Mitigating Mainstream Bias in Recommendation via Cost-sensitive Learning

Roger Zhe Li
Z.Li-9@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Julián Urbano
J.Urbano@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Alan Hanjalic
A.Hanjalic@tudelft.nl
Delft University of Technology
Delft, The Netherlands

ABSTRACT

Mainstream bias, where some users receive poor recommendations because their preferences are uncommon or simply because they are less active, is an important aspect to consider regarding fairness in recommender systems. Existing methods to mitigate mainstream bias do not explicitly model the importance of these non-mainstream users or, when they do, it is in a way that is not necessarily compatible with the data and recommendation model at hand. In contrast, we use the recommendation utility as a more generic and implicit proxy to quantify mainstreamness, and propose a simple user-weighting approach to incorporate it into the training process while taking the cost of potential recommendation errors into account. We provide extensive experimental results showing that quantifying mainstreamness via utility is better able at identifying non-mainstream users, and that they are indeed better served when training the model in a cost-sensitive way. This is achieved with negligible or no loss in overall recommendation accuracy, meaning that the models learn a better balance across users. In addition, we show that research of this kind, which evaluates recommendation quality at the individual user level, may not be reliable if not using enough interactions when assessing model performance.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommender Systems, Mainstream Bias, Bias Mitigation

ACM Reference Format:
Roger Zhe Li, Julián Urbano, and Alan Hanjalic. 2023. Mitigating Mainstream Bias in Recommendation via Cost-sensitive Learning. In *Proceedings of the 2023 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '23), July 23, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3578337.3605134

1 INTRODUCTION

One of the critical limitations of recommender systems based on collaborative filtering (CF) models [5] is that they are *not fair* in how they serve different groups of users [9, 11]. This fairness issue is a result of the varying quality of users' neighborhoods (groups of users with similar preferences) from which information is taken to train a CF model [10, 24]. The information collected from large, coherent, and information-rich neighborhoods will be the dominant one in steering the process of learning to recommend for all users. We refer to such dominant neighborhoods as *mainstream*. Because the users belonging to such neighborhoods —the *mainstream users*— are compatible with the learned model, they are optimally served. For the *non-mainstream* users, e.g. *niche* groups who deviate from the mainstream and whose interaction information is therefore less rich [11], who are less active compared to the mainstream users [15], or where the preferences are not well pronounced, the neighborhoods cannot fully reflect their genuine preferences. All this will make the non-mainstream users receive recommendations of a lower quality than the mainstream users. The difference in the quality of the CF model for these two user groups, further referred to as the *mainstream bias*, will result in the continuous improvement of the performance for the mainstream group, and continuous decrease of the performance for the rest [12].

While the issue of treating users differently by a recommender system in general has been addressed by a number of approaches, making for example assumptions about the relation between users' gender [14] or demographics [4] and the quality of recommendation, not many approaches have focused specifically on addressing the mainstream bias. Li et al. [10] deployed an autoencoder [20] for feature reconstruction as an adversary to a traditional CF model, forcing it to deviate from the pure similarity-based learning and make the learned model more compatible with the non-mainstream users. More specifically, the autoencoder was deployed to steer the process of learning the user/item representation space for rating prediction via optimal reconstruction of the properties of all users, mainstream and otherwise, assuming this would lead to equal treatment of users during recommendation. Still, a more explicit focus on the mainstreamness of users is needed to ensure that the bias is effectively addressed.

Inspired by outlier detection techniques, Zhu and Caverlee [24] did focus on explicitly quantifying mainstreamness via similarities of user-preference profiles, and incorporated them to fine-tune the recommendation process for different user groups. However, in the absence of ground truth data about mainstreamness, it is difficult to assess how well these approaches identify non-mainstream users. In addition, these mainstreamness statistics are model-agnostic in the sense that they are independent of the recommendation strategy, effectively ignoring the model's own capability to reduce the mainstream bias or even amplify it. As a result, the learning process could be tailored to the wrong users.

In this paper, we choose to focus there where the effect of mainstreamness is *directly* observed, that is, the recommendation utility

ICTIR '23, July 23, 2023, Taipei, Taiwan
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0073-6/23/07.
https://doi.org/10.1145/3578337.3605134

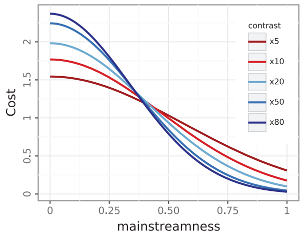

Figure 1: Cost functions used in the paper. The contrast denotes the relative cost between users with mainstreamness 0 and users with 1 (i.e. x10 means 10 times as much).

provided by the data and recommendation model at hand. If a user receives poor recommendations it could be because their preferences deviate from the rest, or because there is not enough data to properly quantify their similarity to other users or to fully exploit it. Therefore, we choose utility as an implicit proxy for mainstreamness. Through this quantification of user mainstreamness, we make the training process focus on the non-mainstream ones by assigning them higher weights. We do so, however, in a *cost-sensitive* way [22], taking the cost of recommendation errors into account while training the CF model. Our results show that our implicit measurement of mainstreamness via utility is better able to differentiate niche users than an explicit approach, and that the cost-sensitive learning strategy does mitigate the bias by balancing the recommendation quality across users. Finally, we investigate data requirements for conducting research on mainstream bias at the individual user level, and provide suggestions for reliable experimentation in this area.

2 PROPOSED APPROACH

The basis of our approach is a weighted loss function where every user $u \in \mathcal{U}$ is assigned a weight $\omega(u)$ that informs the learning process about the importance of every user's individual recommendation loss. The global loss is thus simply

$$\mathcal{L} = \sum_{u \in \mathcal{U}} \omega(u) \mathcal{L}_R(u) , \quad (1)$$

where the recommendation loss $\mathcal{L}_R$ is specific of the model and learning paradigm. This way, we explicitly tell the learning process what users to optimize for by means of ω, which, in our case, should be high for non-mainstream users and low for mainstream users.

