Calibration in Collaborative Filtering Recommender Systems: a User-Centered Analysis

Kun Lin
klin13@depaul.edu
DePaul University
Chicago, USA

Nasim Sonboli
University of Colorado Boulder
Boulder, USA
nasim.sonboli@colorado.edu

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

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

ABSTRACT

Recommender systems learn from past user preferences in order to predict future user interests and provide users with personalized suggestions. Previous research has demonstrated that biases in user profiles in the aggregate can influence the recommendations to users who do not share the majority preference. One consequence of this bias propagation effect is miscalibration, a mismatch between the types or categories of items that a user prefers and the items provided in recommendations. In this paper, we conduct a systematic analysis aimed at identifying key characteristics in user profiles that might lead to miscalibrated recommendations. We consider several categories of profile characteristics, including similarity to the average user, propensity towards popularity, profile diversity, and preference intensity. We develop predictive models of miscalibration and use these models to identify the most important features correlated with miscalibration, given different algorithms and dataset characteristics. Our analysis is intended to help system designers predict miscalibration effects and to develop recommendation algorithms with improved calibration properties.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Collaborative filtering**; *Personalization*; • **Social and professional topics** → *User characteristics.*

KEYWORDS

Calibration; Algorithmic Bias; Bias Amplification; Fairness; Recommendation Algorithms

ACM Reference Format:
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 (HT ’20), July 13–15, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3372923.3404793

1 INTRODUCTION

With the growing influence of recommender systems on our daily decisions across various domains, there has been a shift of focus in research from accuracy to other socially-sensitive concerns such as diversity and fairness. Optimization that enhances overall accuracy in recommendation may have the unwanted side effect of disadvantaging certain users and user groups whose tastes are less mainstream. One aspect of this unfair disparity among users can be measured using the metric of *calibration.* Calibration measures how similar a set of recommended items is to a particular user’s interest profile. A machine learning model’s output is considered well calibrated if the class proportion in the prediction is consistent with the class proportion in the input historical data [29].

A recommendation algorithm’s output is considered calibrated when each user’s distribution of interests over a set of item categories in their profile is consistent with that of their recommendation list [32]. A well-calibrated recommendation list therefore reflects all of a user’s interests in the appropriate proportion. Delivering calibrated recommendations can play a very important role in a user’s experience. Conversely, miscalibrated results, which do not reflect a user’s full range of interests, may have consequential impact in sensitive contexts such as employment or financial services recommendations.

In Figure 1, we demonstrate how recommendation models can fail to deliver calibration. The hex plot examines the relative frequency of “Action” films in both the users’ profiles and in their recommendations. The red line represents calibrated recommendations where each user sees action films in proportion to their individual interest, as represented in their profiles. The x-axis shows the prevalence of action movies in user profiles, and we can see users are generally with higher interest in action movies and an almost bell-shape distribution curve, peaking around 0.7. The y-axis shows the prevalence of action movies in user recommendation lists, and the preference distribution in the recommendation lists is left-skewed, with many recommendation lists containing only action movies. On average, action movies are recommended more heavily than their popularity would warrant, and we can see in the density plot that there are particular individuals (represented

HT ’20, July 13–15, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7098-1/20/07…$15.00
https://doi.org/10.1145/3372923.3404793

in the cells above the red line) who receive these types of movie recommendations more than their profiles would suggest.

We can say that the population preference demonstrated here is a kind of bias. Action movies appear often and are rated generally highly in the MovieLens data set, as opposed to other types of movies, for example, Musicals. What Figure 1 demonstrates is a form of *bias or preference propagation*, the tendency of the algorithm to amplify the population preference in the recommendations it generates. Here the bias is relative to particular item categories, but bias propagation has also been noted relative to item popularity more generally under the heading of popularity bias [19]. We will use the terms bias propagation and preference propagation interchangeably in this paper.

Patterns in preference propagation are not negative by themselves, as they are a key ingredient that recommendation algorithms use to construct predictive models and provide users with personalized outputs. However, as has been noted in the context of popularity bias [2], the fairness of a recommender system suffers if some users get higher quality results than others and calibration is one indication of recommendation quality. The solution proposed in [32] to resolve miscalibration was a post-processing approach. They used a reranking method based on the notion of *Maximum Marginal Relevance* to achieve calibrated recommendations. Without deeper insights on what can lead to miscalibration, the research problem is changed from reaching calibration into finding a trade-off between accuracy and calibration.

We should note that the importance of calibration may vary in different contexts, and may need to be balanced against other system objectives. For example, a user with a relatively sparse profile may not have provided enough information for the category distribution across items to be meaningful. Calibration may be thought of as an objective that is in tension with the goal of diversity in recommendations, although we can also think of it as a form of diversity tailored to the individual. Still, there are recommendation contexts in which more general diversity is sought and where filter bubbles [28] are a concern. For example, in news recommendation, a focus on calibrated recommendations might contribute to political polarization in society. For this reason, we focus here on consumer taste domains where calibration is considered desirable and valued by users [32].

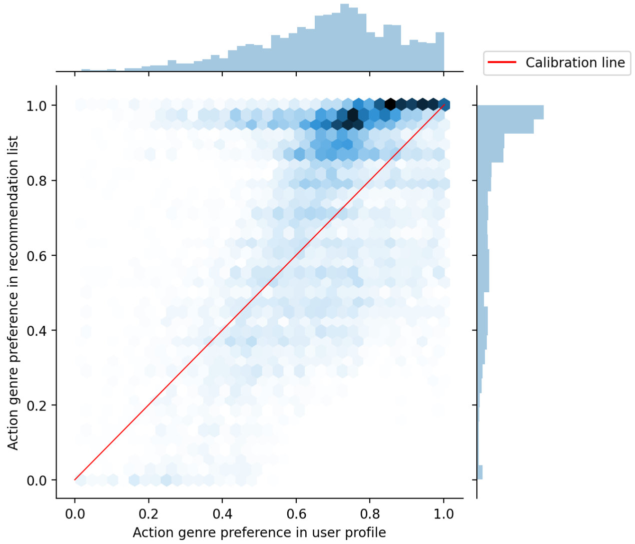

Figure 1: Miscalibration of the action genre in movie recommendation (MovieLens)

The key objective of this research is to understand the factors that are influencing miscalibration in recommender systems. As far as we know, this paper is among the first works to conduct the analysis necessary for a more fundamental understanding of the miscalibration effect. In particular, we focus on certain characteristics of users that have been considered important factors in the behavior of recommender system [13, 35]. We consider several categories of profile characteristics, including similarity to average user, propensity towards popular items, profile diversity, and preference intensity. We empirically address the following three research questions with extensive experiments on two data sets from different domains.

