Information Processing and Management 58 (2021) 102519

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Bias-Aware Hierarchical Clustering for detecting the discriminated groups of users in recommendation systems

Joanna Misztal-Radecka [a,b,*], Bipin Indurkhya [c]

[a] AGH University of Science and Technology, Poland
[b] Ringier Axel Springer Polska, Poland
[c] Jagiellonian University, Poland

ARTICLE INFO

Keywords:
Recommender system
System fairness
Bias detection
Model interpretability
Collaborative filtering

ABSTRACT

One challenge for the modern recommendation systems is the *Tyranny of Majority* — the generated recommendations are often optimized for the mainstream trends so that the minority preference groups remain discriminated. Moreover, most modern recommendation techniques are characterized as black-box systems. Given a lack of understanding of the dataset characteristics and insufficient diversity of represented individuals, such approaches inevitably lead to amplifying hidden data biases and existing disparities. In this research, we address this problem by proposing a novel approach to detecting and describing potentially discriminated user groups for a given recommendation algorithm. We propose a Bias-Aware Hierarchical Clustering algorithm that identifies user clusters based on latent embeddings constructed by a black-box recommender to identify users whose needs are not met by the given recommendation method. Next, a post-hoc explainer model is applied to reveal the most important *descriptive features* that characterize these user segments. Our method is model-agnostic and does not require any a priori information about existing disparities and sensitive attributes. An experimental evaluation on a synthetic dataset and two real-world datasets from different domains shows that, compared with other clustering methods and arbitrarily selected user groups, our method is capable of identifying underperforming segments for different recommendation algorithms, and detect more severe disparities.

1. Introduction

System fairness has been identified as an important aspect for many machine learning applications because an algorithmic-driven design and insufficient diversity of training sets lead to *automatic discrimination* and other types of systematic biases resulting from stereotypical datasets. In particular, data collected from the web is prone to different types of biases related to user demography, nationality, and activity (Baeza-Yates, 2018; Bolukbasi, Chang, Zou, Saligrama, & Kalai, 2016; Caliskan, Bryson, & Narayanan, 2017; Graells-Garrido, Lalmas, & Menczer, 2015). While in most applications, the model fairness is evaluated based on a priori knowledge about the protected groups and sensitive attributes (such as gender or nationality), in real-world systems, the information about such special groups is often not known beforehand, and there may exist additional factors contributing to disparities. As collaborative filtering (CF) (Zhang & Chen, 2020) methods tend to extract *latent* patterns in user–item interactions, understanding the basis of these patterns is crucial for preventing hidden algorithmic biases and disparities that would otherwise be amplified by the decision-making logic. Moreover, the recommendations are often evaluated based on the global average of some performance metric, which results in optimization for the mainstream trends while the minority preference groups remain discriminated.

* Corresponding author.
E-mail addresses: misztalradecka@agh.edu.pl (J. Misztal-Radecka), bipin.indurkhya@uj.edu.pl (B. Indurkhya).

https://doi.org/10.1016/j.ipm.2021.102519
Received 27 August 2020; Received in revised form 30 December 2020; Accepted 14 January 2021
Available online 2 February 2021

Research objectives and contributions

The main goal of this research is to address the problem of *discrimination discovery* (Luong, Ruggieri, & Turini, 2011) by detecting potentially discriminated groups of similar users for a given recommendation algorithm. Towards this objective, we make the following contributions:

- An unsupervised **Bias-Aware Hierarchical Clustering** method is proposed to detect discriminated segments. This method is applicable to different latent user representations and evaluation metrics without needing any information about sensitive attributes.
- A **post-hoc explanation** technique is applied to provide human-understandable explanations for each of the discriminated segments. These explanations help in identifying potential algorithmic biases.

Our approach is based on well-established and efficient machine learning techniques, so it is applicable for many practical recommendation applications. The novelty of our approach lies in applying these methods to solve the problem of automatically detecting the underlying discrimination in a recommendation algorithm.

