 lenders and 110,371 ratings / lending actions.

In this dataset, we observed an imbalance within the following feature values/dimensions: (percentage funding rate), (country), (economic sector), (loan amount), (borrower gender). In keeping with Kiva's mission of providing equal access to capital across regions and economic sectors, we designate the items from the sectors and countries that have less than 1% frequency in the training data as the protected group. More specifically 5 loan purposes in the economic sectors and 23 countries were selected to be the protected group. Although in both datasets we chose two features to achieve fairness within their multiple dimensions, our method supports choosing any number of such features.

Variation in diversity tolerance. By examining the $\vec{\tau}$ vectors for each user, we can get evidence for RQ1: Do users exhibit different patterns of preference across fairness dimensions? Figure 9 shows the τ values computed across for

[6] Loans not fully funded within 30 days are dropped from the system and the money raised is returned to lenders.

all users in the Kiva dataset. As the figure shows, users differ significantly in their profile entropies as measured for features of country and economic sector. (The differences across features are not meaningful, as they are a function of the prevalence of different feature values.) Some users have loans that vary widely across different economic sectors (shown in blue); others less so. Similar variety can be seen in country as well (shown in red), including some users who have loaned only to a single country.

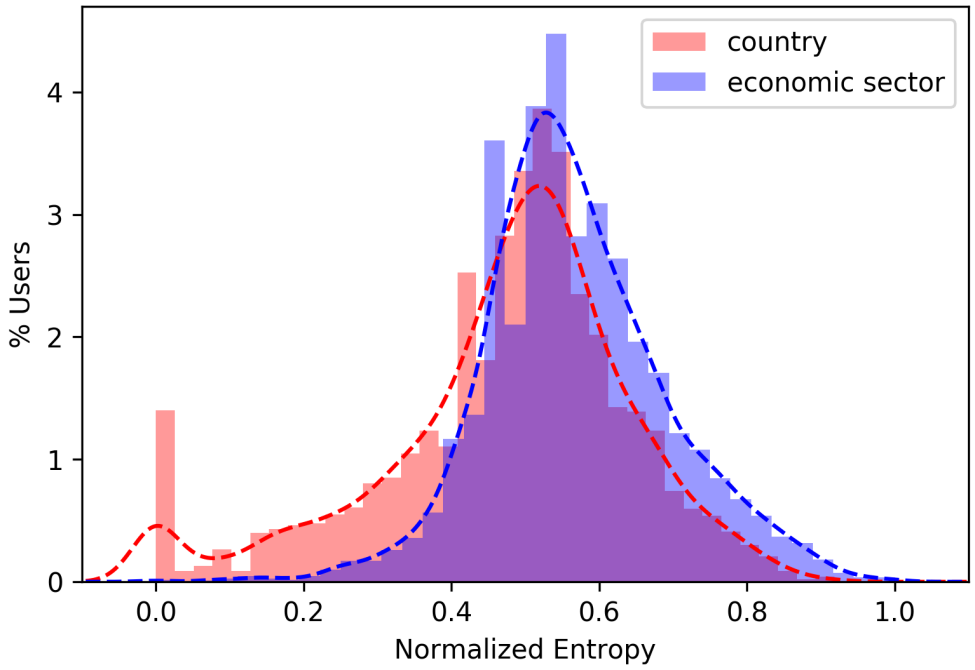


Fig. 9. User tolerance value (τ) for Economic Sector and Loan Country features in Kiva dataset.

Figure 10 shows similar results for the Movies dataset. Again, we see that users in this sample have wide individual variance in the computed τ values for different dimensions of movies. For example, the variation in the entropy for the genre dimension (shown in blue) indicates that most of the users are watching movies from various genres while there are some users who usually prefer to watch the same few genres. The variation in the production countries (shown in red) is flatter and farther to the left, indicating users' narrower choice of movies in this dimension. Possibly, these viewers mostly watch movies that are produced in their countries or in their language.

We note that different features have different baseline entropy values in each dataset. In our future work, we plan to explore a refinement of the personalized tolerance measure using conditional entropy to calculate how much each user profile adds or detracts from the entropy in a particular feature.

Comparing re-ranking algorithms. We use non-negative matrix factorization as our baseline recommendation component. The algorithm was tuned on each dataset separately to achieve the best nDCG. The algorithm was trained

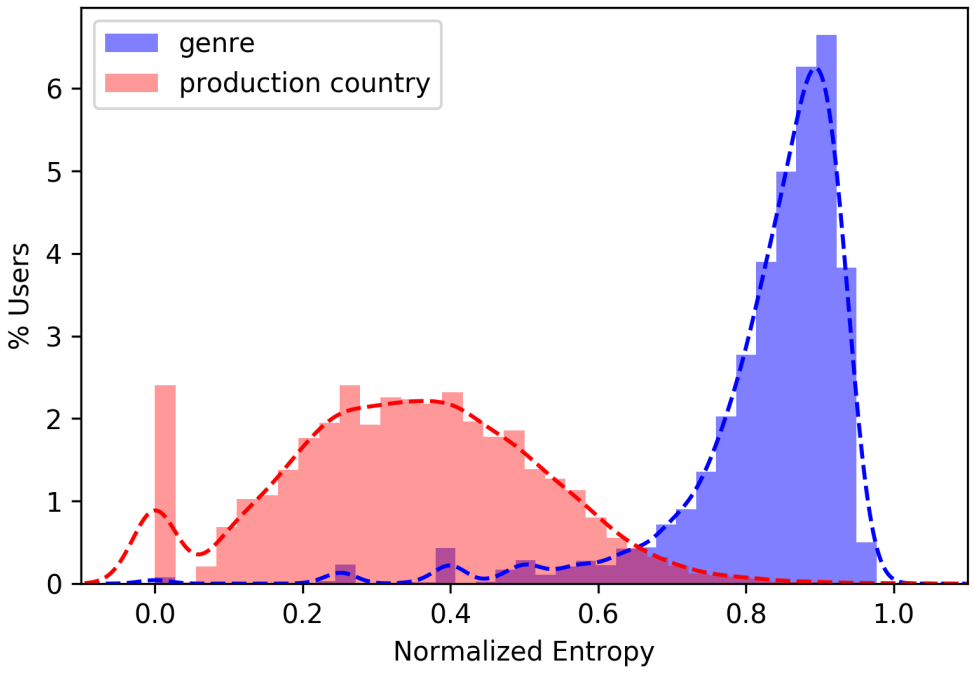


Fig. 10. User tolerance value (τ) for Genre and Production Country features in The Movies dataset.

Algorithm	1%	2%	3%
FAR	12.06%	12.09%	12.12%
PFAR	12.07%	12.08%	12.09%
MMR	12.22%	12.67%	13.08%
MMR w/ tolerance	12.83%	13.29%	12.66%
MMR w/ fairness	14.0%	15.14%	17.03%
OFAiR	16.76%	20.14%	22.81%

Table 6. Fairness vs % Accuracy Loss. Kiva dataset. Larger values mean improved fairness at the given accuracy level.

on 80% of the data and tested on the remaining 20%. The nDCG of NMF was around 0.11 on the ML dataset and 0.076 on the Kiva dataset. For each algorithm, we retrieve $k = 200$ top items for each user and re-rank the list retaining the top $k' = 10$ items.

In our experiments, we compared our OFAiR algorithm with FAR and PFAR, as our baseline methods. We also used MMR by itself, as a diversity-enhancing re-ranker, a variant of OFAiR that includes only user tolerance weights for each feature, and a variant that includes only the fairness weights for the protected feature dimensions without the tolerance weights. In this way, we can study separately the contribution of each of these aspects of the algorithm.

Table 6 summarizes the results across the different algorithms. We indicate the tradeoff between fairness and accuracy by reporting the (interpolated) protected item exposure at different levels of nDCG loss: 1%, 2% and 3%. We arrive at

Algorithm	1%	2%	3%
FAR	28.65%	28.64%	28.64%
PFAR	28.63%	28.63%	28.63%
MMR	28.44%	29.28%	29.92%
MMR w/ tolerance	28.99%	30.83%	32.13%
MMR w/ fairness	32.51%	34.44%	35.85%
OFAiR	36.59%	39.41%	41.34%

Table 7. Fairness vs % Accuracy Loss. The Movies Dataset.

the exposure values in the table by assuming a locally-linear relationship of nDCG and fairness/exposure in between different λ values, basically locating intercepts in the tradeoff graph. (See below.) The table shows that FAR and PFAR do little to improve fairness in this setting. This is not surprising as these algorithms were designed for a situation in which fairness across a number of different providers is sought, rather than the protected item balance situation here. In Figures 11, and 12 below, we will omit FAR and PFAR for this reason. Of the other algorithms, we see a small advantage for OFAiR at the 1% level of loss, increasing greatly at higher levels of loss. Both tolerance weights and fairness weights contribute to the results but their synergy in the OFAiR algorithm is apparent. It must be noted that in absolute terms, the fairness enhancement is somewhat disappointing. 16.76% to 20.14% increase still means that only 1.2 protected items will appear (on average) in each user's recommendation list.

Table 7 shows even stronger findings in favor of the OFAiR algorithm on the Movies dataset. Two trends are noticeable. One is that there is very little change in fairness for increased λ values in the MMR and MMR with tolerance cases. This trend also exists in Kiva dataset. OFAiR is a clear improvement at all levels of nDCG loss, although in absolute terms the improvement is still small.

Figure 11 shows the results on the Kiva dataset for just the MMR-based algorithms: MMR, OFiAR, and the two versions incorporating different aspects of the OFAiR algorithm, tolerance weights (users) only, and fairness weights (items) only. The figure compares ranking accuracy in the form of nDCG versus the average exposure for protected items across recommendation lists. The figure gives a more complete picture of this tradeoff than the tables above, but generally tells the same story.

The general trend shows that by incorporating re-ranking, the algorithms move the fraction of protected group items from around 11% to greater than 34%. At the higher values of λ, the algorithms are quite similar, as might be expected. When we push the algorithms to focus more on fairness, differences emerge. The OFAiR and the MMR variant with only fairness weights are very similar until we get to nDCG loss around 0.1%. At this point, the OFAiR algorithm dominates this tradeoff in terms of nDCG while keeping the fairness comparable. MMR and MMR with tolerance have curves that are essentially vertical, with very small fairness gain from diversification.

Figure 12 shows similar results for the Movies dataset. As suggested by Table 7, both MMR and MMR with tolerance fare poorly as fairness is emphasized. [7] This finding highlights the difference between a user-centered view of diversification, which MMR is targeted towards, and a fairness-oriented, provider-centered view. This effect may be due to the large feature diversity present in the Movies dataset. There are many ways for movies to be diverse without falling into the protected group.

The difference between datasets is also apparent in the relative performance of the tolerance-weighted and the feature-weighted version of the algorithm. In the Kiva dataset, fairness weights greatly enhanced fairness, competing

[7] Although note the small but intriguing bump for the tolerance-weight-based algorithm near $\lambda = 0.95$).

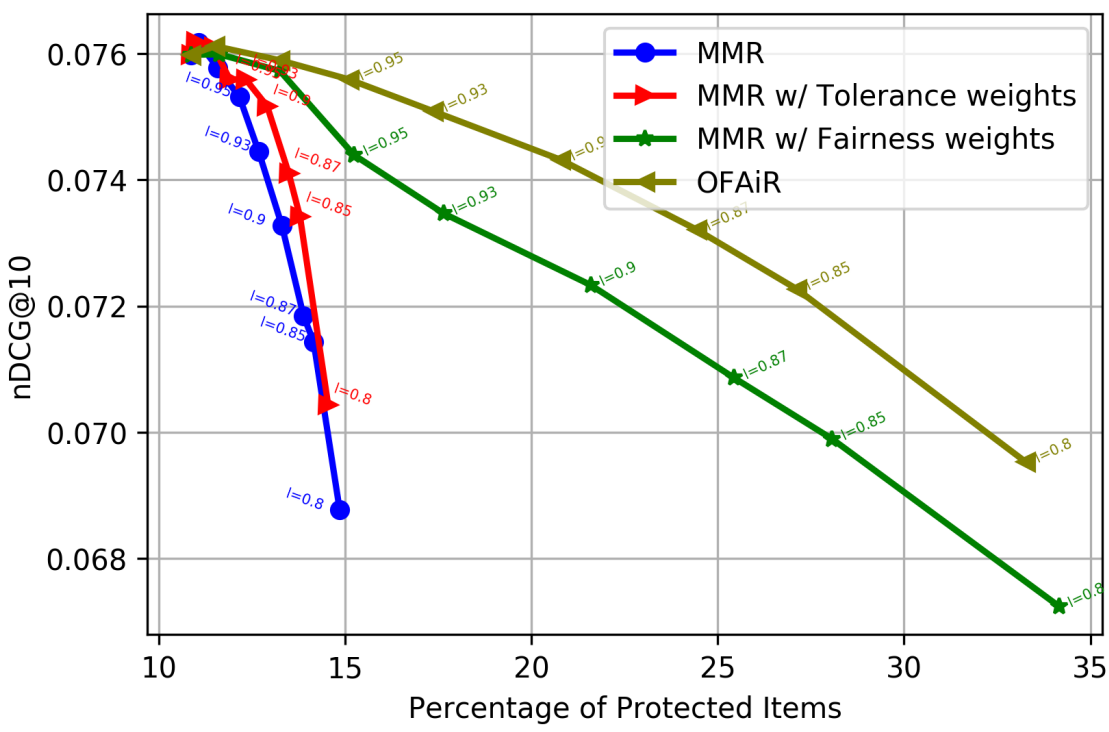


Fig. 11. MMR-based re-ranking methods. Kiva dataset.

with the OFAiR algorithm at some points in the parameter space while in the Movies dataset OFAiR surpasses all the others except in higher lambdas. The other difference is in the effect of these algorithms on the percentage of protected items achieved. As it is shown, we achieve higher fairness gains in the Movies compared to the Kiva dataset. These differences in performance could be due to domain differences in feature distributions, such that diversification along a preferred dimension does not necessarily yield protected items. The feature weights are needed to shift the algorithm's attention to the protected parts of the feature space. As before, much larger fairness gains are possible with OFAiR.