2.1 Definition of Weights

As explained in the previous section, we define ω as a function of the user mainstreamness m_u. However, rather than simply using a naïve transformation of m_u, we introduce flexibility through a cost function that maps user mainstreamness onto a cost value. In particular, and assuming m_u ranges between 0 and 1, we use the density function of a Normal distribution truncated between 0 and 1, with zero mean and variance adjusted to achieve a contrast ranging between 5 (i.e. users with mainstreamness $m_u = 0$ have a cost 5 times as large as users with $m_u = 1$) and 80 (ie. 80 times as much). This is a simple choice to make ω smooth and monotonically decreasing, but other cost functions that emphasize different levels of mainstreamness are of course possible; we leave this discussion for further work. Fig. 1 shows some examples. Nonetheless, the formulation of the cost function may consider various aspects tailored to the business case, as well as different magnitudes for the contrast between users with low and high mainstreamness. For example, it would be reasonable to assign very high weights to non-mainstream users with high activity, or to users with very low activity as an attempt to reduce the churn rate.

An important point to consider when defining ω is the distribution of mainstreamness across users. It could be the case that, given the current data and model, the least mainstream users are actually fairly mainstream already, so their weight relative to the most mainstream users should be adjusted via a smaller contrast. It could also be the case that the dataset is very sparse and there are simply not enough neighbors around users for the model to learn a good representation. That is, the majority of users could be considered non-mainstream, and as a result the cost function would hardly differentiate among them. Lastly, one could decide to compute m_u in several different ways (see next Section), which could potentially lead to quite different mainstreamness score distributions altogether, ultimately leading to a different set of weight values even for the same users.

In order to minimize this dependence on the dataset and mainstreamness definition, and ensure that the full co-domain of the cost function is used, we first normalize the raw mainstreamness scores. Simply re-scaling between the minimum and maximum could still lead to a disproportionate use of small parts of the co-domain, and would also be very sensitive to outlier users. Instead, we use the rank statistic of m_u normalized in $[0, 1]$. We achieve this by using the empirical cumulative distribution function (ecdf)

$$\omega(u) = \mathrm{cost}(\mathrm{ecdf}_{\mathcal{U}}(m_u)) , \quad (2)$$

where, as mentioned, cost is defined in terms of a truncated Normal density function.

2.2 Measurement of Mainstreamness

An **explicit** approach to compute m_u would ideally follow some notion of mainstreamness, but mainstreamness is itself a complex construct very hard to define formally [1, 10, 24]. Recently, Zhu and Caverlee [24] took inspiration from outlier detection techniques to propose four different definitions:

- Sim: users are mainstream to the extent that their interactions are similar to that of the other users. The Jaccard coefficient is used to measure the average similarity between a user and all the others.
- Den: users are mainstream to the extent that there are enough close neighbors to calculate similarity with. The local outlier factor algorithm (LOF) [2] is used to identify niche users.
- Dis: users are mainstream to the extent that their interactions are common in the dataset, that is, they interact with popular items. The cosine similarity is used to measure the similarity between a user and the average user interactions.

[1] Data available from the authors' public repository at https://github.com/Zziwei/Measuring-Mitigating-Mainstream-Bias.

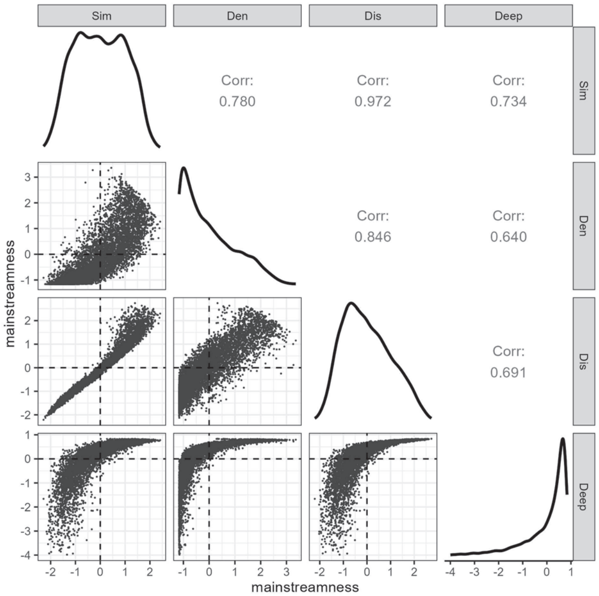

Figure 2: Comparison of the four mainstreamness definitions proposed by Zhu and Caverlee [24], as applied to the MovieLens 1M dataset.[1]Density plots illustrate the distributions of mainstreamness. Scatter plots show the relationship between pairs of definitions, quantified in the upper-right half via Pearson correlation scores. Scores are standardized to zero mean and unit variance for better comparison.

- Deep: similar to Den, niche users are identified by an outlier detection algorithm. In particular, the deep support vector data description algorithm (DeepSVDD) [19] is used.

However, it is difficult to assess how well these, or any other definitions for that matter, correlate with the concept of mainstreamness. To illustrate, Fig. 2 compares these four definitions as applied to the MovieLens 1M dataset. Although they are somewhat correlated to one another, it is evident that they produce very different scores. For instance, Sim and Dis lead to nicely shaped distributions, suggesting few users with extreme (non-)mainstreamness. However, Den and Deep lead to very skewed distributions, even in the opposite direction, pointing to many users with extreme scores. This shows that the same user could be considered both mainstream or non-mainstream, depending on how we choose to define mainstreamness.