- **RQ1** To what extent do different recommendation algorithms exhibit preference propagation and what is the impact on calibration?
- **RQ2** Are some users more likely to receive more miscalibrated recommendations?
- **RQ3** If the answer to RQ2 is "yes", then what are the key characteristics of those users who are likely to have miscalibrated recommendations?

The rest of the paper is organized as follows. We first discuss related work in Section 2, followed by our methodology in Section 3, including the miscalibration metric and the experiment design. Then, we describe a collection of factors that are associated with miscalibration in Section 4. We present our experimental results in Section 5, exploring the impact of these factors across different algorithms and data sets. We discuss our findings in Section 6 and conclude in Section 7.

2 RELATED WORK

2.1 Bias Propagation

Lin et al. [24] explored the propagation of bias in different collaborative filtering (CF) algorithms and how this phenomenon affects user groups differently. It used *bias disparity* metric proposed in [33] to measure the change in the preference bias in the MovieLens data set. This work demonstrated that recommendation algorithms generally distort preference biases present in the input data and do so in sometimes unpredictable ways. Different groups of users may be treated in quite different ways as a result. Generally, the neighborhood-based methods propagate and strengthen biases, consistent with the findings of [19]. They not only prioritize the preferences of the dominant user groups, but also amplify the biases for the dominant group across all users. In [9], Ekstrand et al. also concluded that in collaborative filtering algorithms, users from different demographic groups do not receive the same quality recommendations.

The previous work in [24], however, indicated that matrix factorization methods were unpredictable, some of them dampening the bias propagation while others remaining well-calibrated. However, this was contradictory to what [10] found. They looked into the author gender distribution of CF algorithms in user profiles in the BookCrossing data set and compared it with that of the output recommendations. The results suggested that matrix factorization methods strengthen the biases more whereas, [24] showed the opposite trends possibly indicating the different behavior of algorithms in different domains and data sets.

Celma et al. [7] have discussed the problem of popularity bias, where a small set of popular items may dominate the recommendation lists compared to the newly-arrived or niche items. This might cause an unfair treatment of the item-providers, e.g. sellers, where one group of sellers receives more attention and sells more than the rest. Over time popular item-providers receive more reviews and become trustworthier which is similar to the phenomenon of "rich getting richer". The methods presented in [1, 20] have tried to break this feedback loop and mitigate this issue. These methods generally try to increase fairness for item providers in the system by diversifying the recommendation list of users.

The filter bubble is another common problem in recommender systems [28]. By reinforcing individual user preferences, this phenomenon limits the diversity of information to users. A recent study by Nguyen et al. [27] showed that the filter bubble effect tends to get even worse over time, and as a result exposes users to slightly narrower options. A stream of literature on user-oriented research is summarized in [23].

One limitation of the research in this area so far is that there has not been an in-depth study of the factors that contribute to miscalibration for some users but not others. In this paper we intend to identify these factors and understand the degree to which they are correlated with how users experience miscalibrated recommendations.

2.2 Fairness Metrics

As noted above, there is a close relationship between calibration and fairness, since miscalibration often disproportionately impacts users, as we show below. Various metrics have been introduced for detecting different types of fairness in classification [6, 8, 16, 21], but such metrics must be adapted for application to recommender systems.

Hardt et al. [16] discuss the equality of opportunity, and propose a metric to compare protected and unprotected groups in terms of equality in their true positive rate. In other words, they measure whether there are equal proportions of individuals from the qualified fractions of each group. This metric has inspired error-based metrics in recommender systems.

Beutel et al. [3] propose an accuracy-based, pairwise fairness metric for ranking-based recommendation algorithms where it evaluates whether a model systematically mis-ranks or under-ranks items from a particular group.

The metrics presented in [34], such as absolute unfairness, value unfairness, underestimation and overestimation unfairness, are error-based biases. As such, they focus on the discrepancies between the predicted scores and the true scores across protected and unprotected groups, considering the results to be unfair if the model consistently deviates (overestimates or underestimates) from the true ratings for specific groups.

Steck [32] has proposed a metric to measure recommendation miscalibration which is discussed in detail in Section 3. This metric measures whether recommender systems reflect the various interests of users relative to their initial preference proportions using the Kullback-Leibler (KL) divergence.

The bias disparity metric proposed by Tsintzou et al. [33], is similar in logic to the metric proposed in Steck's work. They both have user-centric points of view and they both calculate the difference between the preference of the user in the input data and the predicted preference of the user by the recommendation algorithm. The bias disparity metric looks at these differences in a more fine-grained way, evaluating the preferences of specific user groups for specific item categories. The KL-divergence used in Steck's approach, however, measures the difference in preference distributions across item categories more generally than in [33]. In this work, we use the miscalibration metric proposed by Steck.

2.3 User Characteristics

User characteristics are an important facet in understanding issues or challenges in the recommender systems, such as long-tail phenomenon [35], influential users [13], diversity [11], gender discrimination [25], and popularity bias [2].

In [35], authors initially proposed entropy-based metrics, which were applied in their proposed Absorbing Cost algorithms to improve the effectiveness of long tail recommendation. They defined item-based user entropy and topic-based user entropy, and evaluated the broadness and intensity of user's preference at the item and the item category levels, respectively. The user entropy concept is extended to other investigations in recommender systems research [11, 25].

Eskandanian et al. [13] used a series of user-side factors to identify the influential users in a recommender systems, including both implicit and explicit features. Similar factors are also used in other recommender system studies [25]. Also, user characteristics related to popularity bias research, like the average recommendation popularity in user's profile or the popular item ratio, are also taken as important user factors [2].