Information about the discriminated groups of users may be incorporated into the model-selection process so that models that represent less significant disparities could be selected. However, as there may be a variety of reasons for the discrimination, human-interpretable descriptions of discriminated segments generated in our approach may be used to take compensatory actions, such as increasing the dataset diversity.

This paper is structured as follows: in the rest of this section, we present an overview of the existing research on recommendation fairness and discrimination detection, and we compare our approach with other current state-of-the-art techniques. In Section 2, we define the problem formally and describe our proposed method. The experimental setting for two real-world and one synthetic dataset is described in Section 3. The results are presented in Section 4, which show that our method can detect more significant segment discrimination compared to other unsupervised techniques and pre-defined user attributes. We also evaluate the segment prediction accuracy, and perform a qualitative analysis of the most impactful features. Finally, we present the conclusions and indicate future research directions in Section 5.

Related work

In this section, we present an overview of the related work concerning recommendation fairness, types of biases, user segmentation approaches, and generating explanations in recommendation algorithms.

1.1. Algorithmic fairness in recommendations

Existing recommendation systems suffer from different types of biases that stem from the cultural and demography stereotypes in the training data and the disparities in user activities (Baeza-Yates, 2018; Caliskan et al., 2017; Eskandanian, Sonboli, & Mobasher, 2019; Graells-Garrido et al., 2015; Tsintzou, Pitoura, & Tsaparas, 2018). For instance, Tsintzou et al. (2018) notes that a popular KNN recommendation algorithm tends to amplify the gender bias related to movie genres. The language bias is another significant phenomenon on the Internet — as observed by Baeza-Yates (2018), more than half of the web content is in English, whereas only 27% of the web users speak English. More generally, a small number of influential users may have a large impact on the recommendations of other users (Eskandanian et al., 2019). Moreover, the CF methods often fail to find adequate recommendations for the *gray sheep* users, which are users having unusual tastes (Ghazanfar & Prügel-Bennett, 2014; Su & Khoshgoftaar, 2009; Zheng, Agnani, & Singh, 2017).

Accordingly, the problem of algorithmic fairness has recently attracted much research interest. From the legal perspective, discrimination is a situation where groups are treated *less favorably* by an algorithm (Legislation, 2009). Generally speaking, fairness may be defined as the lack of discrimination against individuals or groups, but researchers have distinguished its different aspects. In Barocas, Hardt, and Narayanan (2019), three basic non-discrimination criteria are defined as properties of the joint distribution of the sensitive attribute A, the target variable Y, and the classifier or score R based on the conditional independence between those variables. In our work, we focus on the *independence* criterion: $R \perp A$, which is also referred to as *parity by impact* (Gajane, 2017) or group fairness. Four distinct fairness objectives are defined by Yao and Huang (2017) for measuring the disparities in rating prediction errors between the advantaged and disadvantaged users. Similarly, in Sánchez and Bellogín (2019), the evaluation metric for recommendations is aggregated according to the groups defined by the user attributes to detect if an algorithm makes more relevant recommendations for users belonging to some specific groups. The *group fairness for subjects* has also been studied in the context of group and top-N recommenders (Sacharidis, 2019; Serbos, Qi, Mamoulis, Pitoura, & Tsaparas, 2017; Xiao, Min, et al., 2017) where the goal is to recommend items to a group of users. However, in most definitions, fairness is considered with respect to the protected groups of users based on pre-defined attributes A (such as demography or region), while we use unsupervised learning techniques to identify such groups automatically. In this respect, our work is similar to Nasiriani, Squicciarini, Saldanha, Goel, and Zannone (2019) where a hierarchical clustering approach is proposed for discrimination discovery. However, the clusters are assigned homogeneity scores based on their majority label; therefore, this approach is not applicable to non-binary problems such as user–item recommendation, and, in contrast to our approach, the clustering is performed on pre-defined user attributes. In our work, we focus on a model-agnostic definition of fairness, which is applicable to both individualized and group recommenders

based on the performance distribution for each user, and we do not require information about protected groups of users to be given beforehand, but detect such groups automatically.

