How to Diversify any Personalized Recommender?

Manel Slokom[0000−0002−9048−1906], Savvina Daniil[0000−0001−8888−2869], and
Laura Hollink[0000−0002−6865−0021]

Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
manel.slokom@cwi.nl, Savvina.Daniil@cwi.nl, laura.hollink@cwi.nl

Abstract. In this paper, we introduce a novel approach to improve the
diversity of Top-N recommendations while maintaining accuracy. Our
approach employs a user-centric pre-processing strategy aimed at expos-
ing users to a wide array of content categories and topics. We personalize
this strategy by selectively adding and removing a percentage of interac-
tions from user profiles. This personalization ensures we remain closely
aligned with user preferences while gradually introducing distribution
shifts. Our pre-processing technique offers flexibility and can seamlessly
integrate into any recommender architecture. We run extensive experi-
ments on two publicly available data sets for news and book recommen-
dations to evaluate our approach. We test various standard and neural
network-based recommender system algorithms. Our results show that
our approach generates diverse recommendations, ensuring users are ex-
posed to a wider range of items. Furthermore, using pre-processed data
for training leads to recommender systems achieving performance levels
comparable to, and in some cases, better than those trained on original,
unmodified data. Additionally, our approach promotes provider fairness
by facilitating exposure to minority categories.[1]

Keywords: Recommendations · Diversity · Provider Fairness · Pre-
processing · User-centric

1 Introduction

In today's digital age, personalized recommender systems form a solution to the
information overload problem. They filter the enormous amount of information
available online to only the top-N items - e.g. news articles, or books - that are
most relevant to a user [18,21]. While personalized recommendations ensure rele-
vance to users' interests, excessive personalization risks closing users within filter
bubbles [24,22]. Filter bubbles are environments created by personalization algo-
rithms, where users are exposed solely to familiar information or opinions. This
can potentially harm the democratic process and exacerbate polarization [24].

To address this, diversity in recommender systems has become an active re-
search area. This trend aligns with a larger movement in the community towards

---

[1] Our GitHub code is available at: https://github.com/SlokomManel/
How-to-Diversify-any-Personalized-Recommender-

evaluating not only accuracy but also other metrics, which we refer to as 'beyond accuracy', such as diversity, privacy, and fairness [16,45,42]. Although diversity and provider fairness are rooted in distinct normative considerations [9], diversity aims to present a wide range of items, while provider fairness ensures equitable exposure for each content source [19], in our case increasing diversity also improves provider fairness by enhancing the visibility of items from minority providers. Several approaches have emerged to improve beyond-accuracy metrics. In a common categorization, we distinguish between three types of approaches [45]: (i) pre-processing, (ii) in-processing, and (iii) post-processing. Pre-processing approaches involve modifying input data - such as user profiles - before the recommendation process. Several works use pre-processing to improve fairness [26,10,4,3]. [30] propose pre-processing to ensure privacy. In-processing approaches build beyond accuracy metrics directly into the recommendation algorithm itself, e.g. through multi-objective optimization [46,32,27,23,13,40]. Post-processing approaches such as re-ranking have been used for both diversity [20] and fairness [43,44]. We notice that the majority of research into diversity has focused on in-processing and post-processing approaches; relatively little is known about the use of pre-processing to increase diversity.

In this paper, we fill this gap and present a pre-processing approach to diversify the output of a recommender system. We alter user profiles that are input to the system to achieve greater diversity in the recommender's output. Pre-processing is interesting in this respect, as such an approach can be combined with a wide range of algorithms. Any personalized recommender architecture takes a user profile as input, and can thus be used in combination with our approach to diversification. In comparison, in-processing approaches are bound to a specific algorithm [45]. In addition, pre-processing allows diversification beyond what would have been recommended by the initial approach. In contrast, the effect of post-processing is independent of the model and may be constrained by the set of items the algorithm can recommend. Our approach is user-centric, focusing on the personalized addition of items to user profiles. This is because the input data for recommender systems is highly personal. A user profile is more than a list of past interactions with items; it encodes parts of the user's preferences, abilities, and characteristics. Altering a user profile risks diluting this rich encoding of the user's taste. By adopting a personalized approach, we mitigate this risk, maintaining accuracy without compromising the integrity of the user profile. We opt to alter the user profile directly instead of altering processed versions of it, e.g. a latent representation. This increases the potential for transparency and explainability, allowing users to review their (altered) profiles.

Our main research question focuses on how to alter user profiles to improve the diversity of top-N recommendations without compromising recommendation accuracy. We present two variants of our approach: one involves adding interactions to user profiles, while the other includes both adding and removing interactions to remain close to the original profile size. We explore different levels of manipulations ranging from 1% to 10%. We perform extensive evaluations to test which variant achieves optimal accuracy and diversity. We evaluate the

effectiveness of our approach on two public data sets: MIND for news recommendation, where we aim to diversify news categories, and GoodBook for book recommendation, where we aim to diversify genres. As mentioned above, our approach can be integrated with any personalized recommender system. In this study, we test combinations with seven algorithms ranging from collaborative filtering (CF) to neural network recommenders.

We report accuracy (MRR, nDCG), normative diversity (calibration via divergence), descriptive diversity (coverage, Gini index), and provider fairness (fair-nDCG). Our findings indicate that using pre-processed data for training can lead to recommender systems achieving comparable or improved accuracy compared to those trained on original data. Regarding diversity, calibration metrics consistently improved for certain algorithms, while coverage and the Gini index showed mixed results across different levels of pre-processing. Additionally, pre-processed data consistently resulted in higher fair-nDCG scores, indicating enhanced exposure fairness and better representation of minority categories.

2   Background and Related Work

This section, first, provides an overview of existing work on pre-processing approaches to increase beyond-accuracy factors. Then, we examine existing measures used to evaluate diversity in recommender systems.

2.1   Pre-processing approaches

Pre-processing approaches involve modifying input data before feeding it into the recommender system [45]. These approaches offer flexibility, as they can be integrated with any recommendation algorithm without altering its core design [21,3]. Inspired by data poisoning attacks, [26] introduces "antidote" data by simulating new users and ratings in the training set to improve consumer fairness in matrix factorization models. Similarly, [30,37] proposes a gender obfuscation method to protect user privacy while preserving recommendation quality and reducing gender stereotypes. Balancing user groups through re-sampling is another common technique. [10] creates gender-balanced training data by adjusting the user distribution. While this reduces recommendation accuracy slightly, it mitigates gender disparities in data sets like Movielens. [4] explores provider fairness by up-sampling interactions favoring minority groups and shows improvements in coverage and reduced disparate impacts without sacrificing accuracy. Our study builds on this line of work by proposing a personalized pre-processing method that selectively extends user profiles aiming to enhance diversity without compromising recommendation accuracy.