For our explorations of user characteristics and miscalibration, we extracted the user characteristics from user profiles, and where factors were highly correlated, retained a single representative factor.

3 METHODOLOGY

3.1 Miscalibration Metric

As noted earlier, to evaluate the propagation effect we use the miscalibration metric proposed in [32]. It uses Kullback-Leibler (KL) Divergence to measure the difference between the preference distribution across all the item categories in a user profile and the distribution in the user's recommendation set. Both are based on the distribution of item categories c for each item i, denoted by $p(c|i)$.

- $p(c|u)$: the distribution over categories c of the set of items $\mathcal{H}_u$ interacted with by user u in the past.

$$p(c|u) = \frac{\sum_{i \in \mathcal{H}_u} w_{u,i} \cdot p(c|i)}{\sum_{i \in \mathcal{H}_u} w_{u,i}}, \tag{1}$$

where $w_{i,u}$ is the weight of each item i, e.g. how recently it was liked or clicked on, or its popularity or rank.

- $q(c|u)$: the distribution across categories c of the list of items recommended to user u.

$$q(c|u) = \frac{\sum_{i \in \mathcal{I}_u} w_{r(i)} \cdot p(c|i)}{\sum_{i \in \mathcal{I}_u} w_{r(i)}}, \tag{2}$$

where $\mathcal{I}_u$ is the set of recommended items and $w_{r(i)}$ is the weight of an item and can be measured by its rank $r(i)$ in the recommendation list.

KL-divergence [22] is used to measure the difference between these two probability distributions, or the divergence of p from q. KL-divergence is denoted by:

$$MC_{KL}(p||q) = KL(p||\tilde{q}) = \sum_{c \in C} p(c|u) \log \frac{p(c|u)}{\tilde{q}(c|u)}, \tag{3}$$

where $p(c|u)$ is the target distribution. If q is similar to p, MC_{KL} will take small values, and in the case of perfect calibration, it is 0. MC_{KL} diverges if a category c is $q(c|u) = 0$ and $p(c|u) > 0$, so instead we use:

$$\tilde{q}(c|u) = (1 - \alpha) \cdot q(c|u) + \alpha \cdot p(c|u), \tag{4}$$

where $0 < \alpha < 1$, so that $q \approx \tilde{q}$. We set $\alpha = 0.01$ in this experiment.

We rename this metric to MC_{KL} instead of C_{KL} which is described in [32], since it specifies the degree to which we have miscalibration in our recommendations and it is more in line with the values that KL-divergence takes. For example, if p and q are very similar, KL-divergence takes lower values, so miscalibration is low and vice versa.

One properties discussed in [32] is worth mentioning. MC_{KL} is sensitive to small differences when p is small. For example, if a user liked a category 2% of the time and it is recommended to her 1% of the time, MC_{KL} considers it a significant change compared to a situation where a user likes a category 50% of the time, while it's recommended to her 49% of the time.

3.2 Algorithms

The experiments were performed using the `librec-auto` experimentation platform [5], which is a python wrapper built around the Java-based LibRec [15] recommendation library. All experiments were performed using a 5-fold cross validation setting where 80% of each user's rating data is used for the training data set and the rest is used as the test data set (LibRec's `userfixed` configuration). We performed our experiments on several collaborative filtering algorithms, including standard user-based and item-based k-Nearest-Neighbor approaches, Bayesian Personalized Ranking method (BPR) [30], Weighted Regularized Matrix Factorization (WRMF) [17], and the Most-Popular algorithm as the baseline. We evaluated the algorithms based on the normalized Discounted Cumulative Gain (nDCG) of the top 10 recommended items, and for each algorithm we chose the hyperparameters that achieved the highest performance. The optimal hyperparameters are described in more detail in Table 2. In the final results, both nDCG and miscalibration were averaged over all the splits in cross-validation.

3.3 Data sets

We ran our experiments on two data sets from different domains, the MovieLens 1M[1] data set and the Yelp.com[2] data set.

In the movie domain, we used the original MovieLens 1M data set. It contains 1,000,209 ratings from 6,040 users over 3,706 movies, covering 18 movie genres (e.g. "Animation", "Comedy"). The sparsity of MovieLens is 4.47%. All of the users in the MovieLens data set rated at least 20 movies.

The original Yelp.com data included 6.7 million reviews from 1.6 million users on 192,609 businesses from different industries. We used a filtered subset, Yelp_core40 [26], which includes users who rated at least 40 businesses and businesses which were rated by at least 40 users. In the Yelp_core40, there are many business categories (e.g. car dealers, airports, or different types of restaurants, etc.). We limited the data set to restaurant/food establishments, using both "restaurant" and "food" labels for filtering. After preprocessing, the data set included 85,041 ratings by 1,355 users over 1,077 restaurants, covering 231 categories. The sparsity of Yelp is 5.83%. The categories are defined by the labels provided in the original Yelp data set. Some of the categories are very popular (e.g. "American (New)", "Breakfast & Brunch", etc.), while some of them are extremely specific (e.g. "Pretzels", "Shaved snow", etc.). There were also many establishments with multiple labels, which meant that they belonged to different item categories.

3.4 Experimental Design

Items in both data sets normally have multiple labels. For example, a restaurant could be labeled as "American(New)", "Breakfast & Brunch", and "Burger," while a movie could be "Fantasy" and "Sci-Fi" (like Star Wars). This phenomenon is common in real-world scenarios. For an item i belonging to multiple item categories, uniform probabilities $p(c|i)$ are assigned to each category that item i belongs to so that $\sum_{c \in C} p(c|i) = 1$. The $p(c|i)$ is used to represent a user's preference and used later for miscalibration evaluation.

To investigate miscalibration effect across algorithms (RQ1), we compared the miscalibration using both mean with 95% confidence interval as well as the metric distribution over the whole population. Furthermore, we also studied the relationship between the degree of miscalibration and the accuracy in recommendations.

To find out whether some users consistently received more miscalibrated recommenders (RQ2), users were grouped for each algorithm. For each algorithm, the users were partitioned by dividing the whole population into three groups, referred as lower MC (25% population), mid MC (50%), and high MC (25%), in ascending order of their miscalibration values. The high MC group is our key target for this research and we compared the high MC user groups across different algorithms from the two data sets.