Furthermore, it should be noted that these four definitions of mainstreamness are agnostic to the recommendation model. However, the effect of mainstreamness, ultimately, depends on the model and how it is able to exploit the specifics of the dataset it is trained on. It is not far-fetched to think of a user, assessed as non-mainstream, who receives bad recommendations under one model but good recommendations under a more capable one.

This leads us to consider an alternative, **implicit** way to quantify mainstreamness that is *not* model agnostic. In particular, we decide to focus there where the effect of mainstreamness is to be observed, that is, the recommendation *utility* provided by the recommendation model at hand. This is where mainstreamness will ultimately have an impact on. The very nature of collaborative filtering tells us that if a user receives poor recommendations it is because they are non-mainstream under the current model: they cannot be properly represented, either because their preferences are somehow different from their closest neighbors, or because there are not enough data to properly quantify their similarity. Therefore, we use utility as a proxy for mainstreamness. Since utility, just like mainstreamness, is a complex concept difficult to measure, we decide to simply use the accuracy of the recommendation model for that user, measured through a metric like *nDCG* or *AP*.

Table 1: Dataset statistics after pre-filtering.

Dataset	#users	#items	#ratings	Density
MovieLens 1M [6]	6,040	3,609	562,957	2.583%
BeerAdvocate [13]	8,821	43,663	780,752	0.203%
Amazon Digital Music [16]	14,057	379,171	619,673	0.011%
Amazon Musical Instruments [16]	15,270	585,766	862,798	0.010%

But there is the question of what accuracy scores we actually use. In principle, these scores should reflect user mainstreamness when there is no mechanism to minimize its effect, and they should be achieved by the recommendation model in the dataset at hand. Therefore, we decide to use the accuracy achieved, *on a validation set*, by the vanilla model whose loss function is as in Eq. (1) but using no weights. As intended, we thus first see how the model reacts to mainstreamness as reflected in the observed utility for users, and then act upon it in a cost-sensitive way.

3 EXPERIMENTAL DESIGN

We carried out a number of experiments to investigate the effectiveness of the proposed approach in mitigating the mainstreamness bias, as well as the effect of the contrast applied by the cost function. In particular, we study contrasts x5, x10, x20, x50 and x80, that is, the most non-mainstream user has a weight between 5 and 80 times larger than that of the most mainstream user. Fig. 1 details the cost functions. Regarding the measurement of mainstreamness, we consider both an explicit and an implicit quantification. For the former, we follow Zhu and Caverlee [24] and compute Sim scores. This choice is motivated by the time complexity of their four approaches (the computation of mainstreamness may quickly become intractable as the numbers of users and items increase; while their datasets include a few thousand items, ours span from a few thousands to over half a million), and their correlation to one another (Sim is also the one most correlated with the others, in particular with Deep). For the implicit quantification we compute utility scores using the metric *nDCG* as an exemplar of recommender systems research; hereafter, we will refer to this definition of mainstreamness as Util.

We selected four real-world datasets containing user-item rating interactions from various domains and with different densities, especially including some highly sparse datasets (see Table 1). In line with common practice in ranking-oriented recommender systems research, we see all existing interactions in the datasets as relevant, and all other unseen interactions as irrelevant. We use

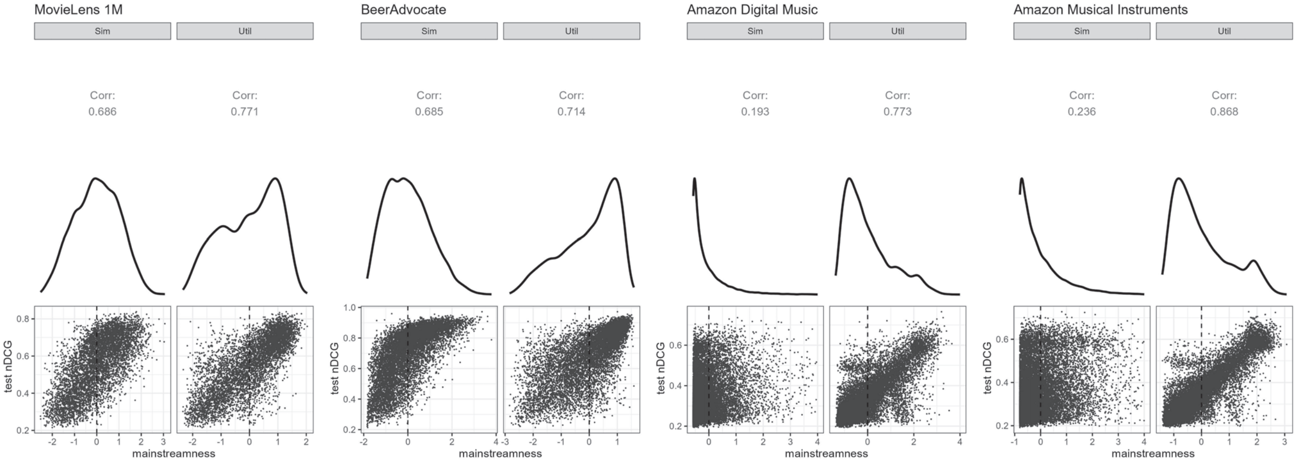

Figure 3: Correlation between mainstreamness and test nDCG in the baseline model (FM), for each mainstreamness definition. Density plots illustrate the distribution of mainstreamness. Scatterplots show their relationship with nDCG, quantified via Pearson correlation scores. Mainstreamness scores are standardized to zero mean and unit variance for better comparison.