2.2   Measuring diversity

Diversity in recommender systems was initially defined as the opposite of similarity, i.e., $1 - \text{similarity}$ [31]. In [47], Intra-List Similarity (ILS) was introduced

to measure the average diversity within a recommendation list, independent of the item order. Other metrics include expected intra-list diversity, which is sensitive to item ranking, and aggregate diversity, which measures the diversity of items across all user recommendations [35], and the Gini index quantifying distributional inequality [5]. Item coverage is also used as a descriptive measure of diversity, calculated as the proportion of unique items recommended across users relative to the entire catalog. While traditional descriptive diversity metrics provide useful insights, they only partially address the broader challenges in recommender systems [34]. Recent research has extended the focus to normative diversity, emphasizing exposure diversity based on principles tailored for specific domains. In [14], the authors outline principles for achieving exposure diversity in recommendations, and [36] introduces RADio, a rank-aware divergence framework designed to evaluate diversity according to normative goals. In this study, we use both descriptive and normative metrics.

3   User-Centric Data Pre-processing

The objective of our personalized pre-processing approach is to increase the recommendation diversity while maintaining the accuracy. For this purpose, we alter user profiles through strategic addition and removal of interactions. Our approach is founded on two principles: (1) that items added to the user profile should be sufficiently different from what a user has interacted with before, and (2) that those items should closely align with the user's preferences. Figure 1 illustrates our approach. Three key inputs drive this process. First, each item in the system is associated with one or more categories ($Item\_categories$). Second, we use logistic regression to identify the categories most representative of each user, based on their past interactions ($User\_categories$). Lastly, we apply userKNN to generate an initial list of items that might be relevant for each user ($Initial\_predictions$). Based on these three inputs, we generate a personalized list of items that can be added to the user's profile ($Personalized\_items\_for\_addition$). For this list, we select items from the user's $Initial\_predictions$ that are associated with categories not in their $User\_categories$. Section 3.1 further explains how we arrive at personalized lists of items for addition.

Our approach has two variants. The one-step variant adds interactions to a user's profile. The amount of added items is determined by a parameter λ. The two-step variant, on top of that, also removes interactions from a user's profile. After adding items as in the one-step approach, this variant randomly removes items that were not recently added. This ensures that the profile size remains approximately equal to its original size. Section 3.2 describes the two variants.

3.1   Creating personalized lists of items for addition

Here, we explain how we create $User\_categories$, $Initial\_predictions$, and based on that, $Personalized\_items\_for\_addition$.
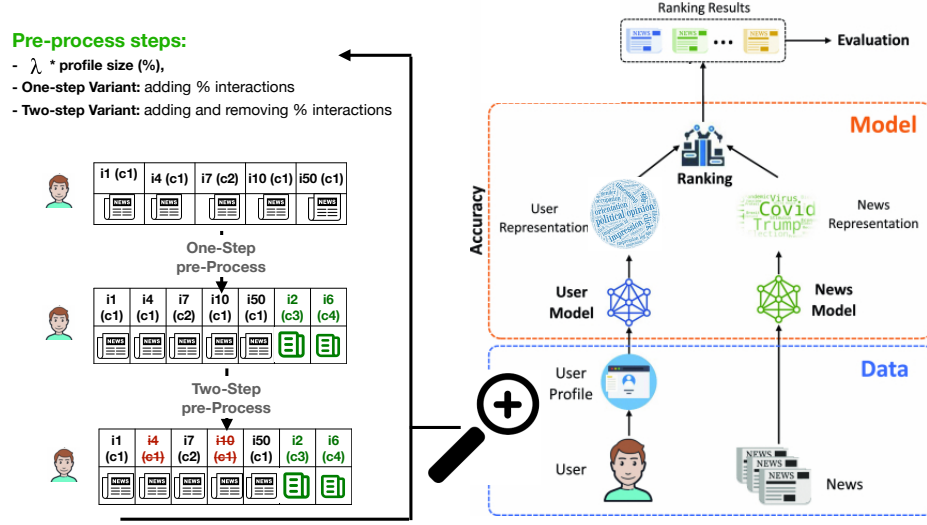
Fig. 1: Our pre-processing approach and its role in a recommendation architecture. $\{i1, i4, i7, i10, i50\}$ are items in user u's past interactions and $\{c1, c2, c3, c4\}$ are categories of these items. In the one-step variant, personalized items $i2$ and $i6$ are added (green). In the two-step variant, items $i4$ and $i10$ are removed (red).

User_categories For each user, we create a *User_categories* list that contains categories that are representative of that user. For this purpose, we train a logistic regression model on labeled training data. We choose logistic regression for its simplicity and interpretability, allowing us to easily understand the relationship between user profiles and categories. Additionally, its coefficients help identify only the truly associated categories, preventing overgeneralization. Lastly, logistic regression has shown strong performance on similar tasks in prior research [37,30]. The input for the model is the initial user-item matrix $\mathcal{D}$, and the target is whether an item belongs to a specific category. We transpose the input data $\mathcal{D}$ into an item-user matrix X_{iu}, which indicates whether an item (in rows) i has interacted with user u. We apply one-hot encoding to the target feature, i.e., categories. Then, we have binary category labels $\mathbf{y}^c$, such that for each category c, a binary vector $\mathbf{y}^c$, where $y_i^c = 1$ if item i belongs to category c, and $y_i^c = 0$ otherwise. For each category c, we train a logistic regression:

$$P(y_i^c = 1 \mid X_{i1}, X_{i2}, \ldots, X_{iM}) = \frac{1}{1 + e^{-(\beta_0^c + \beta_1^c X_{i1} + \beta_2^c X_{i2} + \ldots + \beta_M^c X_{iM})}}$$

Where: y_i^c is the binary label indicating if item i belongs to category c. X_{iu} is the interaction of item i with user u. β_0^c is the intercept term for category c. β_u^c is the coefficient for the interaction of user u with items in category c. The β_u^c coefficients capture the extent to which users are correlated with categories. Users with higher positive coefficients are more indicative of the category; we select user-category pairs with coefficients above 0 for the *User_categories* lists.

Initial_predictions The goal here is to predict a list of items that a user might be interested in interacting with, based on the initial user-item matrix $\mathcal{D}$. Any recommender system could be used for this step. We use an imputation approach to fill in the missing values in the user-item matrix [17]. The goal of the imputation is to infer missing values in the data in such a way that improves the overall performance of recommender systems trained on that data [33,30]. Specifically, we impute to derive a confidence score that allows us to choose the items that will be added to the profile. For simplicity, we opt for a userKNN-based approach, as implemented in [29]. We use the number of neighbors that are used for each prediction as a relevance score. This enables us to account for the fact that some users may have more neighbors available for some items than others. We fix a cutoff k, to ensure that only the most relevant items are added to the list of initial predictions for each user. Algorithm 1 (first part) summarizes this process.