To determine whether specific user characteristics (factors) are more likely to result in miscalibrated recommendations across algorithms (RQ3), we used a decision tree regressor model with variables corresponding to different user characteristics as well as the miscalibration as the predicted variable. We also performed a correlation analysis between user characteristics and miscalibration. The two

[1] https://grouplens.org/data sets/movielens

[2] https://www.yelp.com/data set

approaches present different perspectives on the relationship between dependent and independent features. The important features selected by the decision tree regressor do not necessarily have a strong correlation with the dependent variable.

Considering that the feature importance measures obtained from the tree regression model can only be reliable when the model has suitable hyperparameters, all the decision tree regressor models were tuned by a grid search with 3-fold cross validation. The feature importance values were computed using Gini index, which is the normalized reduction of the criterion brought by the feature. For both data sets, the tuned decision tree regressors used "Friedman MSE" [14] as the criterion, measuring the quality of a split. The correlation matrix, on the other hand, used Pearson's correlation between factors and miscalibration value.

4 USER CHARACTERISTICS

One approach to understanding the impact of miscalibration is to look at user demographic characteristics. For example, [10] looked at the gender of users. However, the miscalibration effect is not only associated with demographic characteristics. In this paper, we are interested in the intrinsic characteristics of user profiles that contribute to miscalibration. These characteristics are all extracted from user profiles, which means they are available as long as the user's input profile (e.g. movie rating history) is accessible. This accessibility is a key advantage over the demographic characteristics, which are often unavailable due to user privacy consideration. Furthermore, the profile-based characteristics show the potential to understand the underlying reasons behind the miscalibration. These characteristics are listed in Table 1 and discussed below in more detail.

Category	Factor#	Factor Meaning
Distance to Majority	1	Category-wise preference
	2	Item-wise preference
Closeness to Popularity	3	Average profile popularity
	4	Popular item ratio
Diversity and Intensity	5	Category-wise profile entropy
	6	Item-wise profile entropy
	7	Intra-list distance
Other Profile Feature	8	Profile size

Table 1: Table of User Characteristics (Factors)

4.1 Distance to Majority

Lin et al. [24] conducted two experiments with Movie rating data (MovieLens 1M) over selected genres and analyzed the propagation effects at different population levels. Their study showed that preference propagation may be related to the distance from a user's preference to that of the majority. Inspired by this research, we included user factors that are related to the distance of the user's profile to the majority from two perspectives, with our initially proposed approach evaluating the distance from item category side.

Generally, the "majority" can be defined in two ways, depending on how one aggregates user data. One approach is to average the difference between the target individual and all other users. The other approach is to define an average user based on the whole population and then to calculate the difference between the target individual and the average user. In our investigation, we implemented both approaches and unsurprisingly, the results from the two approaches were highly correlated (0.99 in MovieLens and 0.98 in Yelp). Considering the computational cost of the first approach, we used the latter approach in this paper.

We consider two factors that capture the notion of distance to majority.

4.1.1 Factor 1 (F1): Category-wise. Factor F1 measures the difference of preference distribution between each individual user and the average user, which is calculated using Jensen-Shannon (JS) divergence. Compared to Kullback-Leibler (KL) divergence, the JS-divergence, shown in Equation 5, is a normalized and symmetric metric ranging from 0 to 1, which is an important property to evaluate the distance between an individual user to the majority. The value 0 means that the two distributions are identical. Lower JS-divergence value means that users tend to have similar preferences across the data set.

$$D_{JS}(u,v) = \frac{1}{2}[D_{KL}(u||avg_{u,v}) + D_{KL}(v||avg_{u,v})] \tag{5}$$

where:

$$avg_{u,v}(c) = \frac{1}{2}(p(c|u) + p(c|v)) \tag{6}$$

The D_{KL} is mainly based on Equation 3 with v standing for average user and u as the target user. Their preference $p(c|u)$ and $p(c|v)$ on item category c in this and the following factors are calculated by Equation 1.

4.1.2 Factor 2 (F2): Rating-wise. The ratings-wise distance to the majority applied in [13] is computed by the Euclidean distance between the individual and the average user, which is represented by the mean ratings vector among all users.

4.2 Closeness to Popularity

Previous research has shown that item popularity has an influential effect on the recommendation results. Recommender systems, in general suffer from popularity bias [19] that not only affects the recommendations of all the users but also reinforces the bias even more. Dominance of popular items diverges users' recommendations from their actual preference and causes miscalibration [2]. To take this effect into consideration, we included two factors representing how much a user prefers popular items.

4.2.1 Factor 3 (F3): Average Recommendation Popularity (ARP). Jannach et al. [18], used three ways to measure *popularity*. In this paper, we use the average recommendation popularity for each user. Each item has a corresponding popularity value based on the number of available ratings for the item. For each user profile, the ARP is the average popularity value of all the rated items in user's history profile.

4.2.2 Factor 4 (F4): Popular Item Ratio. This factor presents the intuitive understanding of how much a user prefers popular items and is determined by the percentage of popular rated items across all rated items in a user's profile. Following the previous work [1, 4],

Algorithm	Dataset	HyperParameter	nDCG	MisCalibration(MC)	95% MC Interval
Most-Popular	MovieLens	NA	0.203	1.104	(1.092, 1.115)
	Yelp	NA	0.065	1.593	(1.581, 1.604)
UserKNN	MovieLens	k=200, sim=cosine	0.277	0.850	(0.843, 0.857)
	Yelp	k=20, sim=jaccard	0.127	1.532	(1.522, 1.542)
ItemKNN	MovieLens	k=200, sim=cosine	0.256	0.892	(0.884, 0.900)
	Yelp	k=200, sim=jaccard	0.121	**1.524**	(1.514, 1.535)
BPR	MovieLens	f=100	0.386	0.747	(0.742, 0.752)
	Yelp	f=200	0.125	1.530	(1.520, 1.539)
WRMF	MovieLens	f=50	**0.387**	**0.739**	(0.734, 0.745)
	Yelp	f=10	**0.140**	1.567	(1.555, 1.578)

Table 2: Models with best nDCG and MisCalibration Values

we categorized the top 20% most rated items in each data set as popular items (741 out of 3,706 items in MovieLens, and 215 out of 1,077 items in Yelp).