The most straightforward approach to select users discriminated by a given recommendation algorithm A is to order them by the metric value and select those with the lowest results. However, this method would not reveal any particular reason for the disparity since there may be multiple sources of bias. For instance, such a group could contain a mixture of *gray sheep* users and those who are discriminated due to their language but who do not necessarily share any common characteristics. In such a case, it would be difficult to gain any useful insights. In our work, we focus on detecting users who are similar to a group of other users (and have a high probability of having some features in common), and may be considered as *white sheep*, and yet the recommender fails to satisfy their needs. However, we consider both the sensitive groups and the user activity segments in the comparison for the bias detection with the automatically detected clusters.

1.2. User clustering in recommendation systems

As observed by Hennig (2019), there may exist different goals for data clustering. Accordingly, different clustering approaches may be suitable for a given application. In Chu et al. (2009), clusters of Yahoo users were built based on over a thousand categorical features describing their demographics and behavioral patterns. The unsupervised clustering technique has been identified in Nurma Sari, Nugroho, Ferdiana, and Santosa (2016) as the most flexible method for automatic detection of underlying behavioral patterns. In Das, Datar, Garg, and Rajaram (2007), a large-scale collaborative filtering recommender system for Google News personalization was built by applying several clustering techniques, and its efficacy and scalability were demonstrated with a real-world experiment on millions of users. As shown in Steinbach, Karypis, and Kumar (2000), the bisecting k-Means algorithm generally outperforms other clustering techniques in terms of clusters quality and run time, while it tends to produce segments of relatively uniform size. In Misztal-Radecka, Rusiecki, Żmuda, and Bujak (2019) and Sarwar, Karypis, Konstan, and Reidl (2002), the bisecting k-Means was applied to scale up the user neighborhood formation process for online services. In Ghazanfar and Prügel-Bennett (2014), the similarity threshold is calculated to isolate the *gray sheep users* from the rest of the clusters built with the k-Means method.

1.3. Explaining recommendations

Providing intuitions behind the decisions of black-box models has become an active area of research (Guidotti et al., 2018), especially for recommender systems (Tintarev & Masthoff, 2011; Zhang & Chen, 2020). As there may be a variety of reasons for generating explanations (Tintarev & Masthoff, 2011), we focus here on generating *descriptions* of particular user segments to better understand the underlying data characteristics and hidden biases. This is similar to Misztal-Radecka et al. (2019), where segment descriptions were generated based on the content-based user profiles constructed from topic vectors, and Misztal-Radecka, Indurkhya, and Smywiński-Pohl (2020) where user and item metadata are represented in the same vector space. However, such model-specific approaches are limited to a certain class of models, and may suffer from the interpretability-performance trade-off (Ribeiro, Singh, & Guestrin, 2016). In our research, we take a universal model-agnostic approach to model explanations which may be formally classified as the *Explanation Through Interpretable Model* (Guidotti et al., 2018). From this perspective, our work is similar to Singh and Anand (2018), which applies a tree-based model trained on a set of interpretable labels to provide explanations for a black-box learning-to-rank algorithm on the web search. However, we aim at explaining the particular segment characteristics to reveal potential disparities, whereas Singh and Anand (2018) focuses on evaluating how well the post-hoc model approximates the original ranking. Our approach is inspired by the method proposed in Misztal-Radecka and Indurkhya (2020); however, we adapted this technique to generate descriptions for the discriminated segments of users rather than items. We define an interpretable proxy classifier that predicts the output of an unsupervised user embedding clustering model based on the *descriptive user features*. To explore the impact of particular features on the model's decisions, we use the SHapley Additive exPlanation (SHAP) (Lundberg et al., 2019; Lundberg & Lee, 2017) algorithm, which computes the average of the marginal contributions of each feature value to the model prediction across all permutations. This method provides both the local and the global interpretability — each observation has its own SHAP values that may be combined as a positive or a negative contribution of each feature to the target output.