Algorithm 1: Personalized list of indicative items per user.

Input:
- User-item matrix $\mathcal{D}$ ($\mathbb{N}$ users, $\mathbb{M}$ items),
- Representative categories for user u ($User\_categories^u$),
- List of items per category ($Item\_categories$)

Output: $Personalized\_items\_for\_addition^u$

     // 0. $Initial\_Predictions^u$: Generate predictions using UserKNN.

for *(user $u \in \mathbb{N}$)* **do**
    for *(item $i \in \mathbb{M}$)* **do**
         Similarity computation finds nearest neighbor candidates;
         Sort predicted items based on the number of possessed neighbor candidates;
         $Initial\_Predictions^u$ contains the top-k predicted items for user u

     // 1. $Personalized\_items\_for\_addition^u$: Create a personalized list of items that can be added to the profile of user u.

for *(user $u \in \mathbb{N}$)* **do**
    for item $i \in Initial\_Predictions^u$ **do**
        if *(category of item $i \notin User\_categories^u$)* **then**
             $Personalized\_items\_for\_addition^u = Personalized\_items\_for\_addition^u$.add($i$)

Personalized_items_for_addition For each user u, we create a list of items that can be added to their profile. This involves selecting items that are present in $Initial\_Predictions^u$ and that are associated with a category that is not in $User\_categories^u$. In other words, we select items that are probably relevant to the user and that are from a category with which they have not had much interaction. As can be seen in Algorithm 1 (second part), items in this list are ranked based on the ranking in $Initial\_Predictions^u$.

3.2   One-step and Two-step pre-processing

Algorithm 2 presents the basic skeleton of our one and two-step pre-processing. For each user, we calculate the number of items to be added based on their initial profile size and parameter λ, which represents the desired addition level as a percentage of the initial profile. We then iteratively pick items from the personalized list $Personalized\_items\_for\_addition^u$ and add them to the user's profile. The two-step pre-processing takes the data generated by one-step pre-processing as input and focuses on removing interactions from a user's profile. The removal process helps in

---

Algorithm 2: One and two-step pre-processing algorithm.

Input:
- λ: level of pre-processing
- User-item matrix $\mathcal{D}$ ($\mathbb{N}$ users, $\mathbb{M}$ items)
- $\mathcal{D}'$ ($\mathbb{N}$ users, $\mathbb{M}$ items) is a copy of $\mathcal{D}$ at time $t = 0$
- $\mathcal{D}''$ ($\mathbb{N}$ users, $\mathbb{M}$ items) is a copy of $\mathcal{D}'$ after the one-step pre-processing.
- Initial count: user profile size at time $t = 0$
- Interaction count: user profile size after adding $\lambda\%$ extra interactions
- Removal threshold
- Total added $= 0$ at time $t = 0$

Output: One-step and two-step pre-processed data $\mathcal{D}'$ and $\mathcal{D}''$ ($\mathbb{N}$ users, $\mathbb{M}$ items)

```
// 1. One-step: adding extra items
```
for *(user u in $\mathbb{N}$)* **do**
  count = initial count [u] $*\lambda$
  added = 0
  while *added $<$ count* **do**
    i = pick the item in the highest position in $Personalized\_items\_for\_addition^u$
    if $\mathcal{D}'[u, i] == 0$ **then**
      $\mathcal{D}'[u, i] = 1$
      added $+= 1$

Total added $+=$ added

```
// 2. Two-step pre-processing: Removing certain items
```
for *user u in $\mathcal{N}$* **do**
  if *Interaction count $\geq$ Removal threshold* **then**
    removed count $+= 1$

To be removed = Total added/removed count
for *user u in $\mathcal{N}$* **do**
  if *Interaction count $\geq$ Removal threshold* **then**
    removed = 0
    while *removed $<$ To be removed* **do**
      i = Randomly pick an item that has not recently been added
      if $\mathcal{D}'[u, i] \neq 0$ **then**
        $\mathcal{D}''[u, i] = 0$
        removed $+= 1$

---

maintaining the density of the modified data close to that of the original data. The removal is performed in such a way that the total number of user interactions for each item remains proximate to the total in the original data set. We distribute the removal of interactions evenly across all users. To account for users with very short profiles, we set a threshold of 20 that exempts them from having interactions removed.

The removal step is driven by several motivations. First, it respects the nature of recommender system data, which is typically sparse and follows a long-tail distribution. Maintaining these properties, we avoid unintended shifts that might impact our results [7]. Second, it addresses the potential impact of the increased data density from the addition step. It has been shown in [11] that the performance of recommendation algorithms is not necessarily and solely impacted by the profile size of the users. When we adjust for the added interactions, we aim to ensure that observed changes in performance are due to our pre-processing approaches rather than shifts in the data's size. Finally, we apply the removal selectively to long profiles, ensuring minimal disruption to users with shorter profiles, thus maintaining a fair representation across the data.

4 Experimental Setup

In this section, we start by providing an overview of our data sets and variants. Next, we describe recommender system algorithms used for both news and book recommendations in our experiments.

Data sets. We test our approach on two publicly available data sets: the MIND data [41] for news recommendations and the GoodBook data for book recommendations. Statistics for the data sets can be found in Table 1. From MIND, we use a subset of the published train data and the entire published test data. In the GoodBook data, we transform explicit ratings into implicit, considering ratings with a rating $>= 3$ as implicit indicators of preference. We randomly sample 80% of each user profile for the training set and we keep the remaining 20% for the test set. It is important to note that the choice of static splitting plays a key role in preventing the data pre-processing from adding items to the test set. Our pre-process is solely applied to the training set, with careful consideration to avoid adding interactions already present in the test set. As a result, the test set remains invariant across all experimental conditions. It is worth noting that the added items through pre-processing are not recommended anymore. While these added items could still be of interest, this is not a concern for MIND, as the added items are older and less likely to be relevant. However, for GoodBook, where random splitting is used, this could pose a challenge since the added items may be more recent and potentially impactful for recommendations. We recommend the use of temporal splitting whenever possible to ensure that recent items are recommended. In the MIND data, there are 17 categories, whereas in the GoodBook data, there are 31 genres. A news article is assigned to a single category, while a book is associated with multiple genres. Both news categories and book genres are considered as *categories* in our approach as discussed above.

Table 1: Statistics of the data sets used in the experiments.