4.3 Diversity and Interest Intensity

Another important intrinsic user profile characteristic is diversity. It is typically measured by entropy or intra-list distance among items in the user's profile or the recommendation lists. Entropy is a concept originated from information theory [31] and has been used in the context of recommendation diversity [35]. If the information is more uniformly distributed, the entropy value would be higher. In the implementation of this concept in the recommender system, a higher entropy means that the user has a more uniformly distributed preference profile. A lower entropy indicates that the user's preference is skewed towards specific items or item categories. So, in our context, entropy can be indicative of a user's preference intensity towards a small set of items or item categories, where a lower entropy means a higher preference intensity. We use both item-wise and category-wise user profile entropy as factors in our analysis.

4.3.1 Factor 5 (F5): User Profile Entropy – Category Wise. In the context of preference across item categories, the entropy is higher if the preferences of the user over different categories are more equally distributed. For users with lower entropy, it indicates that they have an intense preference for specific item categories.

$$E(u) = -\sum_{c \in C} p(c|u) \log p(c|u) \tag{7}$$

4.3.2 Factor 6 (F6): User Profile Entropy – Item Wise. The item-wise profile entropy is based on user's rating behavior and provides a broader perspective on a specific user's preferences. In our experiments, we tested two approaches which represented user rating behavior, with explicit rating (original rating value) and implicit rating (binary rating). Results from these approaches were highly correlated and so we retained the implicit rating based approach, which has lower computational costs, for the further analysis. For users with a higher item-wise profile entropy, it indicates they had rated more items and had broader tastes.

$$E(u) = -\sum_{i \in \mathcal{H}_u} p(i|u) \log p(i|u) \tag{8}$$

where: $p(i|u) = \frac{w(u,i)}{\sum_{i \in \mathcal{H}_u} w(u,i)}$ with $\mathcal{H}_u$ here and in the following factor presenting the rated items in the user's profile.

4.3.3 Factor 7 (F7): Intra List Distance (ILD). Our version of Intra List Distance [12] is an adjusted version of intra list similarity as introduced by Ziegler et al. [36]. Both capture the diversity of a list, and in our case, we used Euclidean distance as the underlying dissimilarity measure in ILD.

$$ILD(\mathcal{H}_u) = \frac{1}{|\mathcal{H}_u|(|\mathcal{H}_u| - 1)} \sum_{i \in \mathcal{H}_u} \sum_{j \in \mathcal{H}_u} d(i, j) \tag{9}$$

5 EXPERIMENTAL RESULTS

5.1 Miscalibration effects across algorithms

Table 2 and Figure 2 provide comparisons of miscalibration differences among algorithms. Table 2 shows the miscalibration values and related 95% confidence intervals. Figure 2 shows the density plots from the two data sets, which reveal more detailed information about the distribution of miscalibration across all the algorithms.

These results provide a partial answer to RQ1. It seems that all algorithms for both data sets are miscalibrated to different degrees. The data characteristics do, however, make a difference. For example, different algorithms in the case of MovieLens diverge much more than in the case of Yelp, with factorization-based algorithms being less prone to miscalibration than the neighborhood-based methods or the non-personalized Most-Popular algorithm. We discuss these observations further below.

To better answer RQ1 and RQ2, the users were grouped based on their miscalibration values into a low MC group, mid MC group, or high MC group. The comparison of the same grouping across different algorithms, may present the similarity between algorithms. We were primarily interested in the high MC group and compared the performance of the algorithms for this group in each data set. Figure 3 shows the ratio of shared users (among different algorithms) over the number of users in the high MC group. For example, in MovieLens (on the left of Figure 3), 52% of the users in the high MC group are the same for the algorithms UserKNN and Most-Popular. With a higher shared user ratio, the two algorithms in the paired comparison may be more similar to each other from the miscalibration perspective.

As shown in Table 2, all the miscalibration values in MovieLens are significantly different from each other at the 95% confidence

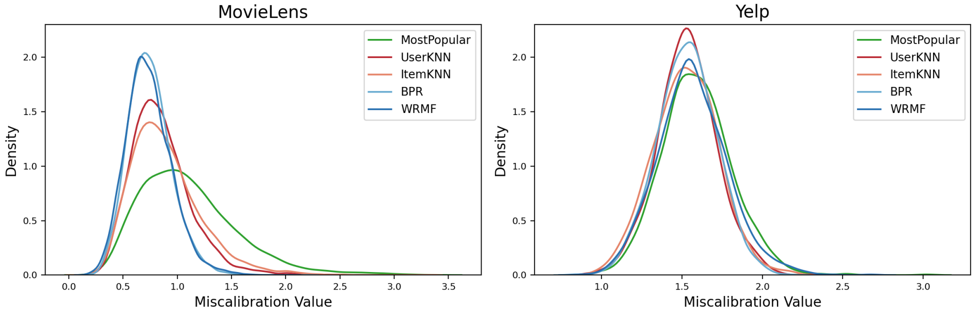

Figure 2: Miscalibration density plots of two data sets. The left is MovieLens, and the right is Yelp.

level. Overall, matrix factorization (BPR and WRMF) outperformed neighborhood-based (UserKNN and ItemKNN) algorithms, which are both better than baseline Most-Popular algorithm, from both the accuracy and calibration perspective. The distributions in Figure 2 present a similar pattern. The bell curve of the miscalibration distribution in MovieLens shows that it is normally distributed. The distribution shapes of the same recommender algorithm category, like neighborhood-based, are similar. Meanwhile, the distribution shapes of different algorithm categories are dissimilar – matrix factorization algorithms are different from neighborhood-based algorithms.

Regarding the relationship between accuracy and miscalibration, Table 2 indicates that accuracy is negatively correlated with miscalibration in MovieLens; the algorithm with a higher nDCG value always has a lower miscalibration. WRMF has the highest accuracy and also the lowest miscalibration at the 95% confidence level for MovieLens data set. As the second most accurate model, BPR is the second least miscalibrated, with less than 0.01 miscalibration increase from WRMF. For neighborhood-based approaches, UserKNN and ItemKNN have lower accuracy and are also more miscalibrated. The baseline, Most-Popular, has the least accuracy and the most miscalibration.