It is significant that OFAiR has a dominant position among the other algorithms in terms of the fairness / accuracy tradeoff when viewed across all items in the protected group. However, a key objective of this work was to ensure distribution of fairness enhancement across multiple categories of protected groups. Figure 13 and Figure 14 show this aspect of our experimental results.

In Figure 13 and 14, we can see the performance of all the algorithms in terms of improvement in the exposure of the protected items in each protected dimension in a more fined-grained manner. Recall that in the Kiva dataset, country and economic sector (shown as activity) were the sensitive features with 23 countries and 5 sectors labeled as protected. It is also worth mentioning that in both of these features, users had a high general entropy as well. The lighter colors show an improvement in fairness. As it is shown, the colors are darker in NMF and MMR. The right side of the heat-map contains lighter colors indicating more inclusion of protected items in recommendation lists. Lightest colors might belong to MMR with fairness weights, and after we add the tolerance weights to the algorithm it becomes slightly

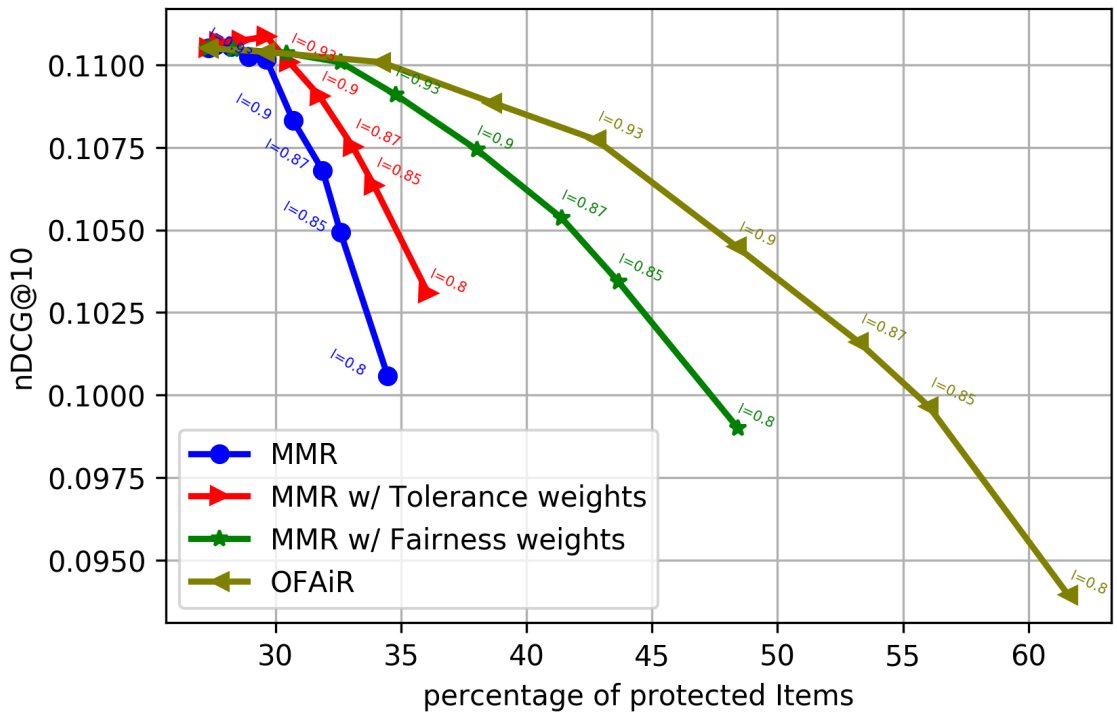


Fig. 12. MMR-based re-ranking methods. The Movies Dataset.

darker. This is due to the fairness/accuracy tradeoff noted above. For some feature values in 13, fairness is not improved by any algorithm. This is because the reranker can only improve the fairness of the results if these dimensions are present in the recommendation list of users and in these cases they rarely are. A similar trend is found in the Movies dataset, with the OFAiR algorithm, showing the best exposure across all of the protected dimensions.

3.2.5 *Related work*.

In examining prior work on re-ranking, it is important to note the distinction introduced in Section 3.2.1 above between user-oriented results diversification and fairness/provider-oriented re-ranking, which is the objective of our work. A user-oriented method will measure success by the diversity of individual lists, whereas a provider fairness approach will be measuring outcomes for providers, especially protected ones.

One of the first efforts to increase diversity in recommendations was [153], which used a taxonomic content-based similarity metric to re-rank recommendation lists. This method did not attempt to personalize its ranking goal relative to different users. The taxonomic item similarity measure used in this work may be appropriate to adapt to OFAiR, which currently uses a one-dimensional representation of item features. A steady stream of user-oriented diversification research followed, as summarized in [88].

More closely related to the present work are the FAR/PFAR algorithms in [95, 96], which have served as an inspiration here. PFAR incorporates the individualized entropy-based user tolerance weight, thus enabling it to increase accuracy for the users with more fixed tastes. As noted above, however, PFAR is based on the aspect-oriented xQuAD algorithm,

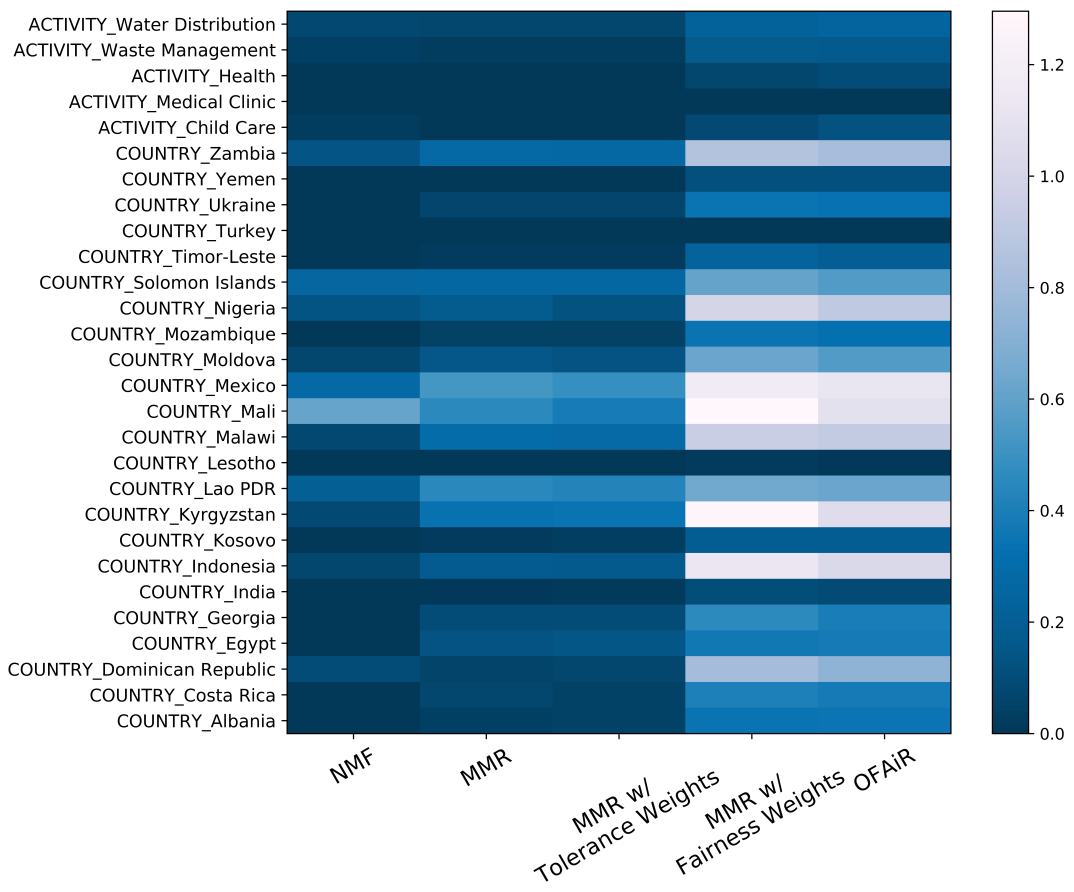


Fig. 13. Cross-category fairness of MMR-based algorithms. Kiva dataset.

which has a binary inclusion objective. Once a provider is represented in the recommendation list, it is no longer boosted in re-ranking. This makes sense for the FAR/PFAR use case, which concentrates on fairness across multiple providers. This is less appropriate for a protected/unprotected binary distinction because the objective is satisfied with only a single protected item included and there is therefore no way to approach parity of representation. This can be seen in the very small improvements in exposure found with these algorithms.

Another approach to fair ranking is the FA*IR algorithm proposed in [145]. This algorithm creates two queues: one of protected and one of unprotected items, and then integrates them to satisfy (in expectation) a probabilistic ranked fairness test. This algorithm does make the protected/unprotected assumption that we are using in this work. However, it applies only to a single such distinction. It might be interesting to extend the FA*IR model to multiple dimensions of fairness.

Fairness for multiple groups has been addressed in classification settings under the idea of *rich sub-group fairness* [80, 81]. In this work, the emphasis is on extending fairness guarantees to all possible combinations of protected groups in a dataset. The SUBGROUP algorithm alternately optimizes for a particular group's fairness and then seeks the group for

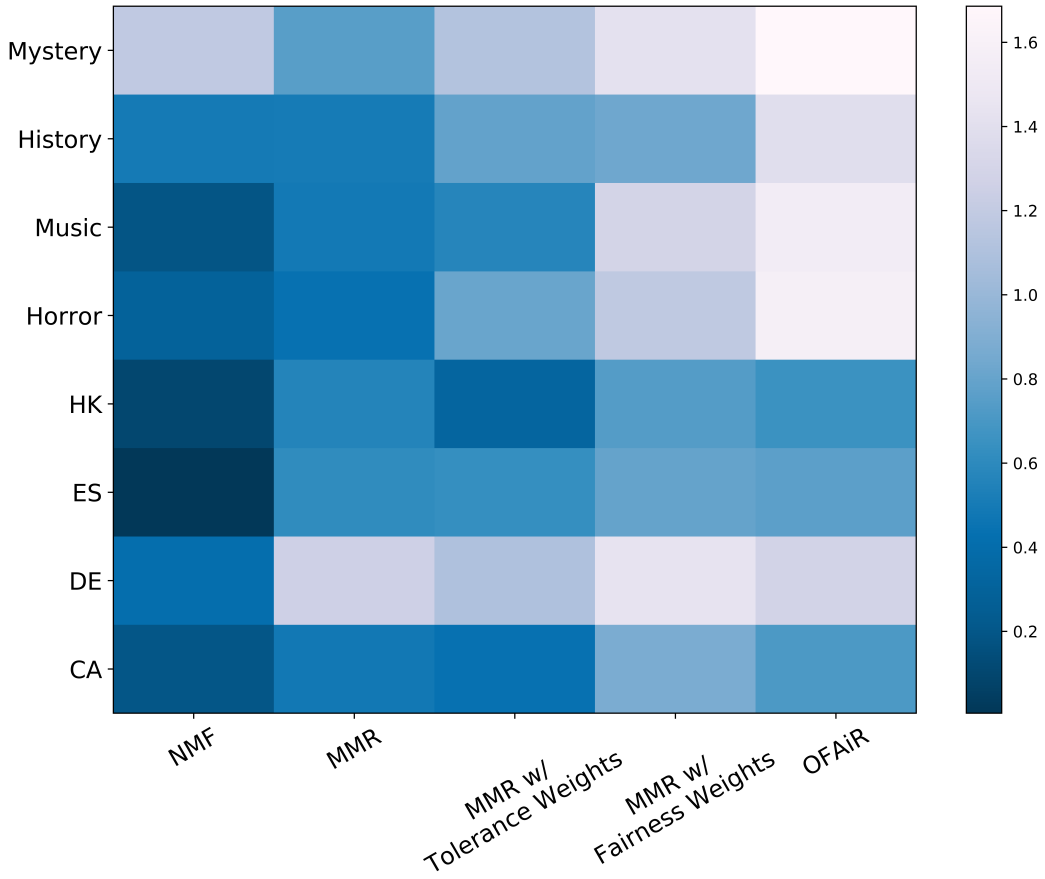


Fig. 14. Cross-category fairness of MMR-based algorithms. The Movies dataset.

whom fairness is most violated. In recommendation, we are not seeking a single decision rule, so we have a different solution in OFAiR: to distribute the optimization "cost" across different users in a personalized way.