2. Proposed approach

To define the problem more formally, let us define a set of N users $u_1, \dots, u_N$ and a recommendation algorithm $A(u) = i_1, \dots, i_K$ that returns a list of K recommended items for user u sorted according to their relevance for this user. $\overline{M}(A(u))$ is an average recommendation evaluation metric that measures the effectiveness of recommendations of $A(u)$ for all users $u \in U$. Then, $p_u^A \in \mathbb{R}^D$ and $q_i^A \in \mathbb{R}^D$ are D-dimensional latent representations of user u and item i respectively, built by algorithm A, while $desc_u \in \mathbb{R}^L$ is L-dimensional representation of the user u based on his or her *descriptive features*.

According to the *independence* criterion (Barocas et al., 2019), a fair recommender A should perform equally for all groups of users: $M \perp U_C, U_C \in U$. Then, we use the following definition of user group discrimination:

Definition 1. If there exists a group of similar users $U_B \subset U$ such that the average metric $\overline{M}(A(u_b)), u_b \in U_B$ is significantly lower than the average metric for the rest of the users $\overline{M}(A(u)), u \in U \setminus U_B$ at a given confidence level α, we define the group U_B as *discriminated* by the algorithm A in terms of metric M.

We measure the level of discrimination (**negative bias**) as a difference between the metric for this segment and the other users, given that the difference is significant:

$$bias_B = \overline{M}_{U \setminus U_B} - \overline{M}_{U_B} \tag{1}$$

As each user is assigned to at most one segment, we calculate the statistical significance of the difference with the Welch's t-test for the means of two independent samples. The discriminated clusters for which the difference is significant (at the confidence level $\alpha = 0.95$) are detected. More precisely, we focus here on finding the user segments with the highest negative bias (the lowest metric compared to the others).

The proposed solution consists of two stages (described in the following sections):

1. **Detecting** the groups of users U_B who are *discriminated* by a black-box recommendation algorithm A: the recommendations for these groups are less effective than for the rest of the users (Section 2.1).
2. **Describing** these groups in terms of human-understandable *descriptive features* $desc_u$ (Section 2.2).

2.1. Detecting discriminated groups of users

First, we aim at discovering the groups of users that are discriminated by the recommendation algorithm A in terms of metric M. To this end, we apply the clustering algorithm to latent user representations p_u that are built by algorithm A.

2.1.1. User embeddings preparation

As an input, we use the user embeddings p_u^A constructed by the blackbox recommender A. In the preprocessing stage, we apply the Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) (McInnes, Healy, & Melville, 2018) to perform non-linear manifold aware dimension reduction to boost the performance of density-based clustering, which makes our algorithm more universal in terms of robustness to different types of embedding representations. The `n_neighbors` value for the UMAP algorithm is set to 20 to reduce the impact of local structure and to avoid granular clusters, and the `min_dist` parameter is set to zero as we are interested in the dense mapping of the neighboring points. Additionally, when there are only two dimensions, the resulting UMAP embeddings may be used for visualizing the user representations.

2.1.2. Unsupervised user embedding segmentation

We propose an unsupervised top-down hierarchical clustering method **Bias-Aware Hierarchical K-Means** (BAH-KM) (described in Algorithm 1) for discovering clusters of users for whom the recommendations do not fit well. In each iteration, the cluster with the largest standard deviation of metric M is split into two clusters by the bisecting k-Means method. We use the k-Means++ (Arthur & Vassilvitskii, 2007) cluster initialization scheme, which converges more quickly than a random initialization. If the maximum bias for the new clusters is greater or equal than that for the original cluster, the original cluster is split; otherwise, the original cluster is kept. The clusters division process is continued until the desired number of iterations is achieved or there are no more clusters that could be split in this way. Hence, in contrast to other unsupervised clustering algorithms (such as standard k-Means or Agglomerative clustering), our method does not require selecting a desired number of clusters.

As the goal of our approach is to detect the most discriminated groups of users, the Euclidean similarity measure is applied to split the clusters with k-Means, but the splitting is conditioned by finding more severe biases rather than minimizing the variance of the cluster.