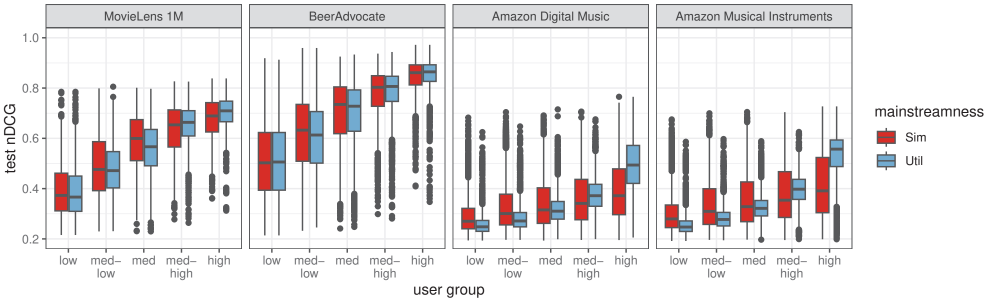

Figure 4: Correlation between user groups, split by mainstreamness, and test nDCG in the baseline model (FM).

LensKit [3] to evenly split the relevant items for each user into training, validation and test sets. To make the modeling of utility —and hence mainstreamness— robust, each user has at least five relevant transactions in each of the three sets; we explain the rationale for this decision in Section 5. For training the model, we follow He et al. [7], Wu et al. [23] and randomly sample four irrelevant items per relevant item in the training partition. For validation and test, we follow DaisyRec [21] and evaluate the model for each user by ranking a total of 500 items consisting of their relevant items in the validation/test partition and a set of randomly sampled irrelevant items. Finally, to make sure relevant items are the minority, as happens in reality, we truncate the number of relevant interactions to 200. The dataset statistics after processing are shown in Table 1.

Regarding the recommendation model, we deploy a simple but effective CF model that only utilizes user-item interactions. Specifically, we choose Factorization Machines (FM) [18], which optimize the binary cross-entropy (BCE) loss via the Adaptive Moment Estimation (Adam) [8] learner, and leave the investigation on other training paradigms for future work. For each user, the BCE loss is normalized by dividing by the total number of relevant and irrelevant items used for training, so that all user losses are on the same scale in (1). After a fine-tuning process based on grid search, we fixed several key hyper-parameters including the dimension of vectors used for interaction (32), learning rate (0.0001), L2-regularization coefficient to avoid overfitting (0.001), and batch size (512).

All models are trained for 300 epochs to ensure full convergence, and with 3 different random initializations to minimize random effects due to the sampling process. The whole pipeline is implemented in PyTorch [17], and all experiments are run on one NVIDIA GeForce GTX 2080Ti GPU [2].

4 RESULTS

4.1 Mainstreamness and Utility

We first examine how Sim and Util differentiate between mainstream and non-mainstream users. In particular, we are interested in how well they correlate with the test *nDCG* scores obtained by the baseline FM model: non-mainstream users should receive recommendations with low *nDCG* scores, while mainstream users should receive higher scores.

[2] All data, code and results are available at https://github.com/roger-zhe-li/ictir23-cost-sensitive.

Table 2: Mean nDCG of the baseline model (FM) per user group, and relative percentage improvement of each cost-sensitive model (e.g. users in group 'low' of MovieLens 1M received a score of .3284 with the baseline, and an improvement of +3.89% with the x80-contrast cost-sensitive model under the Util mainstreamness definition). Column 'Overall' lists the mean across all users. Green/red for statistically significant gain/loss with respect to the baseline (hierarchical linear model with seed and user random effects, Bonferroni correction).

		MovieLens 1M						BeerAdvocate						Amazon Digital Music						Amazon Musical Instruments					
				med-		med-				med-		med-				med-		med-				med-		med-	
		Overall	low	low	med	high	high	Overall	low	low	med	high	high	Overall	low	low	med	high	high	Overall	low	low	med	high	high
FM		.5531	.3284	.4621	.5753	.6613	.7388	.6887	.4144	.6051	.7301	.8132	.8809	.3456	.2324	.2695	.3145	.3828	.5289	.3606	.2348	.2772	.3276	.4085	.5552
Sim	x5	.5465	-0.36	-0.89	-1.62	-1.3	-1.33	.6792	-0.52	-1.72	-1.83	-1.44	-1.13	.3395	+0.65	+0.05	-0.61	-1.8	-4.45	.3581	+1.61	+0.77	-0.09	-1.06	-2.5
	x10	.5437	-0.47	-1.32	-2.23	-1.87	-1.94	.6734	-0.81	-2.87	-2.94	-2.27	-1.8	.3368	+0.99	+0.1	-1	-2.67	-6.27	.3577	+3.29	+1.44	+0.08	-1.68	-3.54
	x20	.541	-0.53	-1.76	-2.85	-2.41	-2.51	.6666	-1.35	-4.23	-4.13	-3.22	-2.56	.3347	+1.31	+0.09	-1.3	-3.46	-7.69	.3576	+3.33	+1.44	+0.04	-1.72	-3.58
	x50	.5376	-0.67	-2.28	-3.64	-3.06	-3.26	.6588	-2.01	-5.65	-5.56	-4.29	-3.54	.3328	+1.63	+0.07	-1.6	-4.13	-8.92	.3576	+3.37	+1.45	+0.03	-1.75	-3.6
	x80	.5359	-0.67	-2.55	-4.06	-3.39	-3.59	.6548	-2.55	-6.39	-6.17	-4.84	-4.08	.3316	+3.66	+0.9	-1.78	-5.07	-10.62	.3576	+3.39	+1.45	+0.02	-1.77	-3.61
Util	x5	.5567	+1.67	+1.81	+0.63	+0.16	-0.13	.6846	+0.63	-0.41	-0.94	-0.83	-0.77	.3454	+1.59	+1.43	+1.2	+0.4	-2.58	.3607	+1.1	+0.96	+0.62	-0.04	-1.19
	x10	.5574	+2.38	+2.34	+0.7	+0.11	-0.27	.6807	+0.44	-1.19	-1.66	-1.39	-1.27	.3453	+2.54	+2.09	+1.53	+0.18	-3.5	.3607	+2	+1.52	+0.78	-0.32	-1.8
	x20	.5579	+3.05	+2.87	+0.73	0	-0.48	.6762	+0.54	-2.11	-2.67	-2.09	-1.74	.3454	+3.59	+2.78	+1.64	+0.11	-4.25	.3607	+2.63	+1.93	+0.8	-0.53	-2.08
	x50	.5579	+3.62	+3.31	+0.68	-0.21	-0.84	.6722	+2	-3.09	-3.86	-2.89	-2.32	.3458	+4.84	+3.67	+1.92	-0.11	-4.9	.3608	+3.94	+2.48	+0.85	-0.83	-2.66
	x80	.5577	+3.89	+3.47	+0.63	-0.32	-1.05	.6715	+2.98	-3.2	-4.25	-3.14	-2.54	.346	+5.45	+4	+2.02	-0.18	-5.15	.3608	+4.48	+2.64	+0.88	-0.97	-2.86