In MovieLens, Most-Popular, UserKNN, and ItemKNN have a high ratio of shared users. Such a phenomenon can also be seen with BPR and WRMF, which have relatively a high ratio of shared users. Among all the high MC groups from different algorithms, 114 users (1.9% of whole population) are always grouped in the high MC group for all the algorithms, which means these 114 users are consistently receiving more miscalibrated recommendations when compared to the rest of the population.

In Yelp, the least miscalibrated algorithm is ItemKNN, whose miscalibration value is not significantly different from the runner-ups, BPR and UserKNN, with the 95% confidence level. WRMF and Most-Popular are the most miscalibrated algorithms; while WRMF has a significantly lower miscalibration value than Most-Popular, it has a significantly higher value than neighborhood-based algorithms and the other matrix factorization algorithm BPR. On the right of Figure 2, we observe that the miscalibration in Yelp is also normally distributed. The distribution shapes across different algorithms overlap. Overall, the difference of the miscalibration across algorithms in Yelp is much smaller than in MovieLens, and the miscalibration difference among algorithm categories is not obvious.

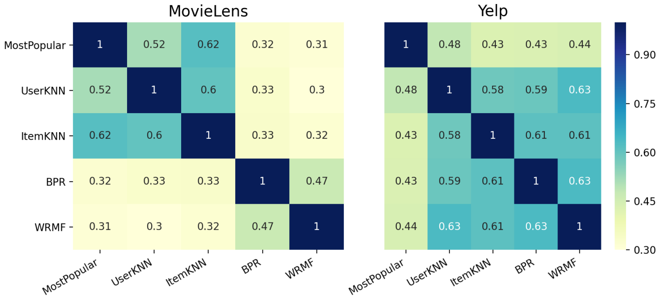

Figure 3: Co-Miscalibrated user ratio for the high Miscalibration (MC) group. The left is MovieLens with 1510 users, and the right is Yelp with 339 users.

Regarding the relationship between accuracy and miscalibration in Table 2, no significant correlation pattern is shown in Yelp. In Figure 3, the shared user ratios across all the algorithms in Yelp are high. Except Most-Popular, other ratios normally are around 60%, which means that 60% of the users are same in the high MC grouping for the algorithm pairs. Among all the population in Yelp, 76 users (5.6% of the whole population) always receive more miscalibrated recommendations than the majority, regardless of the choice of algorithm.

As noted earlier, these results underscore the conjecture that data and domain characteristics may have a significant impact on the miscalibration of different algorithms. Thus, they should be considered carefully in design decisions regarding the choice of recommendation algorithms.

5.2 Important User Factors Affecting Miscalibration

To evaluate the user profile feature importance in miscalibration, we used the decision tree regressor models to predict miscalibration value, using all the user profile factors. Table 3 shows the results of computing feature importances across all the decision tree regressor models. The columns are named after factors and the column values are the feature importance determined by Gini importance criteria as described in the experimental design section. The RMSE is the

Dataset	Algorithm	F1	F2	F3	F4	F5	F6	F7	F8	RMSE
MovieLens	Most-Popular	0.089	0.062	0.094	0.154	**0.389**	0.070	0.100	0.042	0.40
	UserKNN	0.067	0.152	0.195	0.176	**0.196**	0.087	0.066	0.062	0.29
	ItemKNN	0.060	0.143	0.116	0.190	**0.274**	0.068	0.079	0.070	0.33
	BPR	0.080	0.057	**0.293**	0.130	0.249	0.075	0.078	0.039	0.20
	WRMF	0.079	0.151	0.114	**0.231**	0.144	0.086	0.121	0.074	0.22
Yelp	Most-Popular	**0.518**	0.032	0.019	0.014	0.382	0.005	0.017	0.012	0.18
	UserKNN	0.098	0.158	0.057	0.022	**0.488**	0.115	0.045	0.016	0.18
	ItemKNN	0.101	0.073	0.048	0.055	**0.593**	0.035	0.034	0.062	0.19
	BPR	0.065	0.095	0.046	0.037	**0.620**	0.005	0.031	0.102	0.17
	WRMF	0.157	0.057	0.074	0.043	**0.438**	0.095	0.051	0.084	0.21

Table 3: Results of decision tree regressor. F5 (category-wise user profile entropy) is the most important factor for both data sets, with popularity related factors (F3 and F4) as the second most important factors for MovieLens

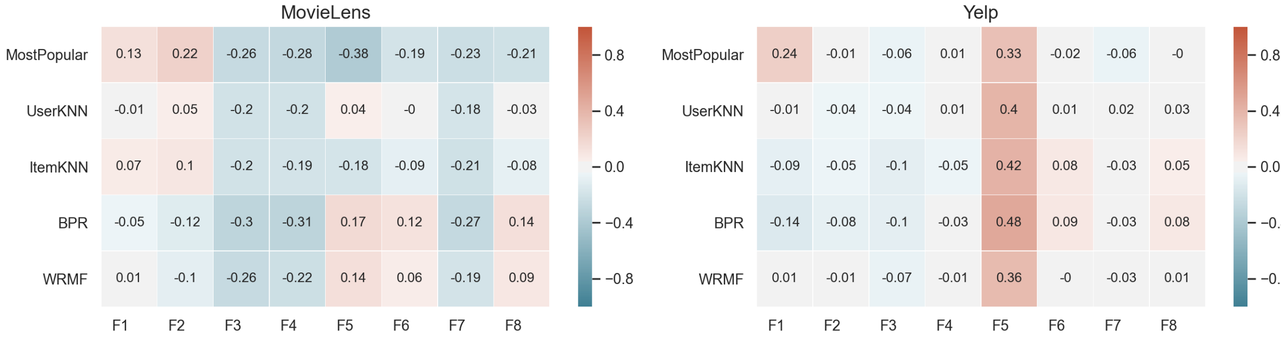

Figure 4: Correlation heatmap of factors and miscalibration from different algorithms

prediction model performance evaluated by 5-fold cross validation. The most important feature for each algorithm is highlighted in bold, while the second most important feature is underlined.