3.2.6 ***Conclusion and Future Work***.

The results of our experiments show that OFAiR works as intended. Its proportion-based MMR model provides a much better tradeoff between ranking accuracy and fairness for the protected-unprotected case than the FAR/PFAR models explored in prior work. In the datasets under study, we show that users' tolerance for diversity varies across features, which justifies our approach of differentiating users based on the opportunities they represent for enhancing provider-side fairness.

We show that the combination of personalized, feature-specific, weights together with weights identifying protected feature values is effective with the feature-specific tolerance helping maintain accuracy and the feature weight promoting protected group items. As we showed, our method can be applied across multiple protected groups at the same time and can ensure fairness with respect to system's designed fairness goal for each feature.

One of the challenges in this work is the lack of proper datasets that have user features and these datasets are specifically lacking in domains where fairness matters. Due to this issue, we chose the Movies dataset to show the capabilities of our method.

As our future work in this section, we intend to run a more thorough experimentation with weights of the weighted cosine similarity and capture the influence of these weights on the final results. We also intend to use different recommendation algorithms as the base recommendation.

In our next work, we intend to explore further the idea of "opportunity" in subgroup-fairness-aware recommendation. In particular, when recommendations are delivered over time, prior outcomes relative to different protected groups may dictate what opportunities should be most salient at any given moment. We intend to publish this work later this year in The ACM Series on Recommender Systems.

4 FAIRNESS IN DYNAMIC RECOMMENDER SYSTEMS

In this section, we focus on the problem of *provider fairness*: namely, how to ensure that a recommender system, over time, is recommending items from protected groups in a fair manner relative to others. We are interested in a multi-aspect version of this problem, where items may be associated with multiple, intersecting, protected groups.

4.1 "And the Winner Is...": Dynamic Lotteries for Multi-group Fairness-AwareRecommendation

Our motivating example is drawn from the peer-to-peer micro-lending platform, Kiva.org. The users of Kiva are lenders, who support entrepreneurs usually from developing countries by lending small amounts. The organization has the goal of providing equitable access to capital across different regions, economic sectors and borrower demographics. This organizational mission needs to be embedded in any system deployed to recommend loans to funders. Without some control over the characteristics of the recommendations delivered, it is easy to imagine that a positive feedback loop [129] could develop in which some types of loans are increasingly disadvantaged by the algorithm.

In general, we may anticipate that a fairness-aware recommender system will need to respond to multiple *fairness concerns* simultaneously. In this work, we adopt a social choice perspective [21] on balancing different fairness concerns, which gives us a rich set of normative properties and algorithms. Social choice is fundamentally concerned with combining preferences from multiple parties into a single outcome in which all parties participate, voting being a paradigmatic example of such a choice [156]. We can think of each fairness concern as a kind of actor, with preferences over which recommendations should be delivered. Combining the preferences of multiple such concerns fits squarely into the social choice realm.

We believe that social choice is a more flexible and realistic framework for representing fairness-aware issues in machine learning than the optimization frameworks typically employed. Social choice is inherently multi-agent, and therefore, the idea of the integration of multiple fairness concerns naturally emerges, rather than being a complex add-on. Importantly for recommendation problems, social choice naturally allows for heterogeneity and hence personalization across decision instances since a user is just another agent with preferences over the outcome. Finally, fairness is inherently a social and political construct, and a social choice formalization allows the preferences of different actors to be foregrounded, rather than relegated to the black box of machine learning optimization. The study of fairness has a long history in the social choice literature [142, 156].

Contribution. In this project we propose a novel framework for recommender systems we call *Social Choice for Re-ranking Under Fairness* (SCRUF). SCRUF uses multiple fairness metrics to evaluate the history of recommendation

delivery and determine whether and how to adjust its performance. It uses feature-specific re-rankers to improve fairness, selecting one re-ranker (possibly non-deterministically) at each time point. We use a set of social choice inspired algorithms to allocate re-rankers to users based on the user preferences. This framework abstracts the particular fairness metrics away from the recommendation algorithm design, an approach that becomes unwieldy when attempting to incorporate multiple fairness concerns. It also supports a dynamic balance between the interplay between personalization and fairness and is therefore sensitive to context and individual differences. We demonstrate the efficacy of our design on two recommendation domains.

4.1.1 ***Fairness-aware Recommendation.***

Both in the machine learning and the recommender systems formulation of fairness, there has been little recognition of the intersection of multiple fairness definitions and dimensions, although recent work has noted the benefits of combining multiple fairness definitions [14]. Most existing research considers only a single protected class, and even in cases where multiple groups are considered as in [23, 67, 80, 152], fairness is conceived the same way for all groups. In recognition of the complexity of the fairness concept, we seek to accommodate different definitions of fairness put forward by different stakeholders, all of which must be integrated in a single framework. This nuanced understanding of the value of fairness is essential for capturing the richness of this social construct in real-world settings such as those studied by scholars of organizational justice, anti-discrimination law, and social justice.

As discussed in before, two standard approaches have emerged to integrating fairness concerns into recommender systems. The *integrated* approach builds a fairness constraint into the recommendation model itself, for example as a regularization constraint, balancing between accuracy and fairness in the optimization process for the recommendation model [75, 140]. The *re-ranking* approach applies fairness to the output of a recommendation algorithm, reordering the results. Re-ranking approaches offer a number of advantages. First, the trade-off between accuracy and fairness can be tuned without re-learning the recommendation model. Second, researchers have found that re-ranking can achieve better trade-offs versus accuracy with this type of model [3, 50, 96]. Due to these advantages we choose to use the latter method.

There has been some recent work in recommendation fairness that incorporates multiple fairness dimensions, most notably the OFAiR method discussed in [125]. This work integrates user tolerance towards different types of variation in item features with a representation of protected groups that spans multiple dimensions. The authors were able to show a beneficial trade-off between fairness and accuracy and improved results across different categories of protected groups. One drawback of the method of [125] is that it relies on weights associated with item features to boost the inclusion of protected group items into recommendation lists. Balancing across different groups requires careful setting of these weights and sometimes unexpected interactions arise.

In addition, this and similar methods are list-wise approaches, which aim to increase protected group representation in individual lists. However, as noted above, the real objective of fairness-aware recommendation is to enhance fairness as measured historically, across the behavior of the system as it provides recommendations to many users over time. The personalization element of recommendation means that this objective cannot be targeted directly: any given user may or may not constitute a good opportunity to enhance fairness relative to a particular protected group. In [125], this aspect of the problem was addressed by incorporating user-specific weighting, which can be interpreted as a preference in a social choice setting.

Analyzing user characteristics enables the system to determine which users constitute good opportunities to pursue different fairness goals. However, there is another side of the problem. At any point in time, the system's historical

performance may have been more favorable to one protected group than another. If we only look at the users, we ignore the signal from past performance about where fairness needs are most critical.

4.1.2 ***Computational Social Choice for Fair Recommendations***.

Fairness has been extensively studied in the economic field of social choice including work focusing on the division of continuous resources such as land or water [104], on more discrete, indivisible settings such as goods and services [133, 134] and more fundamentally in the areas of political economy having to do with justice and fair distribution of resources to individuals [109, 112, 142]. In its classical formulation, *social choice* concerns itself with the study of how groups, where each member is endowed with their own preferences, make decisions that must be then shared by that group [122]. To these considerations the field of computational social choice adds computational tools including algorithms, complexity, and big data [21, 100].

From the literature on social choice we will focus on the *allocation* setting (which is a generalization of the classical matching setting) [21]. In allocation, the items within A are to be distributed or *allocated* to the set of agents in N. Hence, the social part of social choice reinforces the idea that a set of preferences need to be considered and combined since the outcome of a social choice process will affect all the agents. There have been many practical applications of matching and allocations from kidney allocation [117] to conference paper reviewing [92]. There are extensive studies of algorithms for a variety of settings [98] and the study of fair allocations in multi-agent systems is a popular topic in the broad area of artificial intelligence [7]. Equity and other concerns, formalized as economic axioms have a long history in social choice both in allocation [142] and voting [156]. It is this long history of study of the axioms, or properties, of the algorithms and aspects including *fairness* we hope to leverage.

Rather than thinking of integrating the concerns of protected groups into re-ranking decisions indirectly, in the form of weights for particular feature values as in [125], our approach *Social Choice for Re-ranking Under Fairness* (SCRUF) conceives of both users and protected groups as actors with preferences over the items that may be recommended. The goal is to achieve an integration of these preferences over the whole recommendation history. We explore a class of solutions to this problem that assumes multiple fairness criteria can be persued at the same time by deciding which objectives to address and which users to address them with. This choice is made non-deterministically and also take into account both the current state of the recommendation history, which we can think of as defining immediate needs, and the user's propensity towards different item categories, which define current opportunities, in the context of providing personalized recommendation results.

A social choice perspective on recommendation has emerged in recent research as a possible source of methods to integrate the viewpoints of multiple agents or priorities [29]. Chakraborty et al. [35] build a recommendation system for finding fair group recommendations through viewing them as elections between various signals of popularity, leading to a shared group recommendation. This does not take the important aspect of *personalization* into account that we address in SCRUF. Sühr et al. [128] explore driver assignment in two sided matching markets with an emphasis on producer and rider fairness. Patro et al. [107] propose a recommender system for two-sided matching markets with the goal of fair exposure amongst producers. This differs from our work in that the fairness metrics are fixed and embedded into the matching algorithms themselves. Finally, Lee et al. [91] propose a system that uses social choice to embed normative properties for algorithmic governance into the algorithms themselves, as demonstrated on a food bank matching scenario. However, none of this research considers multiple fairness concerns on the provider side of a recommendation system as required in the Kiva case or the dynamic response to historical fairness outcomes as embodied in SCRUF.

	F_1 : Region	F_2 : Gender	F_3 : Sector	F_4 : Amount
v_1	Africa	Male	Agriculture	\$0-\$500
v_2	Africa	Female	Health	\$500-\$1,000
v_3	Middle-East	Female	Clothing	\$0-\$500

Table 8. Set of Potential Loans.

4.2 Formal System Specification

To leverage the power of recommender systems for personalized fairness we will define a set of choice functions to promote fairness. We first detail the formal notation for our recommender system and how to view it as a social choice problem. We then describe our overall system in terms of choice functions and how these can be used to promote fairness.

4.2.1 *Recommendation Systems with Protected Values.*

In a recommendation system setting we have a set of users $\mathcal{U} = \{u_1, \ldots u_n\}$ and a set of items $\mathcal{V} = \{v_1, \ldots, v_m\}$. For each item $v_i \in \mathcal{V}$ we have a k-dimensional feature vector $\vec{v_i} = \langle f_{i1}, \ldots f_{ik} \rangle$ over a set of categorical features $\mathcal{F} = \{F_1, \ldots, F_k\}$, where each feature F_i has finite domain $\mathcal{D}_i$. We assume that all elements in $\mathcal{V}$ share the same set of features. Consider a running example of a funding site that shows micro-loans in emerging markets to potential funders. In this example we have $m = 3$ items in the database which share $|\mathcal{F}| = 4$ features: {Region, Gender, Sector, Amount}, where each has its own domain. For example, $\mathcal{D}_1$ = {Africa, Middle-East, North America}. This setting is illustrated in Table 8.

Though our items are comprised of a set of features, we start with the view that not all features should be treated the same. We assume that there is a subset of the features $\mathcal{S} \subseteq \mathcal{F}$ that are denoted as *sensitive* and there is a subset of values for each such feature, i.e., $\mathcal{P}_i \subseteq \mathcal{D}_i$ that constitute the *protected values* of the sensitive feature. That is, given all items in the recommendation system, we have a subset of sensitive features, each of which may contain protected values. Turning back to our running example, we may wish to target $\mathcal{F}_1 = \{Region\}$ as a sensitive feature and the values $\mathcal{P}_1$ = {Africa, Middle-East} as the protected values. We could designate sensitive features and protected values based on operational goals such as regions or genders that are funded less frequently.

For our recommender system we have a personalized ranking function $R(u_i, \mathcal{V}) \rightarrow \sigma_i(\mathcal{V})$, which given user u_i and set of items $\mathcal{V}$ produces a permutation, i.e., a ranking, over the set of items for that user, i.e. a recommendation. As a practical matter, the recommendation results will always contain a subset of the total set of items, typically the head (prefix) of the permutation σ_i up to some cutoff number of items.

4.2.2 *Re-Ranking Functions.*

In order to promote fairness, we assume that we are also given a set of re-ranking functions $\mathcal{K} = \{\kappa_1, \ldots, \kappa_{|\mathcal{S}|}\}$, which are a set of functions, one for each sensitive feature. For feature j, the re-ranking function $\kappa_j(\sigma) \rightarrow \sigma'$ will take a permutation σ and produce a new permutation σ' of the set of items that is more "fair" towards the particular protected feature values associated with $\mathcal{S}_j$. In real applications, the final recommendation slate is a short list of the most preferred items from this final, re-ranked permutation.