Algorithm 1 Bias-Aware Hierarchical K-Means

```
Start with all user vectors in one cluster k.
for i=1...max iterations do
    Split cluster k into two clusters k_0, k_1 with k-Means algorithm.
    Calculate the bias (from Formula (1)) for each of the new clusters: bias_{k_0}, bias_{k_1}
    if max(bias_{k_0}, bias_{k_1}) >= bias_k then
        Add new clusters to the list of clusters and remove the old one.
    end if
    Select new cluster k as the one with the highest standard deviation σ_M of metric M.
end for
```

2.2. Describing discriminated segments

After identifying the under-performing segments of users formed by the black-box embeddings, we aim at discovering the characteristics of these user groups. To this end, the embedding clustering model $C(u)$ is treated as a black-box, and a post-hoc model is used to predict the cluster to which a given user u belongs based on its *descriptive features* $desc_u$. These features are constructed from different user attributes and item attributes (such as gender, country, activity, the average rating for item categories, tags, or year of production in their browsing history), as described in Table 2. A tree-based Gradient Boosting Trees (GBT) classifier with an efficient distributed implementation `xgboost` (Chen & Guestrin, 2016) is used to provide straightforward and efficient feature interpretability. A binary classifier is trained for each discriminated cluster with positive labels assigned to users belonging to a

Table 1
Synthetic data distribution generation parameters.

Parameter	Values range	Sampling distribution
Segments count	2–10	Uniform
Users in segment count	10–200	Uniform
Feature vector dimensions	2	Constant
Feature value	0–1	Gaussian
Metric value per user	0–1	Gaussian

Table 2
Data summary — item and user metadata from the MovieLens 100K and Book-Crossing recommendation datasets.

	MovieLens	Book-Crossing
Users cnt	943	278 858
Distinct items	1 682	271 379
User metadata	Gender, age, occupation	Age, location
Item metadata	Title, genres, year	Title, author, publisher, year

given segment and negative labels for the rest of users. To retrieve an explanation from the proxy model as the impact of particular *descriptive features* on the classifier output for each cluster, we applied the SHapley Additive exPlanation (SHAP) (Lundberg et al., 2019; Lundberg & Lee, 2017) algorithm described in Section 1.3. We apply the Tree Explainer to calculate the exact Shapley values for the tree-based model (Lundberg et al., 2019). In particular, the most impactful features are visualized with a *summary plot* to show the positive and negative relationships between the features and the target (the segment assignment).

3. Experimental validation

In the experimental evaluation, we aim at answering the following research questions:

1. Are unsupervised clustering methods more effective in detecting groups of discriminated users compared to some arbitrarily selected protected social groups and other pre-defined segments commonly used for evaluating algorithm fairness?
2. Is our proposed BAH-KM algorithm capable of detecting more severe disparities compared to other unsupervised clustering methods?

3.1. Datasets

To address these research questions, we performed experiments on one synthetic dataset (described in Section 3.1.1) and two real-world datasets from different domains (described in Section 3.1.2).

3.1.1. Synthetic dataset

To evaluate how well our proposed algorithm detects discriminated segments of users compared to other unsupervised methods, we generated a synthetic dataset of user features p_u with a biased metric M (see Table 1). We ran the simulation procedure described in Algorithm 2 on ten iterations and compared the average metrics and the statistical significance for the selected unsupervised clustering algorithms. An example of generated data is presented on Fig. 1 — metric distribution (A) and synthetic clusters with 2-dimensional features (B). As the generated data is already 2-dimensional, we did not use UMAP for dimensionality reduction in these experiments on the synthetic dataset.

Algorithm 2 Synthetic data generation.

1: Draw a number of segments $K \sim \mathcal{U}(2, 10)$.
2: **for** segment $k = 1 \dots K$ **do**
3: Draw a number of users in the segment $N_k \sim \mathcal{U}(10, 200)$.
4: Generate user features as normally-distributed clusters of 2D points.
5: Generate a metric for each user $u_i, i = 1 \dots N_k$:
$M_n \sim \mathcal{N}(\mu_{M_k}, \sigma^2_{M_k})\,; \mu_{M_k}, \sigma_{M_k} \sim \mathcal{U}(0, 1)$.
6: **end for**
7: Draw a number of outliers $N_{outliers} \sim \mathcal{U}(10, 200)$ and generate their features from a uniform distribution.