In MovieLens, the most important factors are F5 (category-wise user profile entropy), F4 (popular item ratio), and F3 (average profile popularity). The importance of F5 is consistent across all the algorithms. And also, F3 and F4 are both from the "closeness to popularity" user characteristics category.

The strong influence of F5 suggests that when users have strong exclusive preferences for a category of items (lower F5 values), recommendation algorithms find it easier to predict the user interests and thus provide calibrated recommendations. We see this effect more strongly in the MovieLens data set, probably because there are strong preferences for blockbusters and popular movies that dominate specific movie genres. For this reason, the impact of F5 factor is particularly pronounced in the Most-Popular algorithm.

Based on the correlations in Figure 4, the factors and the miscalibration do not have strong correlations. F3 and F4 are slightly negatively correlated with miscalibration. The closer the user profile is to popularity, the lower the miscalibration value is. This suggests that popularity measures of a user's profile are not as strong a predictor of miscalibration as we might have expected although the effect is present. Other researchers have generally found a strong influence of popularity effects so this finding deserves further study. For F5, the correlation directions with different algorithms are different but for most the correlation is very weak except for the Most-Popular algorithm.

As indicated in Table 3, F5 in Yelp is also the most important factor for almost all the algorithms, with an importance value as high as 0.62. This means that factor F5 itself can cause the normalized reduction of Friedman MSE by 62%.

In contrast to MovieLens, F5 in Yelp is the only factor showing high correlation across all the algorithms with significant positive correlation direction. This means that with the higher user preference intensity (lower F5), the recommendation is less miscalibrated. But the factors that depend on distance to the majority (F1 and F2) or the popularity (F3 and F4) do not show a lot of influence. We believe that this is, in part, due to a wider selection of categories in Yelp than in MovieLens. Essentially, users in Yelp are more likely to show interest in a small number of item categories and are not necessarily influenced by strong popularity bias in the data (in contrast to users in MovieLens). We discuss this phenomenon further in the next section.

6 DISCUSSION

6.1 Comparing Miscalibration Across Data sets

Comparing miscalibration in the two data sets, we observed that recommendations in MovieLens generally have a lower degree of miscalibration than in Yelp. Also, miscalibration in MovieLens

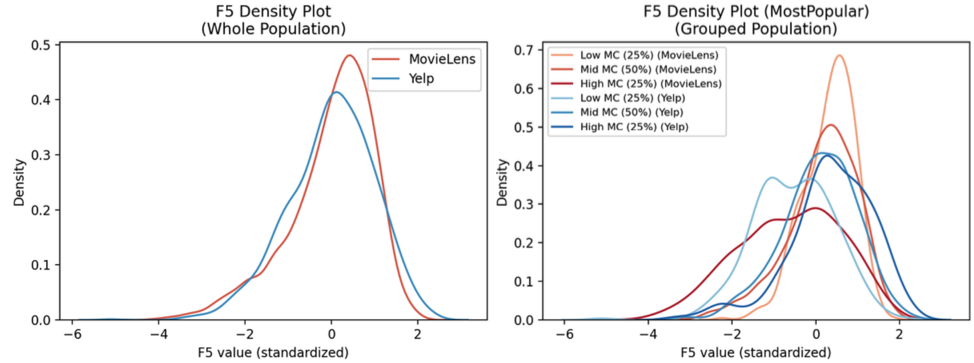

Figure 5: Standardized F5 density plot across two data sets. The left is at the whole population level, and the right is at the grouped population level based on miscalibration value for an example algorithm, Most-Popular.

shows more variance across different algorithms, while in Yelp, the miscalibration across different algorithms is almost identical.

These differences may be in part due to differences in the item category sizes across the data sets. MovieLens has 18 item categories (movie genres), while Yelp has 231 item categories (food/restaurant labels). A data set with a wide item category range like Yelp must have small input preference ratios for most categories in users' profiles. This can possibly lead to a high miscalibration if one item from these categories gets recommended. From this perspective, it may not be meaningful to compare miscalibration values across data sets with significant variations in the number of categories. But this also highlights the need to consider the characteristics of the data and those of user profiles when deciding which algorithms to use for recommendation.

6.2 Relationship between Miscalibration and Accuracy

Steck [32] has suggested that there is a trade-off between recommendation accuracy and calibration, though that study did not provide detailed analysis of such a relationship.

In our experiments, particularly on MovieLens data, we found that, in fact, there is a positive correlation between accuracy and calibration. For example, the WRMF algorithm is not only the most accurate but also the most calibrated recommender. Based on this observation, increasing the calibration degree is not necessarily at the cost of accuracy. With appropriate algorithm design, calibration and accuracy can be achieved simultaneously. However, the choice of the algorithm may be critical and the characteristics of the data set may also play a role in which algorithm may provide the best concordance between accuracy and calibration.

6.3 Factors influencing miscalibration

To further understand the effect of F5 factor on miscalibration, we investigated the differences in the two data sets using the standardized F5 distribution, which is the factor value used in the decision tree regressor. In Figure 5, we observe that the overall F5 distribution is identical for the two data sets but we see marked differences when we examine distributions for different user groups. This indicates that the different effects of F5 in the two data sets are likely due to domain characteristics.

We already noted the differences of the item category size in the two data sets. Another notable difference between the two data sets is the degree to which popularity bias impacts recommendations. For example, in Yelp we observe that smaller neighborhood size works better for the UserKNN model. Furthermore, the Most-Popular algorithm performs poorly when compared to MovieLens. These observations indicate that the Yelp data set may be less prone to popularity bias. Users in Yelp have more unique tastes, and prefer more small but varied sets of item categories. For these reasons, the recommendations are not less influenced by a dominant majority user group or by the popularity bias. The reduced influence from majority and popularity explains why the importance of F5 is much higher than other factors for miscalibration in Yelp, where other factors are highly related to the majority and popularity.