For our system we assume a common form of all re-ranking functions, where the permutation is achieved by sorting items based on a score, and the score is a linear combination of the score from the recommender system (the determiner of the original σ ranking) and a score based on the presence of the protected feature, such that protected group items

are moved up in the ranking list [4]. The scoring function ρ for user u, an item v, and a sensitive feature j is defined as follows:

$$\rho(u, v, j) \triangleq \lambda_j(R(u, v) + (1 - \lambda_j)\mathbb{1}_{\{v \in \mathcal{S}_j\}} \tag{23}$$

The indicator function $\mathbb{1}_{\{v \in \mathcal{S}_j\}}$ has the value 1 if the item v has a protected value of sensitive feature $\mathcal{S}_j$, and 0 otherwise. λ_j is a feature-specific parameter that controls the trade-off between accuracy (as represented by the original σ ranking) and fairness (as represented by the boost given to protected items). All of the items in the list σ are re-scored using ρ, sorted in decreasing order, and truncated to produce the final σ' recommendation list.

4.2.3 *Metrics for Fair Recommendation.*

There are a wide variety of metrics that have been proposed for measuring the fairness of a recommendation result or set of recommendation results. In our setting we are not concerned with the fairness of a particular recommendation but rather the *history of recommendations* the system has generated over within some time window. Hence we track the prior history of recommendations lists that have been generated $\vec{L} = [\ell_1, \ldots, \ell_{t-1}]$ for the users (with a slight abuse of notation) $\vec{U} = [u_1, \ldots, u_{t-1}]$ that have appeared to the system.

Rather than commit to one particular metric in our system, we assume a family of functions $\mathcal{M}_j : \vec{L} \times \vec{U} \rightarrow \mathbb{R}$, one for each sensitive feature $\mathcal{S}_j$, mapping from a set of recommendation results L and the set of users U to whom each of those results have been delivered to a value indicating the degree of fairness in the total set of results. We assume that a higher $\mathcal{M}_j$ values indicate a fairer result. Without loss of generality, we assume that each metric has values in the range $[0, 1]$.

We will assume that the re-ranking functions have a non-decreasing impact on their associated fairness metric. That is, given a recommendation result σ, $\mathcal{M}_j(\sigma, u) <= \mathcal{M}_j(\kappa_j(\sigma), u)$. Because of this property, we can interpret a fairness score as indicating the relative number of times we want to select the different re-ranking functions. If the metrics were all equal, then the different re-ranking functions would be equally desirable.

Note that the inclusion of $\vec{U}$ as an argument to the M_j functions allows us to include a family of fairness metrics that are sensitive to the user's level of interest in items that vary on different feature dimensions. Each recommendation result is then evaluated relative to the user to whom it is delivered. For example, even if our recommendation history tells us we should be favoring loans in the textile sector, it may not be as valuable to recommend such loans to the agriculture-focused user, as opposed to a user that has proved to be more flexible in which sectors they support.

4.2.4 *User Preferences for Fairness.*

To incorporate the social choice aspects of the problem, user preferences over both the overall set of items as well as preferences about the *re-ranking functions themselves* need to be taken into account. In a traditional social choice setting we have a finite set of agents $N = \{1, \ldots, n\}$ and a finite set of alternatives $A = \{1, \ldots, m\}$. Each agent $i \in N$ has a preference $\succsim_i$ over the alternatives. Typically these preference are expressed as a binary relation (weak or linear order) over the set of alternatives A.

While the user preferences are handled by the personalized ranking function $R(u_i, \cdot)$ we will also incorporate the preference over the fairness functions themselves. To this end, replacing the preferences $\succsim_i$ above, we assume that for each user we are also given a vector of real numbers, $\vec{\tau}_{u_i} = \{\tau_1, \ldots, \tau_k\}$ of length k, which indicates the tolerance (preference) of u_i for variation relative to feature F_k. We can then view our problem as one of allocating re-ranking functions to users based both on the their preferences and on the current fairness status.

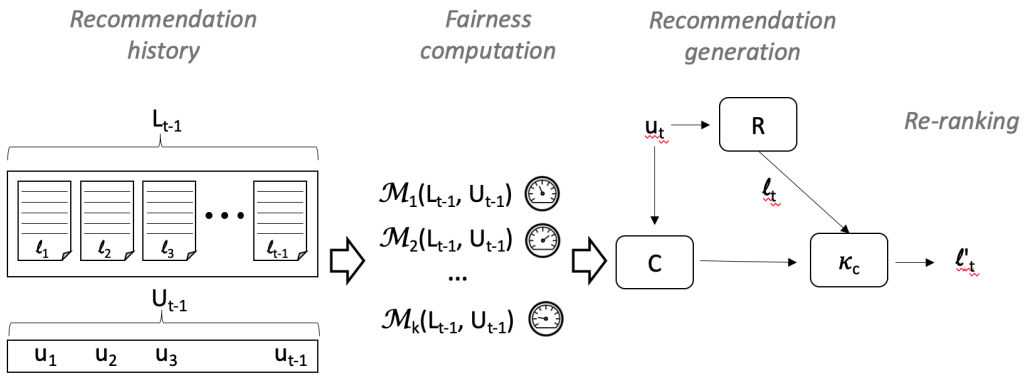


Fig. 15. SCRUF framework, a snapshot at time t: On the left are the recommendation lists L computed at prior time points. Fairness metrics M compute the fairness state, which is input to the choice function C, selecting a re-ranker κ_t that processes the recommendations ℓ from R into a final re-ranked slate ℓ'.

[54] introduced the concept of personalized diversity in collaborative filtering using a user-specific measure based on information entropy. High entropy in a categorical distribution of user profile represents high interest of user in diversity. Liu et al. [95, 96] integrated this concept for the first time in recommendation re-ranking using a quantity τ_u, a user-specific measure of interest in diversity.

$$\vec{\tau}_u(F_j) \stackrel{\triangle}{=} -\sum_{f \in F_j} P(f|u) \log P(f|u), \tag{24}$$

where $P(f|u)$ is computed as the fraction of items in the user's profile that have the feature value f. This can be interpreted as the user's likelihood of liking items with that value. The higher the entropy value is for a user on a feature, the higher their tolerance to see diversity within that feature. We assume that we can interpret this as a *preference* in the social choice sense. In our running example, a user may be particularly dedicated to a particular economic sector, agriculture for example, and may only have supported loans in this sector in the past. Hence, they would have low tolerance for variation in this feature. Note that since these map onto $\mathbb{R}$ we could both interpret these as ordinal rankings: as a preference order for user u_i, $\succsim_i$, over the set of f_j; or as cardinal valuations.

4.2.5 ***Overall Framework***.

SCRUF is our framework for explicitly representing the design decisions that enter into trading off between accuracy and fairness across multiply-defined and intersecting protected groups in the setting described above. Figure 15 shows the general process that the framework instantiates by looking at a snapshot in time. A user u_t arrives at the system and the base recommender algorithm $R(u_t, \mathcal{V})$ generates a recommendation list ℓ_t.

SCRUF is able to accommodate different metrics, one for possibly each sensitive feature, $\mathcal{F}_j$, $\mathcal{M}_j : \vec{L} \times \vec{U} \rightarrow \mathbb{R}$. Since we have access to the history of all recommendations, we can derive particular fairness results relative to the different sensitive dimensions indicated by the meters associated with each metric $\mathcal{M}_j$. This set of metrics are input to a choice function C which picks one dimension to prioritize and selects the corresponding re-ranking function κ_c, which is applied to ℓ_t resulting in a final set of recommendations ℓ'_t that is displayed to the user. (As a practical matter, our implementation described below groups users into batches and calculates the fairness metrics only once per batch.)

Note the arrows from the user u_t point both to the recommendation algorithm R where the algorithm takes into account the user's inferred preferences over items in attempting to predict the user's preferences, and also to the choice function C that may take into account the user's inferred preferences from their tolerance scores $\vec{\tau}_{u_i}$ over item features in choosing a re-ranking algorithm.

4.2.6 ***Choice***.

As noted above, we are investigating both deterministic and non-deterministic mechanisms for selecting, at each point when a recommendation is generated, a single κ_j function to use to re-rank those recommendation to a specific user u_j. The choice function C uses the current state of the recommendation history, as defined by the M_j metrics over the recommendation history so far and optionally the identity of the current user, to compute a feature $c \in \mathcal{S}$ whose corresponding re-ranking function κ_c will be applied to the recommendations for this user, i.e., $\kappa_c(R(u_j, \mathcal{V}))$. We prefer this simple uni-dimensional re-ranking scheme over one that attempts to incorporate multiple fairness dimensions are considered at once because it creates independence between re-ranking operations and avoids complex parameter interactions that might occur in attempting to compute a single re-ranking incorporating multiple fairness dimensions.

One issue that may arise, and we discuss in the next section, is that we need to decide when to *stop* running the re-rankers for a particular feature. If the historical data at time t shows that we have been fair towards a particular feature, we do not need to promote it in this iteration. In the following we will describe how we select which features to consider.

4.3 Choice Functions

What remains to be specified within the SCRUF framework is the choice function C. There are wide variety of ways such a function could be formulated. In this work, we explore three different variants of the lottery, where probabilities are set for each re-ranker and a single re-ranker is chosen by sampling from this distribution.

In order to pick a choice function at time t we will, for operational concerns, also have a parameter w_b which defines the historical (backward) *window* over which we are concerned with our fairness metric. This means that our fairness metrics $\mathcal{M}_j$ are run over over the set of users and lists between $[t - w_b - 1, t - 1]$. Recall that our fairness metrics are all in the range $[0.0, 1.0]$ with 1.0 representing full fairness for that metric. In a slight abuse of notation we will use $\mathcal{M}$ to represent this list of values.

Also, for operational reasons, we are given an ϵ which represents a non-zero cutoff or tolerance for each metric. We will only consider running re-rankers when $\mathcal{M}_j - \epsilon > 0.0$. This allows us to focus on the sensitive features with greater unfairness and provides a way to guard against an over-emphasize on protected groups at the expense of accuracy.

Let the *unfairness vector* be (point-wise) $\vec{UF} = 1 - (\mathcal{M} - \epsilon)$. Intuitively, this captures how unfair we are being towards a particular feature. The following discussion will treat this vector as a probability distribution, and so it will be normalized to unit length, $\vec{UF} = UF/\sum_i(UF_i)$. Note that this will implicitly assume that our target each element of $UF = 1/|UF|$, i.e., that we want unfairness to be equal across all aspects. This is a byproduct of our metrics taking values in $[0.0, 1.0]$ since if all metrics were 1.0 then our normalization would be undefined, hence the ϵ. Normatively, this makes sense in that if we are being unfair the same amount to all sensitive features then we want an equal probability distribution over all features.

4.3.1 ***Baselines***.

We employ two simple baseline techniques to contrast with the more dynamic options discussed below. The simplest is the *Fixed Lottery*, in which each re-ranker is chosen with equal probability for each set of recommendations delivered.

This method has the benefit of great simplicity and does not require any bookkeeping about the historical fairness of the system.

If we want to use the information in the UF vector, another simple alternative is a deterministic *Least Misery* algorithm, in which we identify the feature in the UF with the highest value (most unfair) and chose the associated re-ranker. This method directs the system's attention to the dimensions with the worst historical performance and attempts to correct that. It will be dynamic in the sense that as the performance improves in one dimension, another may be chosen.

4.3.2 ***Dynamic Lottery***.

We have found that it is typical for some dimensions to be more difficult to achieve fairness for than others. In particular, some types of items are rarely retrieved by the base recommendation algorithm and therefore only small improvements can be had through re-ranking. Applying the least misery algorithm in such a setting could lead the system to concentrate all of its effort on one of these intractable dimensions and miss opportunities to achieve fairness in other parts of the item space.

To avoid this problem, we can use UF as a lottery over the re-rankers and select a re-ranker with probability proportional to its weight, so that the poorest performing dimensions (most unfair) would have highest probabilities of being chosen. This is a *Dynamic Lottery* as opposed to the fixed version above, because the probability associated with each re-ranker will change as a function of system performance.

4.3.3 ***Allocation Lottery***.

While the above method is sensitive to the dynamic properties of the system, it is not sensitive to each user's particular propensity or interest towards different dimensions. In prior work, the ability to re-rank selectively based on user characteristics was found to yield a better tradeoff between accuracy and fairness [96, 125]. For this reason we consider in this section *randomized allocation mechanisms* that consider both users and fairness concerns. In a such a mechanism we compute a fractional allocation that we can then sample from in order to compute an assignment. So, for a given set of n agents and m objects, we compute a bi-stochastic matrix of size $n \times m$ which represents the fraction of a particular item is allocated to an agent.

We use a modification of the *probabilistic serial (PS)* mechanism [19]. In PS, also known as the simultaneous eating algorithm, each object is considered to have an infinitely divisible probability weight of one. To find the allocation every agent, simultaneously and at the same speed, begins "eating" their most preferred object that has not been completely consumed already. Once an object is consumed, the agents move to their next most preferred object until all objects have been consumed. The random allocation of an agent by PS is the amount of each object he has eaten. PS satisfies a number of important fairness and efficiency criteria [8, 9] and has been used in real allocation settings such as course selection at universities [22].