Data set	Type	#Users	#Items	#Clicks	Sparsity (%)	#Categories
MIND News	Training	1,000	26,740	9,368	99.58	17
	Testing	5,000	18,723	15,557	99.82	16
GoodBook	Training	943	729	8,477	98.77	31
	Testing	943	688	4,715	99.27	31

Overview of data variants and experimental parameters. In our experimental setup, we generate several data variants to evaluate the impact of our pre-processing approach. We begin with the original data and apply both one-step and two-step pre-processing. For each approach, we create four variants using different values of the parameter λ. Additionally, we include a baseline where λ interactions are added and removed randomly.[2] Lastly, to ensure that the relevant items are adequately included within the two data sets, we set the maximum cutoff value k differently when we use userKNN to create the lists of *Initial_predictions*; k is set to 100 neighbors for the GoodBook data, and to 50 neighbors for the MIND data.

Recommender system algorithms. We select several recommendation algorithms, ranging from standard collaborative filtering to advanced neural-based models. This

---

[2] Further details about our pre-processed data and results of our baseline are available in the supplementary material due to space constraints `https://surfdrive.surf.nl/files/index.php/s/QaUVGB4Vh3sv7kB`

demonstrates the adaptability of our approach across various recommendation architectures. It also allows us to test our approach with the different algorithms that are currently most prevalent in the two domains that we work with, news and books.

For news recommendations, we use three state-of-the-art neural network recommendation algorithms: *NRMS* uses multi-head self-attention networks [39]. *NPA* uses personalized attention networks to learn news and user representations [38]. *LSTUR* is a news recommender focusing on users' long and short-term preferences [1]. To account for the randomness in the algorithms, we repeat each experiment 5 times and show average performance. We use the recommenders package from Microsoft.[3] For hyperparameter tuning, we set epochs to 50. We follow the parameters as suggested by the recommenders package. For book recommendations, we use CF algorithms: *MostPop* is a non-personalized algorithm recommending the most popular items. *ItemKNN* is the item-based K-nearest neighbor algorithm [8]. *ImplicitMF* is matrix factorization trained with alternating least squares [15]. *BPR* is a matrix factorization algorithm using Bayesian personalized ranking [28]. We use the Lenskit toolkit for our implementation of these algorithms.[4] We perform hyperparameter tuning on the training set to optimize the performance. We use the AllItems strategy for our candidate item selection, as proposed in [2]. We measure Normalized Discounted Cumulative Gain (nDCG) and Mean Reciprocal Rank (MRR). For the MIND data, we report MRR for the entire recommendation list as implemented in Recommenders, whereas, for the GoodBook data, we measure MRR@10 as is common with Lenskit.

Diversity and fairness metrics. We apply a normative diversity metric, calibration, which assesses how closely recommendations align with user preferences inferred from their past interactions. We quantify calibration as the difference between the distributions of categories in user preferences and recommendations, measured with the Kullback-Leibler (KL) and Jensen-Shannon (JS) divergence metrics, as described in [36]. We also compute two descriptive diversity metrics: item coverage (coverage) and the Gini index (gini). Item coverage represents the proportion of unique items recommended across users relative to the entire catalog, while the Gini index captures inequality in item distribution among users. To facilitate interpretation, we report $1 -$ Gini. For all three diversity metrics, higher values indicate greater diversity.

Finally, we apply discounted cumulative fairness, or fair-nDCG [25]. Fair-nDCG is an adaptation of nDCG that incorporates fairness by emphasizing exposure to underrepresented categories. It accumulates the gain for items belonging to the protected group while applying a logarithmic discount based on their rank, emphasizing their presence at higher ranks to ensure fair representation in the recommendation list. In our experiments, we designate the least-represented categories in the data set as the protected group. For MIND, these categories are: kids, weather, video, music, autos, movies, middleeast, and northamerica. For GoodBook, they include: music, poetry, horror, spirituality, sports, christian, comics, manga, cookbooks, psychology, and art.

5   Recommendation Performance

In Table 2 and Table 3, we present our recommendation results using one-step and two-step pre-processing techniques on the MIND and GoodBook data sets, respectively. We measure the accuracy of the recommendation using MRR, nDCG@5, and nDCG@10.

---

[3] https://github.com/recommenders-team/recommenders
[4] https://github.com/lenskit/lkpy/tree/main

Table 2: Recommendation performance for the MIND data. Recommendations with the highest scores are marked in bold, comparing original and pre-processed data. Statistical significance is indicated by a *, based on paired t-tests comparing λ values to the baseline ($\lambda = 0$). Standard deviation is denoted by $\pm$.

		One-Step Pre-process						Two-Step Pre-process					
		Accuracy			Calibration @10		Descriptive @10	Accuracy			Calibration @10		Descriptive @10
Algorithms	λ (%)	MRR	nDCG@5	nDCG@10	KL	JS	Cov	MRR	nDCG@5	nDCG@10	KL	JS	Cov
NRMS	0	0.2688 ± 0.001	0.2914 ± 0.002	0.3592 ± 0.001	2.1947	0.3918	1342	0.2688 ± 0.001	0.2914 ± 0.002	0.3592 ± 0.001	2.1947	0.3918	1342
	1	0.2686 ± 0.002	**0.2920** ± 0.002	**0.3596** ± 0.002	2.2307*	0.3949*	1548*	0.2664* ± 0.002	0.2876* ± 0.001	0.3564* ± 0.002	2.2427*	0.3959*	1519*
	2	0.2694 ± 0.002	0.2918 ± 0.002	0.3594 ± 0.002	2.2490*	0.3960*	1557*	0.2700* ± 0.001	0.2924 ± 0.002	0.3594 ± 0.000	2.2672*	0.3978*	1502*
	5	0.2702* ± 0.001	0.2934* ± 0.001	0.3610 ± 0.001	2.2063	0.3939*	1509*	0.2622* ± 0.001	0.2826* ± 0.001	0.3506 ± 0.001	2.2595*	0.3985*	1492*
	10	0.2686 ± 0.002	0.2916 ± 0.003	0.3586 ± 0.002	2.2579*	0.3972*	1554*	0.2636* ± 0.002	0.2864* ± 0.003	0.3532 ± 0.003	2.2221*	0.3950*	1521*
NPA	0	**0.2740** ± 0.002	0.2968 ± 0.003	0.3606 ± 0.002	2.1994	**0.3906**	1314	0.2740 ± 0.002	0.2968 ± 0.003	0.3606 ± 0.002	2.2103	0.3904	1328
	1	**0.2758** ± 0.003	**0.2984** ± 0.004	**0.3610** ± 0.003	2.1975	0.3898	**1323**	0.2759 ± 0.003	0.2978 ± 0.003	0.3618 ± 0.003	2.1938	**0.3906**	1274
	2	0.2754 ± 0.002	0.2988 ± 0.003	0.3624 ± 0.003	2.1702	0.3894	1269	0.2740 ± 0.002	**0.2996** ± 0.003	0.3612 ± 0.003	2.2675	0.3956	**1347**
	5	0.2764* ± 0.002	0.300* ± 0.003	0.3634* ± 0.002	2.1805	0.3906	1282	0.2774 ± 0.003	0.3034* ± 0.002	0.3662* ± 0.003	2.1511	0.3886	1316
	10	**0.2776** ± 0.003	**0.3014** ± 0.004	**0.3650** ± 0.004	2.1878	0.3904	1304	0.2768 ± 0.004	0.3008 ± 0.004	0.3634 ± 0.004	2.2186	0.3929	1292
LSTUR	0	0.2968 ± 0.005	0.3248 ± 0.005	0.3664 ± 0.004	2.3404	0.3979	1480	0.2968 ± 0.005	0.3248 ± 0.005	0.3664 ± 0.004	2.2243	0.3919	1480
	1	0.2952 ± 0.003	0.3230 ± 0.004	**0.3852** ± 0.003	2.3820	0.3995	1491*	0.2900* ± 0.001	0.3158* ± 0.002	0.3800 ± 0.001	2.3839*	0.3996	1523*
	2	0.2964 ± 0.005	0.3244 ± 0.006	**0.3872** ± 0.004	2.3419*	0.3978	1455*	0.2862* ± 0.004	0.3128* ± 0.004	0.3764 ± 0.003	2.3406	0.3969	1504*
	5	**0.2968** ± 0.002	**0.3248** ± 0.003	**0.3872** ± 0.003	2.3615	0.3991	1452*	0.2974* ± 0.002	0.3126* ± 0.002	0.3754 ± 0.001	2.2614*	0.3935*	1470*
	10	0.2962 ± 0.005	0.3236 ± 0.006	**0.3864** ± 0.005	2.3407	0.3991	1442*	0.2884 ± 0.005	0.3138 ± 0.005	0.3766 ± 0.004	2.2915*	0.3944	1371*