Since the metric M is generated from a different distribution probability $\mathcal{N}(\mu_{M_k}, \sigma_{M_k})$ for each cluster k, some clusters may be discriminated.

3.1.2. Real-world datasets

We used two popular public real-world datasets from different domains — MovieLens 100K movie recommendation dataset (Harper & Konstan, 2015) and Book-Crossing (Ziegler, McNee, Konstan, & Lausen, 2005) described in Table 2. For both datasets, we considered only users with at least 20 rated items.

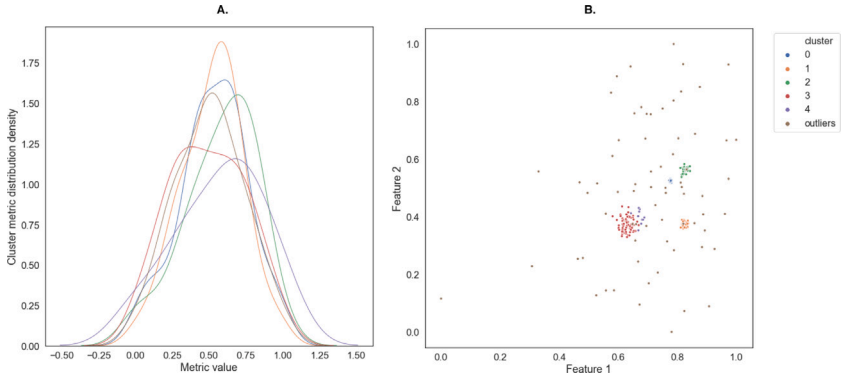

Fig. 1. A synthetic dataset example: A. density distribution of the metric M (on X axis) for each synthetic cluster, calculated by the kernel density estimation method; B. Generated 2D features visualized as the x and y coordinates for each point of a synthetic cluster.

Table 3
User segmentation methods and biases considered in the evaluation experiment.

Algorithm	Cluster characteristics	Hyper-parameters
KM	Behavioral patterns	Number of clusters: 3–50
KM-O	Behavioral patterns & outliers	Number of clusters: 3–50, outlier pct threshold: 0.1–0.5
BAH-KM	Behavioral patterns & metric distribution	–
AG	Behavioral patterns	Number of clusters: 3–50
HDBSCAN	Behavioral patterns & outliers	min. cluster size, min. samples: 10–100
LOF	Outliers	Outlier pct threshold: 0.1–0.5
DEM	DEM groups	Min segment size
ACT	Activity	Activity pct threshold: 0.1–0.5
GC	Popularity	Outlier pct threshold: 0.1–0.5

3.2. Compared user segmentation algorithms

As the choice of an optimal clustering algorithm is dependent on the characteristics of the user embeddings, we compare our proposed approach with the following unsupervised clustering and outlier detection methods, as well as with three methods based on arbitrarily selected protected groups and information about user activity. The detected biases are compared in terms of the metric M:

1. **Bias-Aware Hierarchical Clustering** (BAH-KM) — our proposed Bias-Aware Hierarchical k-Means for detecting the discriminated segments.
2. **k-Means** (Jain & Dubes, 1988) (KM) — standard unsupervised k-Means clustering algorithm for detecting similar groups of users.
3. **Agglomerative clustering** (AG) — a top-down unsupervised hierarchical clustering approach for detecting similar groups of users.
4. **k-Means with outliers** (Ghazanfar & Prügel-Bennett, 2014) (KM-O) — k-Means with outliers detection — the points that are more distant to the nearest cluster center than a threshold α are grouped into a separate segment.
5. **Hierarchical Density-Based Spatial Clustering of Applications with Noise** (HDBSCAN) (Campello, Moulavi, & Sander, 2013) — the outliers are grouped into a separate segment,
6. **Local Outlier Factor** (LOF) (Breunig, Kriegel, Ng, & Sander, 2000) — vectors with the lowest neighborhood density are grouped into a separate cluster, and the rest of the users are put in another cluster (the split ratio is defined by the threshold parameter α). We use the implementation provided by McInnes, Healy, and Astels (2017).
7. **User activity** (ACT) — users with the lowest number of ratings are grouped into a separate cluster (the ratio of users is defined by the threshold parameter α), and the rest of users are put in another cluster.
8. **Global preferences correlation** (GC) — users with the lowest correlation with the global sorting of ratings measured with Pearson correlation coefficient are grouped into a separate cluster, and the rest of users are put in another cluster (the split ratio is defined by the threshold parameter α).
9. **Sensitive attributes** (DEM) — users are grouped according to pre-defined sensitive attributes based on their demography (gender, age, or country) to identify discriminated social groups.