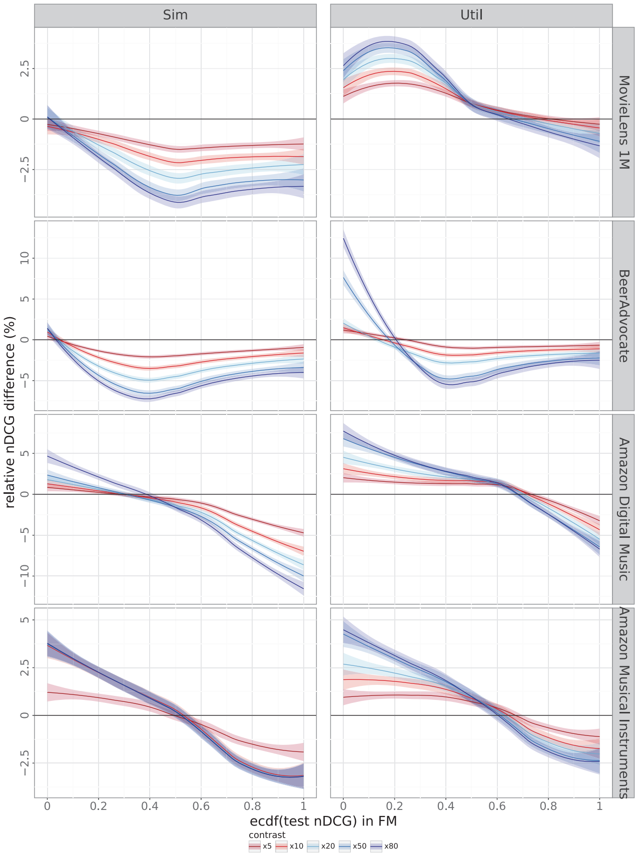

Figure 5: Mean nDCG relative percentage improvement between cost-sensitive models and baseline model, as a function ecdf(test nDCG) in the baseline FM model, for a sample data split. Curves fitted by a LOESS model. Ribbons indicate 95% confidence intervals.

For each of the four datasets, Fig. 3 compares Sim and Util. We can first see that both approaches lead to similar distributions in the Amazon datasets, where there appear to be many non-mainstream users. However, they somewhat disagree in the BeerAdvocate dataset, where Util does not identify many non-mainstream users to benefit from the cost-sensitive approach. In terms of correlation with the test *nDCG* scores, we can see that Util is much better correlated, specially in the Amazon datasets. This points to the possibility that Sim identifies many non-mainstream users to which the model is still able to offer good recommendations. If the training process increases their importance by assigning them a high weight ω, we may loose the opportunity to focus on those users that still receive poor recommendations.

In order to assess the effectiveness of the cost-sensitive approach for the mitigation of the mainstream bias, we will look in the next Section into different groups of users separated by their mainstreamness: group 'low' contains the 20% of users with lowest mainstreamness scores on the baseline model, group 'med-low' contains the next 20% or users, group 'med' contains the middle 20% of users, and so on with groups 'med-high' and 'high'. An effective mitigation of the mainstream bias would be reflected in increased performance for the lower groups, which ideally should be those with lowest test *nDCG* scores in the baseline model. Fig. 4 shows how well Sim and Util separate users in these five groups. We can first see that the groups are indeed correlated with *nDCG*, but we can notice that this correlation is stronger with Util, specially in the Amazon datasets (the low groups receive lower utility, and the higher groups receive higher utility). We can also see that groups tend to overlap substantially when separated by Sim, potentially misplacing users. This overlap can be quantified by an ANOVA model of *nDCG* modeled by two factors: dataset and user-group nested within dataset. Indeed, the user-group effect has a much larger sum of squares (SS) with Util than with Sim (SS=440 vs SS=218; SS of the dataset effect is 843). Finally, Fig. 4 also points that the BeerAdvocate dataset may be hard to further optimize for because the utility scores are already relatively high.

4.2 Bias Mitigation

An effective mitigation of the mainstream bias would be reflected in increased performance for the lower groups (i.e. mainly 'low' and 'med-low'), ideally with no detriment to the higher groups and,

Table 3: Same as Table 2, but user groups defined by Sim scores instead of test nDCG in the baseline model.