On the MIND data, NRMS experiences improvement at $\lambda = 2\%$ and 5% for the one-step variant, and only at 2% for the two-step variant. NPA exhibits improved performance across λ values for both variants. LSTUR consistantly improves on nDCG@10, but decreases for $\lambda = 1\%, 2\%, 10\%$ on MRR and nDCG@5. On the GoodBook data, we observe that all the algorithms outperform MostPop. The recommendation performance of ItemKNN and ImplicitMF varies across different levels of λ, while BPR consistently maintains higher accuracy across λ levels, particularly at 1% and 2%. In book recommendation, performance seems to decrease at the highest λ values. This phenomenon may be attributed to the CF algorithms' sensitivity to popularity bias [7,6]. A higher λ means a larger change in the long-tail distribution of item exposure (in the train set, not in the test set), which negatively impacts the accuracy.

In general, the observed differences in accuracy are small (maximum increase is 0.002 for GoodBook on BPR; maximum decrease is 0.034 for the same data and algorithm), signifying that performance remains relatively stable when we apply our approach. Many of the observed differences are not statistically significant. In the one-step variant, only some observed increases in accuracy are significant. In the two-step approach, we observe both significant increases and decreases in performance. This is in line with our expectations, that a larger profile (as in the one-step variant) leads to improved performance. We see that algorithms react differently to different λ values. This emphasizes the need to carefully select λ to maintain recommendation quality.

6   Recommendation Diversity

Table 2 and Table 3 present our diversity analysis for news and book recommendations, respectively. We report calibration as a normative diversity metric and coverage as well as the Gini index as descriptive diversity metrics. Focusing on news rec-

Table 3: Recommendation performance for the GoodBook Data. We emphasize the recommendations with the highest scores across conditions (original vs pre-processed data). Statistical significance is indicated by *, based on paired t-tests comparing λ values to the baseline ($\lambda = 0$). Standard deviation is denoted by $\pm$.