In translating our recommendation system setting to use PS we use again the tolerance values τ of the agents as representative of their preferences. As PS only requires ordinal preferences we simply use the ordering and not the actual values (breaking ties randomly when needed). A key concept in PS is the idea of an object's capacity, how much it is available to be allocated. We set the capacity of each re-ranker to mirror the sampling lottery probability from above. Specifically, each re-ranker has weight $w_f * UF_i$, thus limiting the amount of that re-ranker to allocate. We then run the PS algorithm and get a fractional allocation for each user for each re-ranker. We interpret this fractional allocation (normalized into a distribution) as the probability that the user should be assigned that particular re-ranker.

4.4 Methodology

4.4.1 *Evaluation Metrics*.

As our work here concentrates on ranking performance, we use normalized discounted cumulative gain (nDCG) as our measure of recommendation accuracy. Note that we are only evaluating re-ranking algorithms so nDCG is limited to some extent by the performance of the base algorithm to which the re-ranking is applied.

Provider-side fairness metrics come in two basic varieties. There are those that respond to the appearance of protected items in a recommendation list: *exposure* metrics, and those that take into account the suitability of the target user as *hit-based* metrics [1]. In this work, we concentrate on exposure metrics, in particular, protected class exposure, which calculates the fraction of a retrieved recommendation list belongs to a particular protected class. This value is related to the fairness concept of "statistical parity," measured relative to items' level of promotion within the recommender system. Because list lengths are fixed (10 in our case), the exposure of unprotected items is just one minus the protected group exposure. We note, however, that exposure metrics may overstate the effectiveness of re-ranking, since they do not evaluate the quality of the protected items promoted into the recommendation list. Exposure e_j of the protected class items relative to feature S_j is defined as:

$$e_j(\ell) = \frac{\sum_{v \in \ell} \mathbb{1}_{\{v \in S_j\}}}{|\ell|} \tag{25}$$

Given this definition, our fairness metrics use the notion of absolute unfairness [140], and have the following form:

$$M_j(L, U) = 1 - |1 - 2\frac{\sum_{\ell' \in L} e_j(\ell')}{|L|}| \tag{26}$$

where L is the list of recommendation $L = [\ell_{t-w_b}, ..., \ell_{t-1}]$. Note that this definition implies ideal fairness consists of equal exposure, that is, recommendation lists containing 50% protected group items.[8] We plan to explore other characterizations of exposure and other fairness metrics in future work.

Each metric has a maximum fairness of 1 and therefore it is possible to calculate regret ω_j as the difference between this ideal $\mathcal{M}_j^*$ and the current state of the metric $\mathcal{M}_j$. For reasons of space, we report only on the average regret over all metrics, and leave more detailed analysis for future work. To understand the consistency of algorithm performance, we also compute the variance of the average regret across time periods.

4.4.2 *Dataset*.

We tested our model on two datasets. The first is The Movies Dataset, which was obtained from the Kaggle website and contains the metadata of 45,000 movies listed in the Full MovieLens Dataset [9] which were released on or before July 2017. Although movies are not a domain to which important fairness concerns are typically applied, we use this dataset as a well-known example with a rich set of provider-side features. Additionally, we extracted two features that contain demographic information on the movie directors and screenplay writers.

The dataset contains 26 million ratings from 270,000 users for all 45,000 movies. Ratings are on a scale of 1-5. Each movie contains a set of features from which the following were used in this project: genres, original language, release date, run-time, popularity, director gender and writer gender. A sample of this dataset was extracted which contained the 361,468 ratings from 6,000 users on 6,037 items (density of 0.99%).

[8] The metric as defined penalizes lists with more than 50% protected items, which might seem counterproductive. However, as a practical matter in our experiments higher exposure values for protected items were never achieved.

[9] https://grouplens.org/datasets/movielens

Dataset	Features	Protected Values	Unprotected Values
Kiva	Activity	Bicycle Repair, Gardening, Souvenir Sales	Taxi, Fishing, Vehicle Repairs
	Country	Indonesia, Nigeria, Yemen	Cameroon, Armenia, Lebanon
	Gender	Male	Female
MovieLens	Genres	Documentary, Foreign, War, Western	Adventure, Crime, Action, Comedy
	Writer Gender	{'01', '012'}	{'0', '02', '12', '2', '1'}
	Director Gender	{'01', '1'}	{'0', '12', '012', '02', '2'}

Table 9. Examples of sensitive features and their values.

All the features with numerical values were transformed into categorical values. Release date is bucketed into four groups, run-time into six groups and popularity is bucketed into five groups. In this dataset, three types of genders were present: 0, 1, 2. And each movie can be directed or written by a group of directors or writers. To capture this diversity, gender was discretized into seven groups. For example if a movie is directed by all the genders, we assign 012 for the gender information and if it is directed only by one gender, a single number was assigned to that movie e.g. 0, 1 or 2. All the categorical features were transformed into dummy variables, resulting in a total of 335 binary features. Table 9 shows some examples of the sensitive features and their protected values.

In a fielded application, the choice of sensitive features and protected groups within those features may be determined by legal liability or business model considerations. Lacking this type of insight, we chose to identify protected features as those associated with rarely-recommended items. To determine the protected values of each feature, we performed a trial run of recommendation generation over the data set, and examined the distribution of features in the results. In a live system, historical recommendation data would be available over which to calculate this distribution. The values in the 25th percentile of the distribution were selected as the protected group for that feature.

Our algorithm is also evaluated on a proprietary dataset obtained from Kiva.org, including all lending transactions over an 12-month period. Initially, there were 1,084,521 transactions involving 122,464 loans and 207,875 Kiva users. Of these loans, we found that 116,650 were funded, that is they received their full funding amount from Kiva users by the 30-day deadline imposed by the site. We selected only the funded loans for analysis. Each loan is specified by features including borrower's name/id, gender, borrower's country, loan purpose, funded date, posted date, loan amount, loan sector, and geographical coordinates. To reduce the feature space, and to solve the multicollinearity problem, highly correlated features were removed.

After preparing the data, the final features for each loan reduced to borrower's gender, borrower's country, loan purpose, loan amount (binned to 10 equal-sized buckets), and loan's percentage funding rate. We found that this dataset was highly sparse (density = $4.2e^{-5}$) and could not support effective collaborative recommendation, because a loan can only attract a limited amount of support (up to that needed for its funding). There are no "blockbuster" loans with thousands of lenders.

As we explained in the previous section, to generate a denser dataset with greater potential for user profile overlap, we applied a content-based technique creating *pseudo-items* that represent groups of items with shared features. The retained dataset has 2,673 pseudo-items, 4,005 lenders and 110,371 ratings / lending actions.

To identify the protected values for each feature, we applied the same method as for the MovieLens data set. We assigned the values that their frequencies are in the 25 percentile of the distribution to the protected group for each feature. The final number of features are 231 for this dataset.

4.4.3 ***Experiments***.

Our experimental methodology is designed to highlight differences between these choice functions. We followed a typical recommendation evaluation process with each user's profile split into 80% training and 20% testing. We chose non-negative matrix factorizing (NMF) as our base algorithm [132] based on prior experience with these data sets. We plan to explore the interaction between base algorithm and choice functions in future work. The factorization model was built using the training data and then used to generate a recommendation list ℓ for each user. Arrival time was simulated in our experiments. Users were shuffled randomly and grouped into batches of size 0.5% of all the users, where each batch was considered to be a single time step. For each batch, we computed fairness metrics $\mathcal{M}$ over the previous 20 batches, so that the backward window w_b equals approximately 10% of the test data. [10] The experiment was run for each of the four choice functions described above: Fixed Lottery, Deterministic Least Misery, Dynamic Lottery, and Allocation Lottery. For the choice functions dependent on $\mathcal{M}$, we computed the lottery probabilities once per batch.

The results of the different algorithms were compared in summary and over the course of each experiment's iterations. Overall nDCG was compared to establish the accuracy loss for each choice function. Over the course of each experiment, we computed cumulative fairness regret on each fairness dimension and on average.

4.5 Overall Results

Table 10a shows the overall results for the MovieLens data set. The first point to notice is that fairness is greatly improved (5x) over the base algorithm for all of the re-ranking methods, which is to be expected. Interestingly, the Fixed choice function, which chooses among the re-rankers with equal probability has the best fairness over all experiment iterations taken as a whole. The other re-rankers are similar. All of the re-rankers show a reduction in ranking accuracy, around 25% of nDCG. We did not seek to minimize nDCG loss in these experiments as doing so would reduce the impact of any given re-ranking operation and require a longer experiment to tease out differences.

Algorithm	nDCG	Fairness	Fairness Variance
Base (NMF)	0.143	0.039	5.3e-6
Fixed	0.107	0.179	3.8e-3
Least Misery	0.106	0.178	1.3e-3
Dynamic	0.104	0.170	1.5e-3
Allocation	0.109	0.171	2.3e-4

(a) MovieLens data set

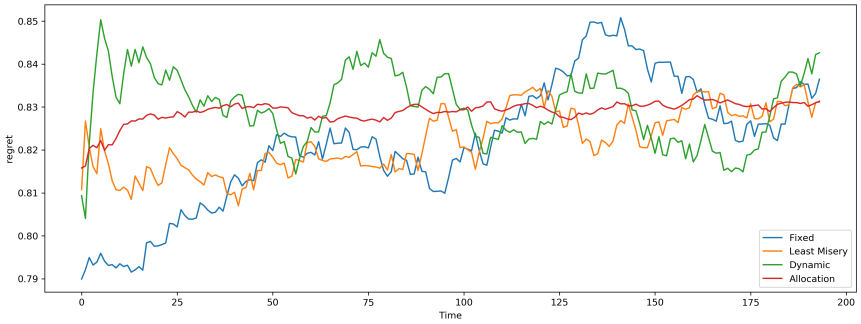
Algorithm	nDCG	Fairness	Fairness Variance
Base (NMF)	0.057	0.214	2e-4
Fixed	0.045	0.323	1.1e-3
Least Misery	0.043	0.322	9e-4
Dynamic	0.045	0.325	1e-3
Allocation	0.048	0.327	1e-4

(b) Kiva data set

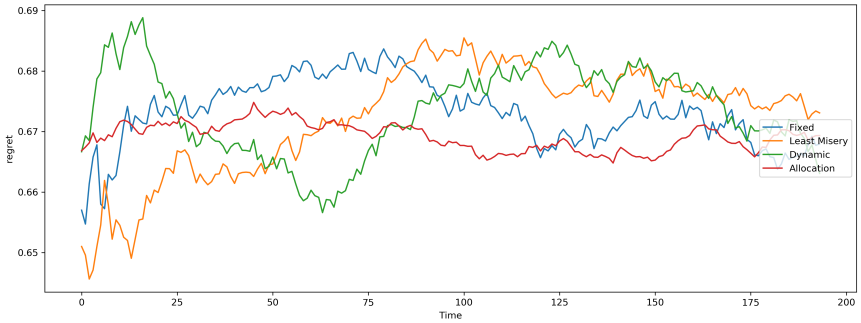
Table 10. Summary results. Fairness measured by percentage of protected item exposure in recommendation lists.

Table 10b shows similar results for the Kiva data set. Here we do not see as much accuracy loss. The Allocation algorithm, which here has the best nDCG, is only 5.5% below the original base algorithm. For this data set, the re-rankers also improve fairness, although not as dramatically as in the MovieLens case. The Allocation method has the highest fairness score in addition to the best nDCG. Figure 16 shows the average fairness regret over time for the experiment. The algorithms all move within a fairly narrow regret bound, indicating the difficulty of achieving fairness in these data sets. The Fixed lottery shows lowest regret over most epochs for the MovieLens data set, but does not do as well with Kiva. Similar inconsistency is shown with the Least Misery algorithm. The low variance of the fairness of Allocation algorithm can be seen, as its regret does not show the swings of the other algorithms.

[10] For the first batch, when no backward window exists, the Fixed Lottery was performed.

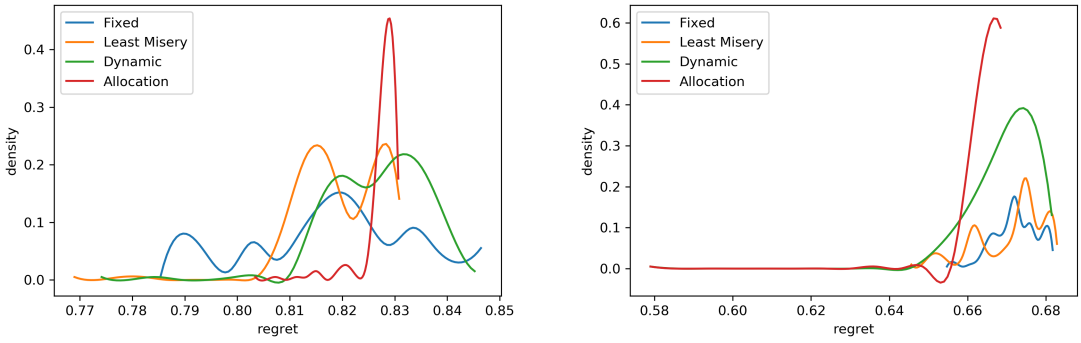


(a) MovieLens data set



(b) Kiva data set

Fig. 16. Average fairness regret over time



(a) MovieLens (b) Kiva

Fig. 17. Distribution of fairness regret

Across both tables and in the time series figure, we see that the Allocation method has lower variance in the fairness it achieves across iterations. Another way to see this consistency is the distribution of the average regret values. Figures 17a and 17b show the distribution of regret for the different choice functions on each data set. The distribution

of the Allocation method (shown in red) falls within a much narrower band than any of the other methods, particularly in the MovieLens data set where we see the Fixed method in blue taking on a wide range of regret values. At any given time, the Allocation algorithm is producing consistently fair results without the large variations in regret seen the other algorithms. As its fairness results are similar to those of the other lottery mechanisms, this consistency is a good reason to prefer it.