		MovieLens 1M						BeerAdvocate						Amazon Digital Music						Amazon Musical Instruments					
		Overall	low	med-low	med	med-high	high	Overall	low	med-low	med	med-high	high	Overall	low	med-low	med	med-high	high	Overall	low	med-low	med	med-high	high
FM		.5531	.3284	.4621	.5753	.6613	.7388	.6887	.4144	.6051	.7301	.8132	.8809	.3456	.2324	.2695	.3145	.3828	.5289	.3606	.2348	.2772	.3276	.4085	.5552
Sim	x5	.5465	-0.62	-0.97	-1.16	-1.34	-1.58	.6792	-1.4	-1.39	-1.43	-1.34	-1.35	.3395	-0.04	-0.61	-1.37	-2.16	-4.05	.3581	+0.01	-0.16	-0.52	-0.89	-1.65
	x10	.5437	-0.84	-1.42	-1.63	-1.87	-2.31	.6734	-2.23	-2.4	-2.23	-2.14	-2.16	.3368	0	-0.86	-2.03	-3.18	-5.71	.3577	+0.26	-0.08	-0.53	-1.14	-2.11
	x20	.541	-1.19	-1.83	-2.12	-2.35	-2.96	.6666	-3.42	-3.54	-3.17	-2.95	-3.06	.3347	-0.05	-1.15	-2.52	-4.01	-6.97	.3576	+0.24	-0.11	-0.57	-1.16	-2.14
	x50	.5376	-1.49	-2.35	-2.71	-3.02	-3.84	.6588	-4.71	-4.74	-4.31	-3.92	-4.2	.3328	-0.17	-1.42	-3.01	-4.67	-8	.3576	+0.24	-0.09	-0.58	-1.18	-2.16
	x80	.5359	-1.76	-2.68	-2.95	-3.32	-4.19	.6548	-5.34	-5.29	-4.83	-4.46	-4.91	.3316	+0.03	-1.52	-3.34	-5.18	-8.87	.3576	+0.23	-0.1	-0.59	-1.18	-2.17
Util	x5	.5567	+1.5	+0.99	+0.53	+0.32	+0.26	.6846	-0.46	-0.5	-0.51	-0.66	-0.72	0.3454	+0.12	+0.11	-0.16	+0.07	-0.3	0.3607	-0.01	+0.06	0	-0.04	+0.12
	x10	.5574	+2.05	+1.24	+0.67	+0.33	+0.2	.6807	-1.11	-1.16	-1.1	-1.19	-1.22	0.3453	+0.13	+0.06	-0.11	+0	-0.43	0.3607	0	+0.12	+0.01	-0.06	+0
	x20	.5579	+2.52	+1.59	+0.71	+0.27	+0.08	.6762	-1.85	-2.03	-1.79	-1.73	-1.72	0.3454	+0.11	+0.06	-0.06	+0.15	-0.5	0.3607	0	+0.14	+0.05	-0.08	+0.01
	x50	.5579	+2.92	+1.8	+0.64	+0.12	-0.16	.6722	-2.66	-2.85	-2.53	-2.13	-2.05	0.3458	+0.19	+0.07	+0.13	+0.22	-0.3	0.3608	+0.16	+0.23	-0.03	-0.05	-0.04
	x80	.5577	+3.12	+1.86	+0.61	+0.01	-0.33	.6715	-2.92	-3	-2.65	-2.15	-2.05	0.346	+0.23	+0.09	+0.15	+0.29	-0.18	0.3608	+0.16	+0.21	+0.01	-0.02	-0.06

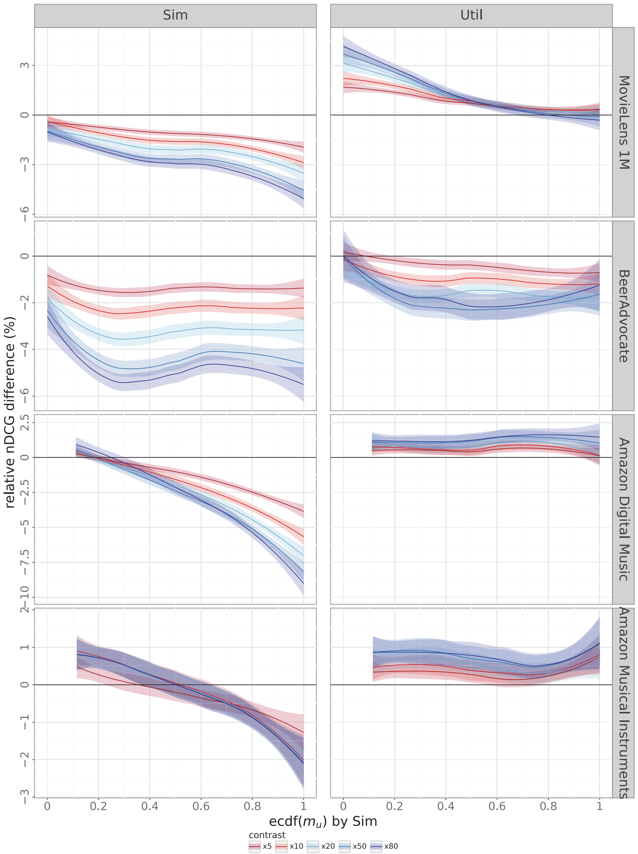

Figure 6: Same as Fig. 5, but plotted against ecdf(m_u) by Sim instead of ecdf(test nDCG) in the baseline model.

especially, overall. In the previous section we separated users into groups by each of Sim and Util, but here we separate them directly by their test *nDCG* with the baseline model FM, because this better illustrates how non-mainstream users suffer from the bias.

Table 2 reports the relative percentage improvement in *nDCG* scores per user group, as well as the overall mean score across all users in the dataset. We can clearly see that the use of Sim benefits the non-mainstream users only in the two Amazon dataset; in MovieLens and BeerAdvocate they are even hurt further. In contrast, Util is always able to improve the utility of non-mainstream users across datasets, achieving relative *nDCG* improvements of up to 5% in the Amazon datasets. Improvements on the lower user groups are generally higher than losses on the higher groups, where users already receive (very) high recommendation utility anyway and such minor losses are probably unnoticed. This redistribution of model performance has a negligible effect on the global performance of the models, as evidenced by the overall *nDCG* scores. This means that, with proper selection of the contrast in the cost function, Util can minimize the mainstream bias at virtually no overall loss in utility. On the other hand, the use of Sim for training leads to inferior overall performance on all four datasets.