| | | One-step Pre-process | | | | | | | Two-step Pre-process | | | | | | |
| | | Accuracy | | | Calibration@10 | | Descriptive@10 | | Accuracy | | | Calibration@10 | | Descriptive@10 | |
Algorithms	lambda (%)	MRR	nDCG@5	nDCG@10	KL	JS	Cov	Gini	MRR	nDCG@5	nDCG@10	KL	JS	Cov	Gini
MostPop	0	0.0739 ± 0.00	0.0347 ± 0.00	0.0507 ± 0.00	0.0626	0.1784	31	0.3493	0.0739 ± 0.00	0.0347 ± 0.00	0.0507 ± 0.00	0.0626	0.1784	31	0.3493
	1	0.0739* ± 0.00	0.0347* ± 0.00	0.0507* ± 0.00	0.1985*	0.3091*	31	0.3493	0.0730* ± 0.00	0.0317* ± 0.00	0.0461* ± 0.00	0.1988*	0.3090*	29	0.3707
	2	0.0742* ± 0.00	0.0348* ± 0.00	0.0510* ± 0.00	0.1994	0.3095	31	0.3494	0.0686* ± 0.00	0.0272* ± 0.00	0.0461* ± 0.00	0.1999	0.3094	29	0.3699
	5	0.0494* ± 0.00	0.0389* ± 0.00	0.0348* ± 0.00	0.2005	0.3102	31	0.3499	0.0438* ± 0.00	0.0172* ± 0.00	0.0298* ± 0.00	0.1809	0.2976	25	0.4246
	10	0.0739* ± 0.00	0.0317* ± 0.00	0.0476* ± 0.00	0.1968	0.3029	31	0.3501	0.0758* ± 0.00	0.0277* ± 0.00	0.0415* ± 0.00	0.1766	0.2883	20	0.5227
ItemKNN	0	0.3881 ± 0.01	0.1821 ± 0.007	0.2248 ± 0.005	0.0819	0.1944	683	0.3471	0.3881 ± 0.01	0.1821 ± 0.007	0.2248 ± 0.005	0.0819	0.1944	683	0.3471
	1	0.3793 ± 0.01	0.1791 ± 0.002	0.2203 ± 0.008	0.1711	0.2899	665	0.3503	0.3661 ± 0.002	0.1756 ± 0.003	0.2134* ± 0.002	0.1711	0.2899	665	0.3503
	2	0.3894 ± 0.008	0.1824 ± 0.006	0.2209 ± 0.004	0.1712	0.2892	663	0.3345	0.3899* ± 0.006	0.1713 ± 0.002	0.2294* ± 0.002	0.1719	0.3094	663	0.3384
	5	0.3566 ± 0.004	0.1725* ± 0.005	0.2061* ± 0.002	0.1741	0.2906	662	0.2886	0.3258* ± 0.010	0.1576* ± 0.004	0.1884* ± 0.004	0.1747	0.2916	665	0.3503
	10	0.3582* ± 0.007	0.1714 ± 0.002	0.2048* ± 0.002	0.1746	0.2907	602	0.2249	0.3378* ± 0.007	0.1576* ± 0.006	0.1949* ± 0.001	0.1729	0.2905	663	0.2961
ImplicitMF	0	0.3706 ± 0.003	0.1793 ± 0.003	0.2192 ± 0.002	0.0853	0.1998	608	0.4629	0.3706 ± 0.003	0.1793 ± 0.003	0.2192 ± 0.002	0.0853	0.1998	608	0.4629
	1	0.3719* ± 0.003	0.1804 ± 0.005	0.2213* ± 0.002	0.1693	0.2890	607	0.4661	0.3533* ± 0.004	0.1656* ± 0.001	0.2100* ± 0.001	0.1701	0.2892	603	0.4695
	2	0.3659* ± 0.007	0.1755 ± 0.002	0.2152 ± 0.003	0.1716	0.2894	605	0.4666	0.3531* ± 0.013	0.1648* ± 0.00	0.2027 ± 0.005	0.1720	0.2911	605	0.4679
	5	0.3481* ± 0.001	0.1656* ± 0.004	0.2035* ± 0.002	0.1732	0.2911	630	0.4533	0.3265* ± 0.00	0.1518* 0.001	0.1892* ± 0.003	0.1724	0.2912	643	0.4554
	10	0.3516* ± 0.006	0.1635* ± 0.004	0.2042* ±0.001	0.1724	0.2895	649	0.4654	0.3346* ± 0.004	0.1546* ± 0.005	0.1931* ± 0.00	0.1696	0.2875	657	0.4889
BPR	0	0.3469 ± 0.00	0.1619 ± 0.00	0.1957 ± 0.00	0.0837	0.1979	503	0.4463	0.3469 ± 0.00	0.1619 ± 0.00	0.1957 ± 0.00	0.0837	0.1979	503	0.4463
	1	0.3470* ± 0.00	0.1624* ± 0.00	0.1956* ± 0.00	0.1686*	0.2885*	495	0.4579	0.3472* ± 0.00	0.1614* ± 0.00	0.1993* ± 0.00	0.1700*	0.2893*	499	0.4481
	2	0.3509* ± 0.00	0.1607* ± 0.00	0.1970* ± 0.00	0.1683*	0.2883*	495	0.4489	0.3529* ± 0.00	0.1630* ± 0.00	0.2004 ± 0.00	0.1656*	0.2884*	509	0.4353
	5	0.2910* ± 0.00	0.1331* ± 0.00	0.1604* ± 0.00	0.1666*	0.2868*	538	0.4358	0.2989* ± 0.00	0.1331* ± 0.00	0.1647* ± 0.00	0.1689*	0.2884*	569	0.4264
	10	0.3034* ± 0.00	0.1405* ± 0.00	0.1690* ± 0.00	0.1665*	0.2876*	576	0.3987	0.3169* ± 0.00	0.1404* ± 0.00	0.1744* ± 0.00	0.1657*	0.2859*	615	0.4048

ommendations, NRMS consistently yields higher calibration and coverage scores for $\lambda = 1\%, 2\%, 5\%, 10\%$ for both 1-step and 2-step variants, as does LSTUR. However, for NPA, we observe mixed results, with diversity going up or down depending on λ. For book recommendations, we see positive reactions to diversity from recommender algorithms with varying λ values. All four algorithms show more calibrated results on preprocessed data compared to the original data, across all λ values.

However, this trend does not hold to the same extent for the descriptive metrics. While ImplicitMF and BPR exhibit optimal coverage results for $\lambda = 5\%, 10\%$, ItemKNN performs best at $\lambda = 1\%$. We observe discrepancies between the different diversity metrics, where an increase in calibration metrics does not necessarily translate to an increase in descriptive metrics.

7 Provider Fairness

As mentioned in Section 1, while diversity and provider fairness are based on distinct normative principles, increasing diversity can also contribute to greater provider fairness [9]. Figure 2 illustrates fair-nDCG results for MIND and GoodBook Data, respectively. Each pair of figures corresponds to a specific algorithm, with one figure depicting

the results using one-step pre-processing and the other for two-step pre-processing. The fair-nDCG scores are evaluated across different levels of λ for top-k recommendations ranging from 1 to 100. We see that recommendations generated using pre-processed data consistently exhibit higher fair-nDCG scores, indicating increased exposure fairness. Also, we observe that the 2-step pre-processing is more effective at lower levels of λ. For instance, in Figure 2a (NRMS), recommendations using 10% one-step variant (and 5% for two-step) achieve the highest fair-nDCG scores, closely followed by the 1% and 2% pre-processing levels. This emphasizes that pre-processing not only improves the accuracy of the recommendations but also contributes to increased exposure to minority categories. For the LSTUR algorithm, we note that at top-k = 20 with 1-step pre-processing, recommendations using the original data ($\lambda = 0$) achieve the best fair-nDCG score. However, with 2-step pre-processing, removing some interactions helps to produce more fair recommendations by increasing the exposure of minority categories ($\lambda = 1\%, 10\%$). Similar observations hold for GoodBook Data in Figure 2b. In the case of BPR-Figure 2b, recommendations with 1% and 5% One-step variants exhibit the highest fair-nDCG-scores. Additionally, recommendations with 2% and 5% two-step pre-processing demonstrate noteworthy exposure.
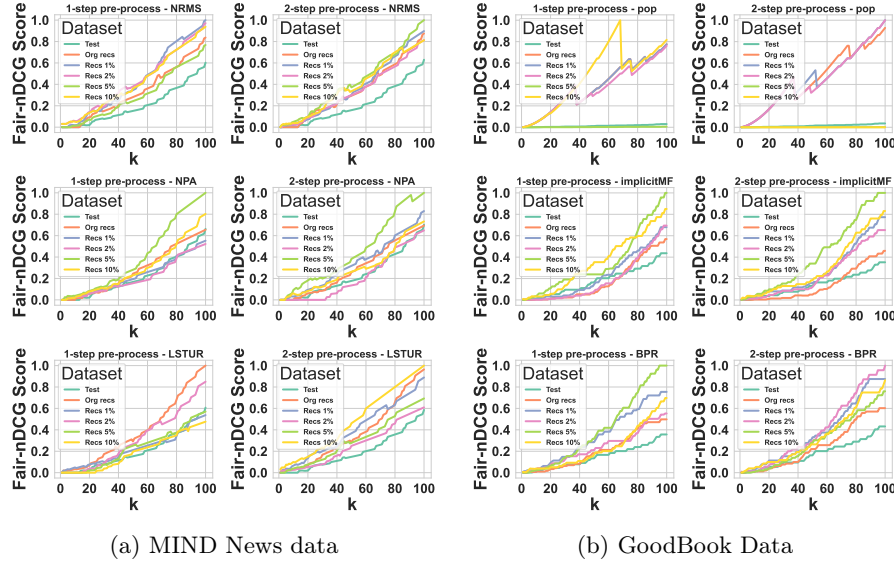

(a) MIND News data   (b) GoodBook Data

Fig. 2: Fair-nDCG measured for different top-k recommendation lists. The fair-nDCG-score is measured for the recommendation of the minority *categories*.