For each of these clustering strategies, we selected the optimal choice of hyper-parameters (defined in Table 3) by maximizing the negative bias for the clusters. For simple baselines based on a single metric (such as ACT, GC and LOF), the threshold was calculated as the percentage of users with the highest or lowest value in the cluster. For instance, a threshold 0.1 for ACT means that 10% of the least active users are considered for the discrimination analysis.

3.3. Compared recommendation algorithms

To show that our proposed method is model-agnostic and may be applied to different recommendation strategies, we evaluated our approach on the following recommendation algorithms. We used three standard collaborative algorithms based on matrix factorization and nearest neighbors (models 1–3), and five recent Deep Learning architectures (models 4–8).

1. Non-negative Matrix Factorization (NMF) (Lee & Seung, 1999),
2. Singular Value Decomposition (SVD) (Koren, Bell, & Volinsky, 2009),
3. User-based k-Nearest Neighbors (KNN),
4. DeepFM (Guo, Tang, Ye, Li, & He, 2017),
5. Attentional Factorization Machine (AFM) (Xiao, Ye, et al., 2017),
6. Deep & Cross Network (DCN) (Wang, Fu, Fu, & Wang, 2017),
7. FiBiNET (Huang, Zhang, & Zhang, 2019),
8. Wide & Deep (WDL) (Cheng et al., 2016).

Each of the factorization models was trained to represent the user profiles as 100-dimensional vectors (based on the interaction in the training set). The DL models (4–8) were trained on the user–item interactions and the metadata with mean squared error minimization, Adam optimization, and early stopping. To ensure that our results can be reproduced, we used publicly available open-source implementation of the algorithms for CF (Hug, 2020) and DL models (Shen, 2018). The models were trained to predict the user–item ratings. We calculated the Normalized Discounted Cumulative Gain (NDCG) (Wang, Wang, Li, He, Liu, & Chen, 2013), which involves a decreasing function of the item's rank (discount) as a measure of how good is the ranking for each user, which is a more realistic measure than the rating prediction error. The measure was divided by the best possible perfect-ranking score to obtain a score between 0 and 1, which made it independent of the user activity rate. We note that while our approach is in general metric-agnostic, if the metric M is dependent on the user activity (such as hit ratio), the detected segments may in fact be less active rather than being discriminated. Hence, the analyzed approach is more useful for the analysis of metrics that are not explicitly related to the users' activity rate.

While the average metric value for each segment was calculated for detecting disparities, following Dmitriev, Gupta, Kim, and Vaz (2017), the final results for each recommendation algorithm were aggregated for the whole dataset rather than as an average of the segment metrics. The dataset was randomly split into a 80% training set and a 20% test set in 10 iterations. We report here the average results from all the test runs. T-test with Bonferroni correction was used for calculating the statistical significance of the results with the following notation: No significance (ns): $0.05 < p \leq 1$; *: $0.01 < p \leq 0.05$; **: $0.001 < p \leq 0.01$; ***: $0.0001 < p \leq 0.001$.