4.6 Conclusion and Future Work

In this paper, we conceptualize algorithmic fairness and recommendation fairness, in particular, as a problem of *social choice*. That is, we define the task of computing a recommendation as a problem of arbitrating among the preferences of different individual agents to arrive at a single outcome. For our purposes, the agents in question include the user and also multiple *fairness concerns* that may be active within a particular organization.

The move to frame fairness as a problem of social choice has several important consequences. First, it highlights the multiplicity and diversity of fairness (and other stakeholder) concerns that might be relevant in a given application. This approach allows us to be agnostic to different definitions and metrics of fairness and does not impose any particular structure on stakeholder preferences.

Second, we are able to make use of the large body of research in computational social choice, including the study of fairness, that has emerged in the past decades.

Building on these ideas, we demonstrate the SCRUF framework for dynamic adaptation of recommendation fairness using social choice to arbitrate between different re-ranking methods. We define a set of choice functions, ranging from a simple fixed lottery to an adaptation of the probabilistic serial mechanism, and demonstrate their performance on two data sets where multiple fairness concerns have been defined. We found relatively minor differences between the different lottery mechanisms, except that the Allocation mechanism, which takes user preferences over features into account, provides lower variance in fairness over time and therefore a more consistently fair output.

5 APPROACH AND METHODOLOGY

Despite progress in recent years, reproducibility remains a challenge in recommender systems research [12]. Minor differences in parameters and experimental settings can yield incompatible results, which make it difficult to provide definitive answers about the relative properties of different algorithms. Progress in this area is supported by providing platforms on which comparative experiments can be conducted using declarative experimental configuration (so that experimental settings can be easily shared), with pre-implemented methodological workflows, and with a large library of algorithms for rapid benchmarking.

In this section, I describe `librec-auto`, an open-source command-line Python package providing a wrapper for the well-known LibRec 2.0 recommender systems algorithm library[11]. My contributions to this project were the implementation and addition of in-processing and post-processing fairness algorithms and fairness metrics. The detail of each method is describe in the following section.

5.1 Librec-auto

This tool has been presented in [99] and provides an environment that supports automated experimentation. Therefore it supports reproducibility of research in recommendation algorithms.

[11] www.librec.net

Key advantages of `librec-auto` were its ability to support typical research workflows, to offer a declaration configuration system, and to supplement experiment execution with the scripted production of human-friendly outputs including visualizations.

We have now extended this platform in a number of ways, particularly to support research in fairness-aware recommendation. `librec-auto` now supports the current 3.0 version of the LibRec library and can take advantage of the new algorithms (including deep learning algorithms) found there. The `librec-auto` project has also enhanced LibRec with a suite of metrics for measuring the fairness of recommendation outcomes. Most significantly, the tool now supports recommendation re-ranking, a common approach to enhancing fairness, diversity, and other non-accuracy properties of recommendation outcomes.

5.1.1 ***Core features.***

LibRec 3.0 is a Java-based recommendation generation platform. It has been available to the recommender systems community since 2015 [62], and has large library of implemented recommendation algorithms (more than 70 as of this writing). The platform supports a variety of evaluation metrics and evaluation methodologies. However, our experience indicates that for practical experimentation and reproducibility research, LibRec by itself is not sufficient. For example, intermediate computational outputs, such as recommendation results, cannot be reused as input for new evaluation metrics, requiring the re-execution of potentially lengthy experiment executions.

We developed `librec-auto`[12] to retain the benefits of working with LibRec while adding support for experimentation. A sketch of the functionality of `librec-auto` is provided in Figure 18. As the figure indicates, LibRec is encapsulated and its various component elements are used to execute particular portions of the experimental workflow. In addition, the optional re-ranking component allows for the study of re-ranking algorithms not within the scope of LibRec's design. The key inputs are data and an XML-based configuration file. A particular study may consist of multiple experiments, all of which are configured at the same time in the configuration file. Configuration files are modular, so that, for example, multiple studies can share the same methodology elements, preventing inadvertent misconfiguration.

Although the figure indicates a straight-line of execution, parallelism is built into `librec-auto` at the level of experiment execution. Because experiments can have lengthy execution times, the post-processing phase allows for integration with messaging platforms, including Slack, so that experimenters are notified when their tasks are complete. These messages can include visualizations of experimental output, to provide a quick overview of results.

5.1.2 ***Fairness-aware extensions.***

Although `librec-auto` has been under development since 2018, the latest release incorporates several key advances that specifically support common tasks in the study of recommendation fairness. These advances are (1) new evaluation metrics that report on fairness aspects of recommendation output, (2) an optional re-ranking step in the experiment pipeline, to support what is one of the most common category of fairness enhancing techniques, and (3) additional support for working with user (demographic) and item (content) features in algorithms and metrics. With these features, `librec-auto` now can support a wide range of research activities in fairness-aware recommendation, and we will be adding additional capabilities in future releases.

Previously in the literature, many re-ranking algorithms have been proposed to achieve a balance between diversity and accuracy. The following methods try to achieve a fair representation between groups by penalizing the score of over-represented groups or reinforcing the score of the under-represented groups: (1) **FAR**, defined in [97], combines a

[12] github.com/that-recsys-lab/librec-auto

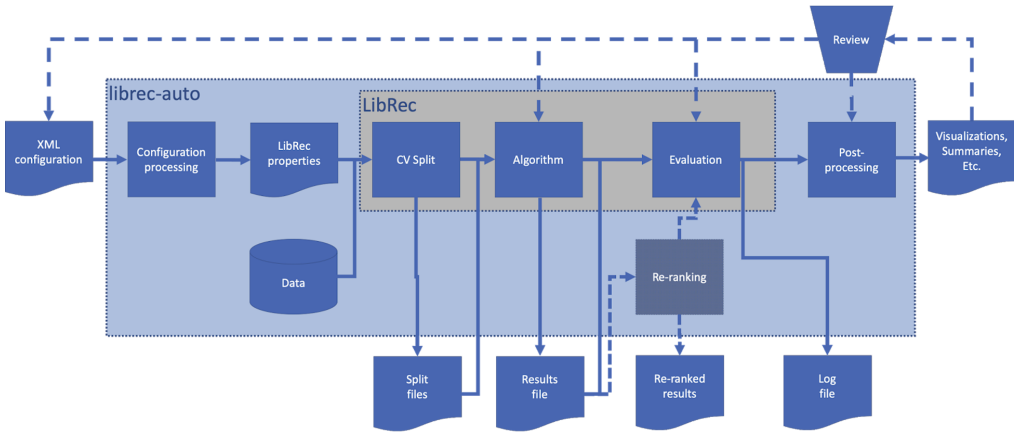


Fig. 18. Schematic of experimentation workflow with `librec-auto`. The LibRec library (Java, shown in grey) is encapsulated by `librec-auto` (Python, shown in blue), which manages configuration, experimental outputs and post-processs. Added from [99] is the new re-ranking module shown in dark blue.

personalization-induced and fairness-induced scores with hyper-parameter λ; (2) **PFAR**, from [97], adds a personalized weight to FAR, calculated based on item-features in user profile, representing the tolerance of the user for diverse resultsl and (3) **OFAiR** incorporates similar personalization and allows fine-grained control of protected group promotion when there are multiple protected groups [126]. By contrast, (4) **FA*IR** [145] builds a queues of protected and unprotected items and draws from each queue to build the final re-ranked list. We also include two more general diversity-enhancing re-rankers first promoted in the information retrieval literature: (5) **MMR** diversifies result lists by greedily adding items with maximal marginal relevance [32], and (6) **XQuAD** defined in [121] has similar goal to MMR algorithm, but it enhances diversity with respect to specific aspects. Finally, we include (7) **Calibrated Recommendations**, an algorithm closely tied to the Calibration metric above, which re-ranks recommendations to ensure a close match to the user's distribution of interests in item features [127]. The re-ranking methods are part of `librec-auto` and are implemented in Python.

Recommendation fairness and associated fairness metrics can be defined from the perspective of two main stakeholders: providers and consumers [25]. Additionally, both provider-side and consumer-side metrics come in two basic varieties: exposure-based and hit-based. *Exposure* metrics focus on the the appearance of protected items [124] in a ranked list and *hit-based* metrics take into account the suitability of the target user [2]. Many metrics have been offered to measure recommendation fairness [14, 18, 33, 87, 127, 136, 139, 140]. We implement the following metrics in `librec-auto` and where possible, both consumer-side and item-side versions of the metric are available: (1) **Discounted Proportional Fairness** (DPF), a hit-based fairness metric similar to the metric offered in [33] where it measures the ranking utility (nDCG) of the protected group with respect to the other groups. (2) **Calibration** [127], a distribution-based metric that uses KL-Divergence to measure the difference in item category distribution between the preferences of users and their respective recommendation lists. (3) **Statistical parity**, based on the ideas discussed in [114, 146], measuring the difference in outcomes between protected and unprotected groups relative to various recommendation outcomes. Both ranking and prediction accuracy measures are supported. (4) **P-Percent-Rule** (PPR) discussed in [16], is a two-sided extension of statistical parity [11]. (5) **Error-based** metrics proposed in Yao et al. [140] including value-unfairness, absolute unfairness, underestimation unfairness, overestimation unfairness, and non-parity

unfairness by Kamishima et al. [78]. Additionally, we offer the following diversity-based metrics (6) **Intra-list distance (ILD)** [154], a pairwise distance between all the item features in each user's recommendation list, and (7) **Gini Index** calculated over the exposure of all the present groups in the recommendation list. All metrics are implemented in Java and integrated with the LibRec code base.

We plan future releases of `librec-auto` to include integration with additional recommendation libraries, including LKPY [48], LibFM [110], and DeepRec [149].

6 PROPOSED WORK AND TIMETABLE

Overall, the presented methods in previous sections were looking to find a balance between accuracy and fairness either using a re-ranking method or a regularization based method. Most of the focus of the previous methods were on reaching provider-side fairness. Here are the next steps for the projects that are introduced earlier and are going to be extended on my thesis by December 2021.

- **A Survey of Fairness in Recommender Systems**: Research on recommender systems fairness usually revolves around new algorithms. we are in the process of producing a journal article that surveys existing algorithms and compares their fairness properties on several data sets, from the perspective of both recommendation consumers and item providers.
  We examine prominent examples from three different classes198of algorithms: neighborhood-based, factorization using prediction loss, and factorization using ranking loss. From the neighborhood based algorithms we picked Sparse Linear Methods (SLIM) and item-based kNN. From the Matrix Factorization family using prediction loss we picked Biased Matrix Factorization, Weighted Regularized Matrix Factorization, Non-Negative Matrix Factorization, and from the factorization methods that use ranking loss we picked Bayesian Personalized Ranking. We are testing their fairness properties on the following datasets: The Movies Dataset, MovieLens 1M, Kiva dataset and Lastfm.
  I plan for these experiments and implementation to take about 2 months. This work is intended to be submitted as a journal article.
- **Fairness through Balanced Neighborhoods**: In this project, we built on the standard nearest neighbor techniques in recommender systems and built balanced neighborhoods to ensure diversity among the peers from whom recommendations are generated. In our future work for this project, we plan to extend these findings in several ways. We would like to have a more extensive experimentation of the fairness properties of the balanced neighborhood SLIM. And we would like to run these experimentation for both consumers and providers. We would also like to test this idea on different neighborhood-based methods besides SLIM.
  It is possible that a multisided platform may require fairness be considered for both consumers and providers at the same time: a CP-fairness condition. For example, a rental property recommender may treat minority applicants as a protected class and wish to ensure that they are recommended properties similar to unprotected renters. At the same time, the recommender may wish to treat minority landlords as a protected class and ensure that highly-qualified tenants are referred to them at the same rate as to landlords who are not in the protected class. One important question for future research is how the outcomes for each stakeholder and the overall system performance are affected by combining consumer- and provider-fairness concerns.
  Finally, we expect to publish a journal article of these thorough experiments in the Information and Management Journal. I plan to spend 2 month on the implementation and experiments of this project.

- **Fairness in Dynamic Recommender Systems**: In this paper, we conceptualized algorithmic fairness and recommendation fairness, in particular, as a problem of *social choice*. That is, we define the task of computing a recommendation as a problem of arbitrating among the preferences of different individual agents to arrive at a single outcome. For our purposes, the agents in question include the user and also multiple *fairness concerns* that may be active within a particular organization.