Fig. 5 presents a more fine-grained picture with one of the three random initializations in our experiments. Curve segments above 0 represent an improvement by the cost-sensitive models, while segments below 0 represent a loss. We can confirm that the cost-sensitive approach indeed makes the models focus on the non-mainstream users, as shown by the nicely smooth correlation between observed utility and relative improvement, moderated by the contrast in the cost function. As expected though, this focus on the non-mainstream users comes at the cost of a utility loss for the mainstream users on the right-hand side of the plots. Nevertheless, when using Util the relative loss for those users is generally much smaller than the gain for the very non-mainstream users, which are our main target. The figure also shows that the actual relation between improvement and utility varies across datasets, as reflected by the different curve shapes. This is explained by the differences in the shape of their *nDCG* distributions (see Fig. 3); recall that we use the *ecdf* of the scores. In a side-by-side comparison between Sim and Util, we see that Util offers better performance nearly everywhere along the x-axis, but especially for the non-mainstream users.

In summary, we see that our cost-sensitive approach brings better balance across users, thus helping in the mitigation of the mainstream bias. In addition, we confirm that an implicit quantification of mainstreamness like Util works better than an explicit quantification like Sim in steering the learning process towards better recommendations for the users that receive low utility from the baseline model. In addition, we note that the mitigation effect via Util does not decay with increasing data sparsity (refer back to Table 1).

One could be tempted to argue that Util should obviously offer better results than Sim when analyzing *test nDCG* because it is based on *validation nDCG* scores; test and validation scores should be highly correlated (we will come back to this in Section 5). After all, both Table 2 and Fig. 5 analyze results by test *nDCG*. The argument made above is that differences between mainstream and non-mainstream users can be immediately identified by test scores, but for the sake of clarity and to avoid potentially unfair assessment towards Sim, Table 3 reports the same results but separating users by Sim, while Fig. 6 does so by plotting against Sim. While the results are less clear with this partition of users, the table confirms that models trained with Sim are generally better at mitigating the bias than those trained with Util. In particular, results for the BeerAdvocate dataset show that higher contrasts even lead to worse performance for the lower user groups, suggesting that Sim is perhaps not properly identifying non-mainstream users. The figure shows that Util improves over the baseline across all levels of mainstreamness in the Amazon datasets, further suggesting that Sim identifies as non-mainstream users that are probably not. In summary, and even though this comparison could in turn be considered favorable to Sim (note that previously we assessed against *test nDCG*, not against the *validation nDCG* calculated by Util), the results again support the use of Util to quantify user mainstreamness and mitigate the bias.

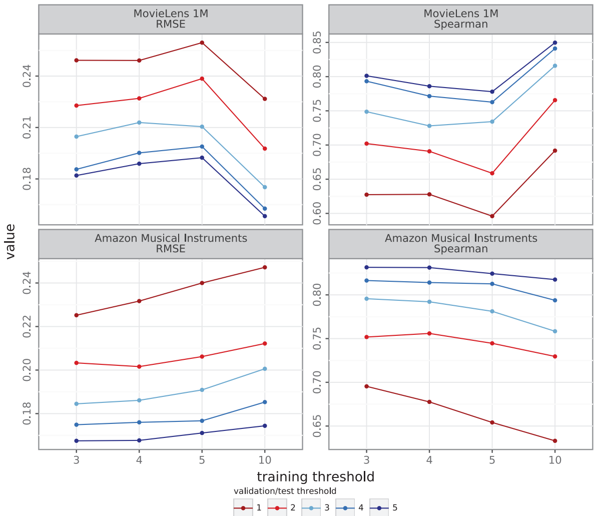

Figure 7: Correlation between validation and test scores as a function of the amount of data used for training, validation, and testing, for two sample datasets (most and least dense).

5 DISCUSSION

A key assumption of our approach based on Util is that we can reliably use utility, measured as the accuracy on a validation set, to determine the weight that each user should have in the training process. This implies that the accuracy on the validation set is a good estimate of the accuracy on the test set, which is where the effect will ultimately be assessed. If there was a low correlation between validation and test accuracy, the loss function would apply high weights for users that do not really need it, limiting or even altogether canceling the potential of our approach.

Intuitively, how well validation and test scores correlate is mainly determined by the amount of data. If only a few interactions are involved in the calculation of accuracy, the resulting scores will bear a high degree of noise or random error, thus lowering the correlation. In principle, we would therefore use as much data as possible in the validation and test sets. However, we would generally prefer to use all that data to actually train the model, but we note that the validation scores are somehow part of the training process itself, because they determine the weights.

A balance is therefore necessary, so we need to study the strength of the validation-test scores correlation as a function of the number of interactions in their data partitions. We did this by running the baseline FM model on different data partitions with varying minimum numbers of relevant items in the training set (3, 4, 5 and 10), and validation and test sets (1, 2, 3, 4 and 5 each). The actual split was conducted maintaining proportions (i.e. for the combination of 4/3/3 minimum items per set, a user has 40% of their relevant items for training, 30% for validation, and 30% for testing). We then measured the strength of the validation-test correlation via the RMSE of the scores and their Spearman ρ correlation.

Fig. 7 shows that, as expected, the correlation increases (low RMSE, high ρ) with the number of relevant interactions used in the validation and test sets. More interestingly, it shows that the amount of training data has a much smaller and varying effect, so despite it being a major factor to maximize model performance, it is not so to robustly assess that performance. The plots indicate that requiring only one or two interactions in the validation set would lead to noisy scores; four interactions seem the bare minimum. As for the training set, the usual practice of having at least as much data as for validation and testing still applies in this context of non-time-aware recommendation.