In contrast, in MovieLens data, there is a greater degree of influence by a more cohesive majority user group as well as by dominant popular movies. This is clear from the strong performance of the baseline Most-Popular algorithm. This favorability towards popularity and majority are captured by the corresponding factors. For Most-Popular and ItemKNN, users with higher preference intensity (lower F5) have more miscalibrated recommendations, while for BPR and WRMF, users with higher preference intensity (higher F5) have more calibrated recommendations. Most-Popular and ItemKNN provide inconsistent directions from Yelp. These two algorithms are most easily influenced by popularity bias [19].

Another important domain consideration is the idea of item availability. In the era of home video, many movies are available to be watched at will, regardless of their provenance. Thus, many users in the MovieLens data set would have had the same potential set of movies available to them[3]. Establishments in the Yelp data set, on the other hand, are fixed to particular localities and users may not have equal access to them. For example, a user in a big city might have access to a broad set of restaurants, whereas a user in a small town might have more limited opportunities to try a

[3] Note that Netflix was renting DVDs nationwide by mail in 1999, the year that collection for the MovieLen 1M data set began.

range of cuisines. There may be less of a "win" for an algorithm in optimizing for popularity in such a domain.

7 CONCLUSION AND FUTURE WORK

Miscalibration in recommendation results derives from a conjunction of different facets of the recommender system environment, including algorithm design, user profile characteristics, and data set characteristics. In this paper, we focused on user profile characteristics and investigated the impact of these characteristics on the miscalibration witnessed in different algorithms and data characteristics. We found that the category-wise user profile entropy factor (F5), which indicates a user's preference intensity on a small number of categories, has great influence on miscalibration.

But we also observed that depending on certain data characteristics (such as the number and size of item categories and the degree of popularity bias in the data), other user profile characteristics such as closeness to majority or preference towards popular items can emerge as influences on recommendation miscalibration. Further exploration of the influence from these factors is part of future work. For example, we plan on expanding our research to address the potential influence from the difference of item category characteristics. In doing so, we will conduct experiments on data sets with similar number of categories.

Overall, our analysis shows that considering these intrinsic user characteristics, as well as the overall data, characteristics are critical in the choice of appropriate algorithms in recommender systems. Through this type of analysis, system designers might be able to predict and mitigate miscalibrated recommendations for different user groups based on their profile characteristics, possibly by using different combinations of algorithms for different users. Development and evaluation of such recommender systems will be the subject of future research.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 42–46.

[2] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The impact of popularity bias on fairness and calibration in recommendation. *arXiv preprint arXiv:1910.05755* (2019).

[3] 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.

[4] Erik Brynjolfsson, Yu Jeffrey Hu, and Michael D Smith. 2006. From niches to riches: Anatomy of the long tail. *Sloan Management Review* 47, 4 (2006), 67–71.

[5] Robin Douglas Burke, Masoud Mansoury, and Nasim Sonboli. 2020. Experimentation with Fairness-Aware Recommendation Using Librec-Auto: Hands-on Tutorial. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) *(FAT* '20)*. Association for Computing Machinery, New York, NY, USA, 700. https://doi.org/10.1145/3351095.3375670

[6] Toon Calders and Sicco Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery* 21, 2 (2010), 277–292.

[7] 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.

[8] Cynthia S Crowson, Elizabeth J Atkinson, and Terry M Therneau. 2016. Assessing calibration of prognostic risk scores. *Statistical methods in medical research* 25, 4 (2016), 1692–1706.

[9] 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*. 172–186.

[10] 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*. ACM, 242–250.

[11] 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, 280–284.

[12] Farzad Eskandanian, Bamshad Mobasher, and Robin D Burke. 2016. User Segmentation for Controlling Recommendation Diversity.. In *RecSys Posters*.

[13] 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*. 225–233.

[14] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning*. Vol. 1. Springer series in statistics New York.

[15] Guibing Guo, Jie Zhang, Zhu Sun, and Neil Yorke-Smith. 2015. LibRec: A Java Library for Recommender Systems.. In *UMAP Workshops*, Vol. 4.

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

[17] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets.. In *ICDM*, Vol. 8. Citeseer, 263–272.

[18] Dietmar Jannach, Lukas Lerche, Fatih Gedikli, and Geoffray Bonnin. 2013. What recommenders recommend–an analysis of accuracy, popularity, and sales diversity effects. In *International conference on user modeling, adaptation, and personalization*. Springer, 25–37.

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

[20] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2014. Correcting Popularity Bias by Enhancing Recommendation Neutrality.. In *RecSys Posters*.

[21] 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.

[22] Solomon Kullback. 1997. *Information theory and statistics*. Courier Corporation.

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

[24] Kun Lin, Nasim Sonboli, Bamshad Mobasher, and Robin Burke. 2019. Crank up the volume: preference bias amplification in collaborative recommendation. *arXiv preprint arXiv:1909.06362* (2019).

[25] Masoud Mansoury, Himan Abdollahpouri, Jessie Smith, Arman Dehpanah, Mykola Pechenizkiy, and Bamshad Mobasher. 2020. Investigating Potential Factors Associated with Gender Discrimination in Collaborative Recommender Systems. *arXiv preprint arXiv:2002.07786* (2020).

[26] Masoud Mansoury, Bamshad Mobasher, Robin Burke, and Mykola Pechenizkiy. 2019. Bias disparity in collaborative recommendation: Algorithmic evaluation and comparison. *arXiv preprint arXiv:1908.00831* (2019).

[27] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*. 677–686.

[28] Eli Pariser. 2011. *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin.

[29] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. 2017. On fairness and calibration. In *Advances in Neural Information Processing Systems*. 5680–5689.

[30] 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*. AUAI Press, 452–461.

[31] Claude E Shannon. 1951. Prediction and entropy of printed English. *Bell system technical journal* 30, 1 (1951), 50–64.

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

[33] Virginia Tsintzou, Evaggelia Pitoura, and Panayiotis Tsaparas. 2018. Bias Disparity in Recommendation Systems. *arXiv preprint arXiv:1811.01461* (2018).

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

[35] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. 2012. Challenging the long tail recommendation. *arXiv preprint arXiv:1205.6700* (2012).

[36] 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*. 22–32.