  The most important consequence of framing fairness as a problem of social choice is that it highlights the multiplicity and diversity of fairness (and other stakeholder) concerns that might be relevant in a given application. This approach allows us to be agnostic to different definitions and metrics of fairness and does not impose any particular structure on stakeholder preferences. The other important consequence is that we can use the extensive body of research on fairness in this field.

  We build the SCRUFF framework for dynamic adaptation of recommendation fairness using social choice to arbitrate between different re-ranking methods. We defined a set of choice functions, ranging from a simple fixed lottery to an adaptation of the probabilistic serial mechanism, and demonstrate their performance on two data sets where multiple fairness concerns have been defined. However, we found relatively minor differences between the different lottery mechanisms, except that the Allocation mechanism, which takes user preferences over features into account, provides lower variance in fairness over time and therefore a more consistently fair output.

  In this regard, there are many aspects and variations to the experiments in this framework. For example, the definition of fairness can be measured through various metrics that I have not explored them completely. An appropriate metric such as Generalized Cross-entropy would help us better show the differences in performance of the various methods. Also, since our method works on top of a base recommendation, the choice of the recommendation algorithm may have a great impact on the final result. In my previous experiments I did not explore these methods thoroughly. Another extension for this work would be to add some constraint in the fairness objective to avoid increasing fairness for aspects that have already gained enough exposure. One approach for such constraints can be achieved through Hinge loss. Finally, in the current work fairness may not always be guaranteed in any given period of time. New objectives can be defined to overcome this problem and I am going to explore them as another extension.

  I plan for these experiments to take about two to three months. This Project is intended to be submitted to either the ACM Conference Series on Recommender Systems 2021 or the ACM conference on Fairness, Accountability, and Transparency 2021.

I plan to spend two month on writing the thesis and eventually defend it on December 2021.

REFERENCES

[1] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Pizzato. [n.d.]. Multistakeholder recommendation: Survey and research directions. *User Modeling and User-Adapted Interaction* ([n. d.]), 1–32. https://doi.org/10.1007/s11257-019-09256-1

[2] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Pizzato. 2020. Multistakeholder recommendation: Survey and research directions. *User Modeling and User-Adapted Interaction* 30, 1 (2020), 127–158.

[3] 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*.

[4] Gediminas Adomavicius and Y Kwon. 2009. Improving recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering* 10 (2009).

[5] Gediminas Adomavicius and YoungOk Kwon. 2011. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering* 24, 5 (2011), 896–911.

[6] Muhammad Ali, Piotr Sapiezynski, Miranda Bogen, Aleksandra Korolova, Alan Mislove, and Aaron Rieke. 2019. Discrimination through Optimization: How Facebook's Ad Delivery Can Lead to Biased Outcomes. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 199 (Nov. 2019), 30 pages. https://doi.org/10.1145/3359301

[7] Haris Aziz. 2019. Developments in Multi-Agent Fair Allocation. *CoRR* abs/1911.09852 (2019). arXiv:1911.09852 http://arxiv.org/abs/1911.09852

[8] Haris Aziz, Jiashu Chen, Aris Filos-Ratsikas, Simon Mackenzie, and Nicholas Mattei. 2015. Egalitarianism of Random Assignment Mechanisms. *CoRR* abs/1507.06827 (2015). arXiv:1507.06827 http://arxiv.org/abs/1507.06827

[9] Haris Aziz, Serge Gaspers, Simon Mackenzie, Nicholas Mattei, Nina Narodytska, and Toby Walsh. 2015. Equilibria Under the Probabilistic Serial Rule. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Qiang Yang and Michael J. Wooldridge (Eds.). AAAI Press, 1105–1112. http://ijcai.org/Abstract/15/160

[10] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2018. Fairness and machine learning. 1 (2018), 23. http://www.fairmlbook.org

[11] Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *Calif. L. Rev.* 104 (2016), 671.

[12] Joeran Beel, Corinna Breitinger, Stefan Langer, Andreas Lommatzsch, and Bela Gipp. 2016. Towards reproducibility in recommender-systems research. *User modeling and user-adapted interaction* 26, 1 (2016), 69–101.

[13] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. 2017. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409* (2017).

[14] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2212–2220.

[15] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. 2017. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075* (2017).

[16] Dan Biddle. 2006. *Adverse impact and test validation: A practitioner's guide to valid and defensible employment testing*. Gower Publishing, Ltd.

[17] Asia J Biega, Fernando Diaz, Michael D Ekstrand, and Sebastian Kohlmeier. 2020. Overview of the trec 2019 fair ranking track. *arXiv preprint arXiv:2003.11650* (2020).

[18] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. biega2018equity. In *The 41st international acm sigir conference on research & development in information retrieval*. 405–414.

[19] Anna Bogomolnaia and Hervé Moulin. 2001. A new solution to the random assignment problem. *Journal of Economic theory* 100, 2 (2001), 295–328.

[20] Engin Bozdag. 2013. Bias in algorithmic filtering and personalization. *Ethics and Information Technology* 15, 3 (Sept. 2013), 209–227. https://doi.org/10.1007/s10676-013-9321-6

[21] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia (Eds.). 2016. *Handbook of Computational Social Choice*. Cambridge University Press.

[22] Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. 2013. Designing random allocation mechanisms: Theory and applications. *American economic review* 103, 2 (2013), 585–623.

[23] Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*. 77–91.

[24] Robin Burke. 2017. Multisided Fairness for Recommendation. In *Workshop on Fairness, Accountability and Transparency in Machine Learning (FATML)*. Halifax, Nova Scotia.

[25] Robin Burke. 2017. Multisided Fairness for Recommendation. , 5 pages. arXiv:1707.00093 [cs.CY]

[26] Robin Burke and Himan Abdollahpouri. 2017. Patterns of Multistakeholder Recommendation. *arXiv preprint arXiv:1707.09258* (2017).

[27] Robin Burke, Jackson Kontny, and Nasim Sonboli. 2018. Synthetic attribute data for evaluating consumer-side fairness. *arXiv preprint arXiv:1809.04199* (2018).

[28] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. 2018. Balanced Neighborhoods for Multi-sided Fairness in Recommendation. In *Conference on Fairness, Accountability and Transparency*. 202–214.

[29] Robin Burke, Amy Voida, Nicholas Mattei, and Nasim Sonboli. 2020. Algorithmic Fairness, Institutional Logics, and Social Choice. In *Harvard CRCS Workshop: AI for Social Good*.

[30] Burke, Robin. 2017. Multisided Fairness for Recommendation. In *Workshop on Fairness, Accountability and Transparency in Machine Learning (FATML)*. arXiv, Halifax, Nova Scotia, Article arXiv:1707.00093 [cs.CY], 5 pages.

[31] Rocío Cañamares and Pablo Castells. 2018. Should I Follow the Crowd? A Probabilistic Analysis of the Effectiveness of Popularity in Recommender Systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 415–424. https://doi.org/10.1145/3209978.3210014

[32] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 335–336.

[33] Carlos Castillo. 2019. Fairness and transparency in ranking. In *ACM SIGIR Forum*, Vol. 52. ACM New York, NY, USA, 64–71.

[34] Òscar Celma and Pedro Cano. 2008. From hits to niches?: or how popular artists can bias music recommendation and discovery. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*. ACM, 5.

[35] Abhijnan Chakraborty, Gourab K Patro, Niloy Ganguly, Krishna P Gummadi, and Patrick Loiseau. 2019. Equality of voice: Towards fair representation in crowdsourced top-k recommendations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 129–138.

[36] Allison J. B. Chaney, Brandon M. Stewart, and Barbara E. Engelhardt. 2018. How Algorithmic Confounding in Recommendation Systems Increases Homogeneity and Decreases Utility. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 224–232. https://doi.org/10.1145/3240323.3240370

[37] Irene Y. Chen, Fredrik D. Johansson, and David Sontag. 2018. Why is My Classifier Discriminatory?. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 3543–3554.

[38] Juanyi Chen, Amber Y Chang, and Garry D Bruton. 2017. Microfinance: Where are we today and where should the research go in the future? *International Small Business Journal* 35, 7 (2017), 793–802.

[39] Jaegul Choo, Changhyun Lee, Daniel Lee, Hongyuan Zha, and Haesun Park. 2014. Understanding and promoting micro-finance activities in kiva.org. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 583–592.

[40] Jaegul Choo, Daniel Lee, Bistra Dilkina, Hongyuan Zha, and Haesun Park. 2014. To gather together for a better world: Understanding and leveraging communities in micro-lending recommendation. In *Proceedings of the 23rd international conference on World wide web*. ACM, 249–260.

[41] Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5, 2 (2017), 153–163.

[42] Sam Corbett-Davies and Sharad Goel. 2018. The measure and mismeasure of fairness: A critical review of fair machine learning. *CoRR* (2018). https://doi.org/abs/1808.00023

[43] Kate Crawford. 2017. The trouble with bias. In *Conference on Neural Information Processing Systems (NIPS)*.

[44] Anubrata Das and Matthew Lease. 2019. A Conceptual Framework for Evaluating Fairness in Search. *arXiv preprint arXiv:1907.09328* (2019).

[45] Fernando Diaz, Bhaskar Mitra, Michael D. Ekstrand, Asia J. Biega, and Ben Carterette. 2020. *Evaluating Stochastic Rankings with Expected Exposure*. Association for Computing Machinery, New York, NY, USA, 275–284. https://doi.org/10.1145/3340531.3411962

[46] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, Cambridge, Massachusetts, 214–226.

[47] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through Awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference* (Cambridge, Massachusetts) *(ITCS '12)*. Association for Computing Machinery, New York, NY, USA, 214–226. https://doi.org/10.1145/2090236.2090255

[48] Michael D Ekstrand. 2018. The LKPY package for recommender systems experiments: Next-generation tools and lessons learned from the LensKit project. *arXiv preprint arXiv:1809.03125* (2018).

[49] 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 *Conference on Fairness, Accountability and Transparency*. PMLR, 172–186.

[50] 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 *Proceedings of the 1st Conference on Fairness, Accountability and Transparency (Proceedings of Machine Learning Research, Vol. 81)*, Sorelle A. Friedler and Christo Wilson (Eds.). PMLR, New York, NY, USA, 172–186.

[51] Michael D Ekstrand, Mucun Tian, Mohammed R Imran Kazi, Hoda Mehrpouyan, and Daniel Kluver. 2018. Exploring author gender in book rating and recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 242–250.

[52] Farzad Eskandanian, Bamshad Mobasher, and Robin Burke. [n.d.]. User Segmentation for Controlling Recommendation Diversity. In *Poster Proceedings of the 10th ACM Conference on Recommender Systems*.

[53] Farzad Eskandanian, Bamshad Mobasher, and Robin Burke. 2017. A Clustering Approach for Personalizing Diversity in Collaborative Recommender Systems. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, ACM, New York, 280–284.

[54] Farzad Eskandanian, Bamshad Mobasher, and Robin Burke. 2017. A Clustering Approach for Personalizing Diversity in Collaborative Recommender Systems. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization* (Bratislava, Slovakia) *(UMAP '17)*. Association for Computing Machinery, New York, NY, USA, 280–284. https://doi.org/10.1145/3079628.3079699

[55] Farzad Eskandanian, Nasim Sonboli, and Bamshad Mobasher. 2019. Power of the Few: Analyzing the Impact of Influential Users in Collaborative Recommender Systems. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization* (Larnaca, Cyprus) *(UMAP '19)*. Association for Computing Machinery, New York, NY, USA, 225–233. https://doi.org/10.1145/3320435.3320464

[56] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Sydney, NSW, Australia) *(KDD '15)*. Association for Computing Machinery, New York, NY, USA, 259–268. https://doi.org/10.1145/2783258.2783311

[57] Andres Ferraro. 2019. Music Cold-Start and Long-Tail Recommendation: Bias in Deep Representations. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 586–590. https://doi.org/10.1145/3298689.3347052

[58] Benjamin Fish, Jeremy Kun, and Ádám D Lelkes. 2016. A confidence-based approach for balancing fairness and accuracy. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 144–152.

[59] Sorelle A. Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P. Hamilton, and Derek Roth. 2019. A Comparative Study of Fairness-Enhancing Interventions in Machine Learning *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 329–338. https://doi.org/10.1145/3287560.3287589

[60] Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. 2007. Pathwise coordinate optimization. *The Annals of Applied Statistics* 1, 2 (Dec. 2007), 302–332. https://doi.org/10.1214/07-AOAS131

[61] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2221–2231. https://doi.org/10.1145/3292500.3330691

[62] Guibing Guo, Jie Zhang, Zhu Sun, and Neil Yorke-Smith. 2015. LibRec: A Java Library for Recommender Systems. In *UMAP Extended Proceedings*. CEUR, Dublin, Ireland, Article 9, 4 pages.

[63] Sara Hajian and Josep Domingo-Ferrer. 2012. A methodology for direct and indirect discrimination prevention in data mining. *IEEE transactions on knowledge and data engineering* 25, 7 (2012), 1445–1459.

[64] Moritz Hardt, Eric Price, Nati Srebro, et al. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*. 3315–3323.