8   Discussion and Conclusion

In this paper, we introduced a simple yet effective user-centered pre-processing approach to diversify recommender system outputs. The central components of our approach include: (1) using a personalized list of items to maintain accuracy, (2) a well-

established selection of categories within user profiles to enhance diversity, and (3) adding and removing λ % of interactions from a user profile.

Main findings Our results demonstrated that keeping data pre-processing closely aligned with user preferences, has the potential to maintain or even improve the performance of the original profiles. As for *fairness and diversity*, our experiments have also shown that our pre-processing approach achieves diverse recommendations and simultaneously promotes provider fairness. For diversity, through extensive analysis utilizing normative and descriptive diversity measures, we have demonstrated that our approach has the potential to increase item coverage and Gini index and amplify divergence. We found that there is a discrepancy between the results of calibration metrics and descriptive metrics for diversity. This misalignment could be explained by the fact that normative metrics focus on ideal distributions whereas descriptive metrics capture observed variations in user preferences. As a consequence, relying solely on one type of metric could lead to either overestimating or underestimating the true diversity of recommendations. For fairness, we have demonstrated that our approach actively promotes the recommendation of underrepresented categories. This success is driven by (1) using personalized lists to preserve accuracy, (2) strategically selecting user categories to enhance diversity, and (3) adjusting λ % of interactions.

Future work Our paper opens an important new vista for future work. One direction could be on *adapting to changes in user preferences and catalogs*. This implies that recommender systems must adaptively and periodically update their data to reflect evolving user preferences and item catalogs. Our user-centered pre-processing approach complements the original data by adding extra interactions without altering the core model. This dynamic adjustment provides a practical solution to address biases in training data, promoting consistent gains in both fairness and diversity.

Also, another direction could be related to *ethical considerations*. While our approach aims to improve diversity, it is essential to acknowledge the ethical implications of modifying user profiles [12]. One key consideration is user autonomy. Users have the right to control their data and how it is used. Introducing modifications without explicit consent could undermine this autonomy and lead to ethical concerns. It is crucial to incorporate mechanisms that allow users to opt in or out of such modifications, giving them control over their data and the recommendations they receive. Additionally, the explainable nature of our approach opens up new research lines around transparency for end users. Future work could investigate to what extent users benefit from knowing how their profile has been altered, or, one step further, how they could benefit from functionality to modify their profiles themselves. Another aspect to consider is user autonomy. Users have the right to control their data and how it is used in recommendation systems. Introducing modifications to user profiles without their explicit consent could undermine user autonomy and potentially lead to ethical dilemmas. It is crucial to incorporate mechanisms that allow users to opt in or opt out of such modifications, giving them greater control over their data and the recommendations they receive.

Acknowledgment

References

1. An, M., Wu, F., Wu, C., Zhang, K., Liu, Z., Xie, X.: Neural news recommendation with long- and short-term user representations. In: Korhonen, A., Traum, D., Màrquez, L. (eds.) Proceedings of the Annual Meeting of the Association for Computational Linguistics. pp. 336–345 (2019)
2. Bellogin, A., Castells, P., Cantador, I.: Precision-oriented evaluation of recommender systems: An algorithmic comparison. In: Proceedings of the Fifth ACM Conference on Recommender Systems. p. 333–336 (2011)
3. Boratto, L., Fabbri, F., Fenu, G., Marras, M., Medda, G.: Fair augmentation for graph collaborative filtering. In: Proceedings of the 18th ACM Conference on Recommender Systems. p. 158–168 (2024)
4. Boratto, L., Fenu, G., Marras, M.: Interplay between upsampling and regularization for provider fairness in recommender systems. User Modeling and User-Adapted Interaction **31**(3), 421–455 (2021)
5. Daniel, F., Kartik, H.: Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. Management Science **55**(5) (2009)
6. Daniil, S., Cuper, M., Liem, C.C.S., van Ossenbruggen, J., Hollink, L.: Reproducing popularity bias in recommendation: the effect of evaluation strategies. ACM Transaction on Recommender Systems (2023)
7. Deldjoo, Y., Bellogin, A., Di Noia, T.: Explaining recommender systems fairness and accuracy through the lens of data characteristics. Information Processing & Management **58**(5), 102662 (2021)
8. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Transaction on Information Systems **22**(1), 143–177 (2004)
9. Ekstrand, M.D., Das, A., Burke, R., Diaz, F.: Fairness in Recommender Systems, pp. 679–707. Springer US (2022)
10. Ekstrand, M.D., Joshaghani, R., Mehrpouyan, H.: Privacy for all: Ensuring fair and equitable privacy protections. In: Friedler, S.A., Wilson, C. (eds.) Proceedings of the International Conference on Fairness, Accountability and Transparency. Proceedings of Machine Learning Research, vol. 81, pp. 35–47 (2018)
11. Ekstrand, M.D., Tian, M., Azpiazu, I.M., Ekstrand, J.D., Anuyah, O., McNeill, D., Pera, M.S.: All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. In: Friedler, S.A., Wilson, C. (eds.) Proceedings of the International Conference on Fairness, Accountability and Transparency. Proceedings of Machine Learning Research, vol. 81, pp. 172–186. PMLR (2018)
12. Grisse, K.: Recommender Systems, Manipulation and Private Autonomy: How European Civil Law Regulates and Should Regulate Recommender Systems for the Benefit of Private Autonomy, pp. 101–128. Springer International Publishing (2023)
13. Hansen, C., Mehrotra, R., Hansen, C., Brost, B., Maystre, L., Lalmas, M.: Shifting consumption towards diverse content on music streaming platforms. In: Proceedings of the ACM International Conference on Web Search and Data Mining. pp. 238–246 (2021)
14. Helberger, N., Karppinen, K., D'Acunto, L.: Exposure diversity as a design principle for recommender systems. Information, Communication & Society **21**(2), 191–207 (2018)
15. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: IEEE International Conference on Data Mining. pp. 263–272 (2008)