All in all, our suggestion for this line of research on mainstream bias that works at the individual user level, is to have no less than four items per user in each of the three standard data partitions. Because the strength of the correlation is a key factor in our approach, we decided to require at least five to be on the safe side. In fact, we also observed that the effect of cost-sensitive learning in the validation sets is similar to what is reported in Figs. 5 and 6.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we tackled the challenge of mainstream bias in CF-based recommendation. The main aspect we focused on is to steer the process of mitigating this bias directly by the utility resulting from the recommendation model and data at hand. For this purpose, we proposed an approach that assigns each user an importance weight during training, with these weights defined in a cost-sensitive manner. By choosing to steer the model directly towards the users that receive low utility, and not towards those that *appear* to be non-mainstream, we avoid the model to focus on users that already receive high utility even if they were not expected to. This way, the model does focus on the niche users that suffer from the bias.

Empirical results show that such models produce a more effective balance of the recommendation utility among the mainstream and non-mainstream users, in a way that is consistent across datasets with varying properties. In addition, we provide suggestions regarding the minimum number of interactions to require when partitioning datasets. Without enough interactions, research on mainstream bias at the level of individual users might produce unreliable results.

For future work, we will first explore other ways to quantify mainstreamness. In the implicit measurement sense, an evident question is whether other metrics such as AP, or even the combination of multiple metrics, work better at identifying niche users. Additionally, we can think of ways to make the validation-test correlation robust to issues like sample selection bias, for example via inverse propensity scoring. Another line is to explore more principled approaches for an explicit quantification through an extensive study of the factors that influence mainstreamness, such as the temporal dynamics.

Regarding our cost-sensitive learning approach, we will explore its generality, to see how it works for underlying models other than FM or other ranking frameworks such as pairwise and listwise. We will also investigate the combination of cost-sensitive and adversarial learning strategies to mitigate mainstream bias: cost-sensitive to tell the model where to focus on, and adversary to tell how.

Finally, we note that our focus in this paper has been on the effect of mainstream bias mitigation on the users, but one could wonder about what effect it has on the items. One hypothesis is that non-mainstream users are better served because the less popular items are now more likely to be recommended, so it would be interesting to study whether mitigating one bias amplifies or mitigates other biases, such as popularity or position.

REFERENCES

[1] Christine Bauer and Markus Schedl. 2018. Investigating Cross-Country Relationship between Users' Social Ties and Music Mainstreaminess. In *ISMIR*. 678–686.

[2] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. In *SIGMOD Conference*. ACM, 93–104.

[3] Michael D. Ekstrand. 2020. LensKit for Python: Next-Generation Software for Recommender Systems Experiments. In *CIKM*. ACM, 2999–3006.

[4] 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 *FAT (Proceedings of Machine Learning Research, Vol. 81)*. PMLR, 172–186.

[5] David Goldberg, David A. Nichols, Brian M. Oki, and Douglas B. Terry. 1992. Using Collaborative Filtering to Weave an Information Tapestry. *Commun. ACM* 35, 12 (1992), 61–70.

[6] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2016), 19:1–19:19.

[7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. ACM, 173–182.

[8] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

[9] Jurek Leonhardt, Avishek Anand, and Megha Khosla. 2018. User Fairness in Recommender Systems. In *WWW (Companion Volume)*. ACM, 101–102.

[10] Roger Zhe Li, Julián Urbano, and Alan Hanjalic. 2021. Leave No User Behind: Towards Improving the Utility of Recommender Systems for Non-mainstream Users. In *WSDM*. ACM, 103–111.

[11] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented Fairness in Recommendation. In *WWW*. ACM / IW3C2, 624–632.

[12] Yudan Liu, Kaikai Ge, Xu Zhang, and Leyu Lin. 2019. Real-time Attention Based Look-alike Model for Recommender System. In *KDD*. ACM, 2765–2773.

[13] Julian J. McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning Attitudes and Attributes from Multi-aspect Reviews. In *ICDM*. IEEE Computer Society, 1020–1025.

[14] Alessandro B. Melchiorre, Navid Rekabsaz, Emilia Parada-Cabaleiro, Stefan Brandl, Oleg Lesota, and Markus Schedl. 2021. Investigating gender fairness of recommendation algorithms in the music domain. *Inf. Process. Manag.* 58, 5 (2021), 102666.

[15] Mohammadmehdi Naghiaei, Hossein A. Rahmani, and Yashar Deldjoo. 2022. CPFair: Personalized Consumer and Producer Fairness Re-ranking for Recommender Systems. In *SIGIR*. ACM, 770–779.

[16] Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 188–197.

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*. 8024–8035.

[18] Steffen Rendle. 2010. Factorization Machines. In *ICDM*. IEEE Computer Society, 995–1000.

[19] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert A. Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep One-Class Classification. In *ICML (Proceedings of Machine Learning Research, Vol. 80)*. PMLR, 4390–4399.

[20] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. *Learning internal representations by error propagation*. Technical Report. California Univ San Diego La Jolla Inst for Cognitive Science.

[21] Zhu Sun, Di Yu, Hui Fang, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. 2020. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison. In *RecSys*. ACM, 23–32.

[22] Nguyen Thai-Nghe, Zeno Gantner, and Lars Schmidt-Thieme. 2010. Cost-sensitive learning methods for imbalanced data. In *IJCNN*. IEEE, 1–8.

[23] Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. 2022. Rethinking InfoNCE: How Many Negative Samples Do You Need?. In *IJCAI*. ijcai.org, 2509–2515.

[24] Ziwei Zhu and James Caverlee. 2022. Fighting Mainstream Bias in Recommender Systems via Local Fine Tuning. In *WSDM*. ACM, 1497–1506.