[65] F Maxwell Harper and Joseph A Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 4, Article 19 (2015), 19 pages.

[66] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. 2018. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*. PMLR, 1929–1938.

[67] Úrsula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. 2018. Multicalibration: Calibration for the (Computationally-identifiable) masses. In *International Conference on Machine Learning*. 1944–1953.

[68] Rong Hu and Pearl Pu. 2011. Enhancing recommendation diversity with organization interfaces. In *Proceedings of the 16th international conference on Intelligent user interfaces*. 347–350.

[69] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. Ieee, 263–272.

[70] Michael Jahrer and Andreas Töscher. 2012. Collaborative Filtering Ensemble for Ranking. In *Proceedings of KDD Cup 2011 (Proceedings of Machine Learning Research, Vol. 18)*. PMLR, 153–167.

[71] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[72] Nathan Kallus, Xiaojie Mao, and Angela Zhou. 2020. Assessing Algorithmic Fairness with Unobserved Protected Class Using Data Combination. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) *(FAT* '20)*. Association for Computing Machinery, New York, NY, USA, 110. https://doi.org/10.1145/3351095.3373154

[73] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. 2010. Discrimination aware decision tree learning. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, University of Technology Sydney, Australia, 869–874.

[74] Toshihiro Kamishima and Shotaro Akaho. 2017. Considerations on recommendation independence for a find-good-items task. (2017).

[75] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases*. Springer, Heidelberg, Germany, 35–50.

[76] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2018. Recommendation Independence. In *Conference on Fairness, Accountability and Transparency (Proceedings of Machine Learning Research, Vol. 81)*, Sorelle A. Friedler and Christo Wilson (Eds.). PMLR, New York, NY, USA, 187–201.

[77] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Issei Sato. 2016. Model-based approaches for independence-enhanced recommendation. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, IEEE, New York, 860–867.

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

[79] Mesut Kaya, Derek Bridge, and Nava Tintarev. 2020. Ensuring Fairness in Group Recommendations by Rank-Sensitive Balancing of Relevance. In *Fourteenth ACM Conference on Recommender Systems* (Virtual Event, Brazil) *(RecSys '20)*. Association for Computing Machinery, New York, NY, USA, 101–110. https://doi.org/10.1145/3383313.3412232

[80] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2017. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. *arXiv preprint arXiv:1711.05144* (2017).

[81] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2019. An empirical study of rich subgroup fairness for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 100–109.

[82] Michael Kearns, Aaron Roth, and Zhiwei Steven Wu. 2017. Meritocratic Fairness for Cross-Population Selection. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, International Convention Centre, Sydney, Australia, 1828–1836. http://proceedings.mlr.press/v70/kearns17a.html

[83] Frank P Kelly, Aman K Maulloo, and David KH Tan. 1998. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society* 49, 3 (1998), 237–252.

[84] kiva. 2018. Kiva - Loans that change lives. https://www.kiva.org/

[85] Jon Kleinberg and Manish Raghavan. 2018. Selection Problems in the Presence of Implicit Bias. , 38 pages. arXiv:1801.03533 [cs.CY]

[86] Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA (LIPIcs, Vol. 67)*, Christos H. Papadimitriou (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 43:1–43:23. https://doi.org/10.4230/LIPIcs.ITCS.2017.43

[87] Caitlin Kuhlman, MaryAnn VanValkenburg, and Elke Rundensteiner. 2019. FARE: Diagnostics for Fair Ranking using Pairwise Error Metrics. In *The World Wide Web Conference*. 2936–2942.

[88] Matevž Kunaver and Tomaž Požrl. 2017. Diversity in recommender systems–A survey. *Knowledge-Based Systems* 123 (2017), 154–162.

[89] Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. 2019. ifair: Learning individually fair data representations for algorithmic decision making. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 1334–1345.

[90] Eric L Lee, Jing-Kai Lou, Wei-Ming Chen, Yen-Chi Chen, Shou-De Lin, Yen-Sheng Chiang, and Kuan-Ta Chen. 2014. Fairness-aware loan recommendation for microfinance services. In *Proceedings of the 2014 International Conference on Social Computing*. ACM, Beijing, China, 3.

[91] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, et al. 2019. WeBuildAI: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–35.

[92] J. W. Lian, N. Mattei, R. Noble, and T. Walsh. 2018. The Conference Paper Assignment Problem: Using Order Weighted Averages to Assign Indivisible Goods. In *32nd*.

[93] Kun Lin, Nasim Sonboli, Bamshad Mobasher, and Robin Burke. 2019. Crank up the volume: preference bias amplification in collaborative recommendation. , 9 pages. arXiv:1909.06362 [cs.IR]

[94] Kun Lin, Nasim Sonboli, Bamshad Mobasher, and Robin Burke. 2020. Calibration in Collaborative Filtering Recommender Systems: A User-Centered Analysis. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media* (Virtual Event, USA) *(HT '20)*. Association for Computing Machinery, New York, NY, USA, 197–206. https://doi.org/10.1145/3372923.3404793

[95] Weiwen Liu and Robin Burke. 2018. Personalizing Fairness-aware Re-ranking. , 6 pages. arXiv:1809.02921 [cs.IR]

[96] 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.

[97] 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* (Copenhagen, Denmark) *(RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 467–471. https://doi.org/10.1145/3298689.3347016

[98] David F. Manlove. 2013. *Algorithmics of Matching Under Preferences*. Series on Theoretical Computer Science, Vol. 2. WorldScientific. https://doi.org/10.1142/8591

[99] 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*. 500–501.

[100] Nicholas Mattei. 2020. Closing the Loop: Bringing Humans into Empirical Computational Social Choice and Preference Reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, Christian Bessiere (Ed.). ijcai.org, 5169–5173. https://doi.org/10.24963/ijcai.2020/729

[101] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. 2018. Towards a Fair Marketplace: Counterfactual Evaluation of the Trade-off between Relevance, Fairness & Satisfaction in Recommendation Systems *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 2243–2251. https://doi.org/10.1145/3269206.3272027

[102] Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. 2021. Algorithmic Fairness: Choices, Assumptions, and Definitions. *Annual Review of Statistics and Its Application* 8 (2021).

[103] Natwar Modani, Deepali Jain, Ujjawal Soni, Gaurav Kumar Gupta, and Palak Agarwal. 2017. Fairness Aware Recommendations on Behance. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 144–155.

[104] Hervé Moulin. 2004. *Fair division and collective welfare*. MIT press.

[105] Xia Ning and George Karypis. 2011. SLIM: Sparse linear methods for top-n recommender systems. In *11th IEEE International Conference on Data Mining (ICDM)* (Vancouver, BC, Canada). IEEE, New York, NY, USA, 497–506.

[106] Eli Pariser. 2011. *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin, New York, NY, USA.

[107] 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.

[108] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, New York, NY, USA, 560–568.

[109] J. Rawls. 1971. *A Theory of Justice*. Harvard University Press.

[110] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 1–22.

[111] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[112] Nicholas Rescher. 2002. *Fairness: Theory and practice of distributive justice*. Transaction Publishers.

[113] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.

[114] Ya'acov Ritov, Yuekai Sun, and Ruofei Zhao. 2017. On conditional parity as a notion of non-discrimination in machine learning. *arXiv preprint arXiv:1706.08519* (2017).

[115] Lior Rokach and Oded Maimon. 2005. Clustering methods. In *Data mining and knowledge discovery handbook*. Springer, 321–352.

[116] Andrea Romei and Salvatore Ruggieri. 2014. A multidisciplinary survey on discrimination analysis. *The Knowledge Engineering Review* 29, 5 (2014), 582–638.

[117] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. 2005. Pairwise kidney exchange. *J. Econ. Theory* 125, 2 (2005), 151–188. https://doi.org/10.1016/j.jet.2005.04.004

[118] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

[119] Alan Said, Brijnesh J Jain, Sascha Narr, Till Plumbaum, Sahin Albayrak, and Christian Scheel. 2012. Estimating the magic barrier of recommender systems: a user study. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1061–1062.

[120] Rodrygo LT Santos, Craig Macdonald, Iadh Ounis, et al. 2015. Search result diversification. *Foundations and Trends® in Information Retrieval* 9, 1 (2015), 1–90.

[121] Rodrygo LT Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. 2010. Explicit search result diversification through sub-queries. In *European conference on information retrieval*. Springer, 87–99.

[122] Amartya Sen. 2018. *Collective Choice and Social Welfare*. Harvard University Press.

[123] Claude Elwood Shannon. 2001. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review* 5, 1 (2001), 3–55.

[124] 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*. 2219–2228.

[125] 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* (Genoa, Italy) *(UMAP '20)*. Association for Computing Machinery, New York, NY, USA, 239–247. https://doi.org/10.1145/3340631.3394846

[126] 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*. to appear.

[127] Harald Steck. 2018. Calibrated recommendations. In *Proceedings of the 12th ACM conference on recommender systems*. 154–162.

[128] Tom Sühr, Asia J Biega, Meike Zehlike, Krishna P Gummadi, and Abhijnan Chakraborty. 2019. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3082–3092.

[129] Wenlong Sun, Sami Khenissi, Olfa Nasraoui, and Patrick Shafto. 2019. Debiasing the human-recommender system feedback loop in collaborative filtering. In *Companion Proceedings of The 2019 World Wide Web Conference*. 645–651.

[130] Cass R Sunstein. 2009. *Republic. com 2.0*. Princeton University Press, Princeton, NJ.

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

[132] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. 2008. Investigation of various matrix factorization methods for large recommender systems. In *2008 IEEE International Conference on Data Mining Workshops*. IEEE, 553–562.

[133] William Thomson. 2011. Fair allocation rules. In *Handbook of Social Choice and Welfare*. Vol. 2. Elsevier, 393–506.

[134] William Thomson. 2016. Introduction to the Theory of Fair Allocation. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, 261–283. https://doi.org/10.1017/CBO9781107446984.012

[135] Nava Tintarev, Matt Dennis, and Judith Masthoff. 2013. Adapting recommendation diversity to openness to experience: A study of human behaviour. In *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 190–202.

[136] Virginia Tsintzou, Evaggelia Pitoura, and Panayiotis Tsaparas. 2018. Bias disparity in recommendation systems. *arXiv preprint arXiv:1811.01461* (2018).

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

[138] Alice Xiang and Inioluwa Deborah Raji. 2019. On the legal compatibility of fairness definitions. *arXiv preprint arXiv:1912.00761* (2019).

[139] Ke Yang and Julia Stoyanovich. 2017. Measuring fairness in ranked outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*. 1–6.

[140] Sirui Yao and Bert Huang. 2017. Beyond parity: Fairness objectives for collaborative filtering. In *Advances in Neural Information Processing Systems*. Article CoRR abs/1705.08804, 10 pages.

[141] Sirui Yao and Bert Huang. 2017. New Fairness Metrics for Recommendation that Embrace Differences. In *Workshop on Fairness, Accountability and Transparency in Machine Learning (FATML)*. arXiv:1706.09838, Halifax, Nova Scotia, Article arXiv:1706.09838, 5 pages.

[142] H. Peyton Young. 1995. *Equity - in theory and practice*. Princeton University Press.

[143] Muhammad Yunus. 1998. *Banker to the Poor*. Penguin Books India.

[144] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. 2017. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1171–1180. https://doi.org/10.1145/3038912.3052660

[145] 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.

[146] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. ACM, Atlanta, GA, USA, 325–333.

[147] Lu Zhang and Xintao Wu. 2017. Anti-discrimination learning: a causal modeling-based framework. *International Journal of Data Science and Analytics* 4, 1 (2017), 1–16.

[148] Mi Zhang and Neil Hurley. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*. ACM, 123–130.

[149] Shuai Zhang, Yi Tay, Lina Yao, Bin Wu, and Aixin Sun. 2019. Deeprec: An open-source toolkit for deep learning based recommendation. *arXiv preprint arXiv:1905.10536* (2019).

[150] Xueru Zhang, Mohammad Mahdi Khalili, Cem Tekin, and Mingyan Liu. 2019. Group retention when using machine learning in sequential decision making: the interplay between user dynamics and fairness. *arXiv preprint arXiv:1905.00569* (2019).

[151] Yong Zheng, Bamshad Mobasher, and Robin Burke. 2014. CSLIM: Contextual SLIM recommendation algorithms. In *Proceedings of the 8th ACM Conference on Recommender Systems*. ACM, Silicon Valley, CA, USA, 301–304.

[152] Ziwei Zhu, Xia Hu, and James Caverlee. 2018. Fairness-Aware Tensor-Based Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1153–1162.

[153] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving Recommendation Lists Through Topic Diversification. In *Proceedings of the 14th International Conference on World Wide Web* (Chiba, Japan) *(WWW '05)*. ACM, New York, NY, USA, 22–32. https://doi.org/10.1145/1060745.1060754

[154] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. ACM, 22–32.

[155] Indre Zliobaite. 2015. A survey on measuring indirect discrimination in machine learning. *arXiv preprint arXiv:1511.00148* (2015).

[156] William S. Zwicker. 2016. Introduction to the Theory of Voting. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, 23–56. https://doi.org/10.1017/CBO9781107446984.003