16. Kaminskas, M., Bridge, D.: Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. ACM Transactions on Interactive Intelligent Systems **7**(1) (2016)

17. Lakshminarayan, K., Harp, S.A., Samad, T.: Imputation of missing data in industrial databases. Applied intelligence **11**(3), 259–275 (1999)

18. Liu, Q., Reiner, A.H., Frigessi, A., Scheel, I.: Diverse personalized recommendations with uncertainty from implicit preference data with the bayesian mallows model. Knowledge-Based Systems **186**, 104960 (2019)

19. Liu, Y., Li, M., Ariannezhad, M., Mansoury, M., Aliannejadi, M., de Rijke, M.: Measuring item fairness in next basket recommendation: A reproducibility study. In: Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part IV. p. 210–225. Springer-Verlag (2024)

20. Mansoury, M., Abdollahpouri, H., Pechenizkiy, M., Mobasher, B., Burke, R.: Fairmatch: A graph-based approach for improving aggregate diversity in recommender systems. In: Proceedings of the ACM International Conference on User Modeling, Adaptation and Personalization. pp. 154–162 (2020)

21. Meng, X., Huo, H., Zhang, X., Wang, W., Zhu, J.: A survey of personalized news recommendation. Data Science and Engineering **8**(4), 396–416 (2023)

22. Michiels, L., Vannieuwenhuyze, J., Leysen, J., Verachtert, R., Smets, A., Goethals, B.: How should we measure filter bubbles? a regression model and evidence for online news. In: Proceedings of the 17th ACM Conference on Recommender Systems. p. 640–651 (2023)

23. Oliveira, R.S., Nóbrega, C., Marinho, L.B., Andrade, N.: A multiobjective music recommendation approach for aspect-based diversification. In: Proceedings of the International Society for Music Information Retrieval Conference. pp. 414–420 (2017)

24. Pariser, E.: The filter bubble: What the Internet is hiding from you. penguin UK (2011)

25. Pitoura, E., Stefanidis, K., Koutrika, G.: Fairness in rankings and recommendations: an overview. The VLDB Journal **31**(3), 431–458 (2021)

26. Rastegarpanah, B., Gummadi, K.P., Crovella, M.: Fighting fire with fire: Using antidote data to improve polarization and fairness of recommender systems. In: Proceedings of the 12th ACM International Conference on Web Search and Data Mining. p. 231–239 (2019)

27. Raza, S., Bashir, S.R., Naseem, U.: Accuracy meets diversity in a news recommender system. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 3778–3787 (2022)

28. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the International Conference on Uncertainty in Artificial Intelligence. p. 452–461. AUAI Press (2009)

29. Rubinsteyn, A., Feldman, S.: fancyimpute: An imputation library for python, https://github.com/iskandr/fancyimpute

30. Slokom, M., Hanjalic, A., Larson, M.: Towards user-oriented privacy for recommender system data: A personalization-based approach to gender obfuscation for user profiles. Information Processing & Management **58**(6) (2021)

31. Smyth, B., McClave, P.: Similarity vs. diversity. In: Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development. p. 347–361. Springer-Verlag (2001)

32. Stamenkovic, D., Karatzoglou, A., Arapakis, I., Xin, X., Katevas, K.: Choosing the best of both worlds: Diverse and novel recommendations through multi-objective reinforcement learning. In: Proceedings of the 15th ACM International Conference on Web Search and Data Mining. p. 957–965 (2022)

33. Su, X., Khoshgoftaar, T.M., Greiner, R.: Imputed neighborhood based collaborative filtering. In: International Conference on Web Intelligence and Intelligent Agent Technology. vol. 1, pp. 633–639. IEEE (2008)

34. Treuillier, C., Castagnos, S., Dufraisse, E., Brun, A.: Being diverse is not enough: Rethinking diversity evaluation to meet challenges of news recommender systems. In: Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization. p. 222–233 (2022)

35. Vargas, S., Castells, P.: Rank and relevance in novelty and diversity metrics for recommender systems. In: Proceedings of the ACM International Conference on Recommender Systems. pp. 109–116 (2011)

36. Vrijenhoek, S., Bénédict, G., Gutierrez Granada, M., Odijk, D., De Rijke, M.: Radio – rank-aware divergence metrics to measure normative diversity in news recommendations. In: Proceedings of the ACM International Conference on Recommender Systems. p. 208–219 (2022)

37. Weinsberg, U., Bhagat, S., Ioannidis, S., Taft, N.: BlurMe: Inferring and obfuscating user gender based on ratings. In: Proceedings of the ACM International Conference on Recommender Systems. pp. 195–202 (2012)

38. Wu, C., Wu, F., An, M., Huang, J., Huang, Y., Xie, X.: Npa: Neural news recommendation with personalized attention. In: Proceedings of the ACM International Conference on Knowledge Discovery & Data Mining. p. 2576–2584 (2019)

39. Wu, C., Wu, F., Ge, S., Qi, T., Huang, Y., Xie, X.: Neural news recommendation with multi-head self-attention. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the ACM International Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 6389–6394 (2019)

40. Wu, C., Wu, F., Qi, T., Huang, Y.: End-to-end learnable diversity-aware news recommendation. arXiv preprint arXiv:2204.00539 (2022)

41. Wu, F., Qiao, Y., Chen, J.H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., Zhou, M.: MIND: A large-scale dataset for news recommendation. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the Annual Meeting of the Association for Computational Linguistics. pp. 3597–3606. Association for Computational Linguistics (2020)

42. Wu, Y., Cao, J., Xu, G.: Fairness in recommender systems: Evaluation approaches and assurance strategies. ACM Transaction on Knowledge Discovery Data **18**(1) (2023)

43. Zehlike, M., Bonchi, F., Castillo, C., Hajian, S., Megahed, M., Baeza-Yates, R.: FA*IR: A fair top-k ranking algorithm. In: Proceedings of the ACM on Conference on Information and Knowledge Management. p. 1569–1578 (2017)

44. Zehlike, M., Sühr, T., Baeza-Yates, R., Bonchi, F., Castillo, C., Hajian, S.: Fair top-k ranking with multiple protected groups. Information Processing & Management **59**(1) (2022)

45. Zhao, Y., Wang, Y., Liu, Y., Cheng, X., Aggarwal, C., Derr, T.: Fairness and diversity in recommender systems: a survey. arXiv preprint arXiv:2307.04644 (2023)

46. Zheng, Y., Gao, C., Chen, L., Jin, D., Li, Y.: DGCN: Diversified recommendation with graph convolutional networks. In: Proceedings of the Web Conference 2021. p. 401–412. Association for Computing Machinery (2021)

47. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th International Conference on World Wide Web. p. 22–32 (2005)