3.4. Evaluation of the detection of discriminated groups

To evaluate how well our proposed algorithm distinguishes diverse user interest groups, first, the quality and characteristics of the constructed user clusters was analyzed by calculating the following metrics:

- Number of clusters
- Silhouette Coefficient (Rousseeuw, 1987), which is a measure of similarity of an object to its own cluster compared to other clusters. Values are from −1 to 1, with 1 indicating a better clustering.
- Davies–Bouldin Index (Davies & Bouldin, 1979) — the average similarity between each cluster and its most similar one. Values are non-negative, with values close to zero indicating a better clustering.

Next, we evaluated the number and biases of the detected discriminated segments compared to the baselines with the following bias detection metrics:

- Lowest metric of a segment with negative bias
- Number of clusters with negative bias
- Number of users in segments with negative bias

3.5. Evaluation of segment prediction

After detecting the discriminated segments, we evaluated the cluster prediction to verify how well we can approximate these groups with the available set of descriptive features. If the prediction quality is low, but the cluster quality is high, it means that there exist some groups in the data that are not described well by the features, and it is necessary to collect other descriptive information. Following Guidotti et al. (2018), we evaluated our approach in terms of fidelity to validate how accurately the post-hoc explanatory model is capable of imitating the behavior of a black-box clustering system $C(u)$. A binary classifier was trained for each of the detected segments, and 10-fold cross-validation was used for evaluating the quality of the cluster prediction with a binary AUC–ROC classification metric. The results are reported as an average metric for the test sets. Additionally, we performed a qualitative analysis of the resulting segment explanations to understand their underlying characteristics. In particular, we analyzed the outputs of the SHAP *summary plot* as visualizations of positive and negative impact of the feature value on the predictor outputs.

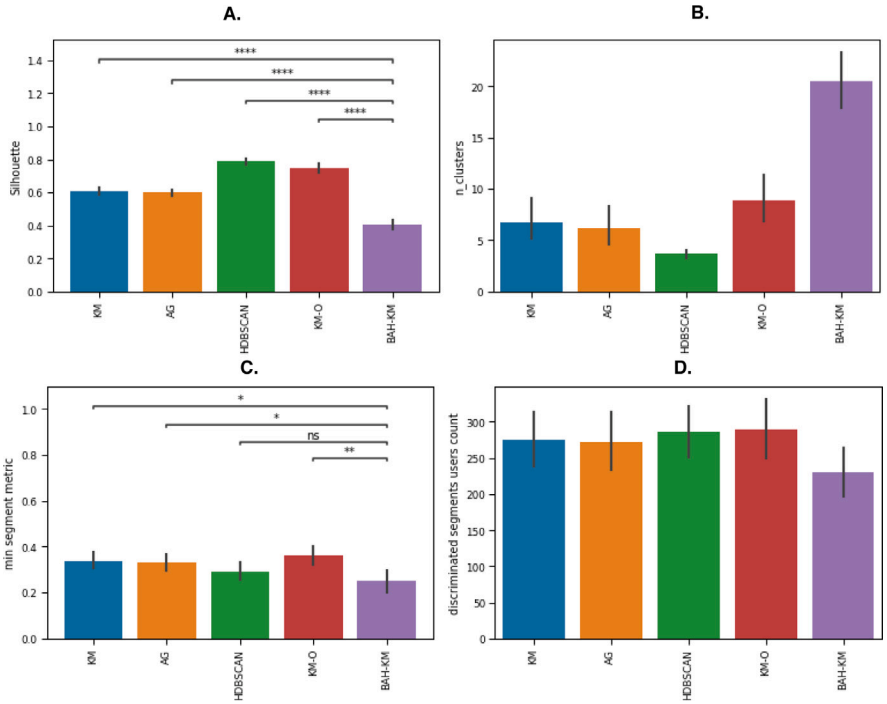

Fig. 2. Synthetic dataset results averaged over 10 runs for different unsupervised clustering algorithms: Silhouette metric (A), number of clusters detected optimized for bias detection (B), result for the most discriminated segment (C) and number of users in discriminated segments (D).

Table 4
Results for the synthetic dataset averaged over 10 runs for different unsupervised clustering algorithms.

Clustering	Silh.	DB	seg. cnt	%outliers	max neg. bias	bias seg. cnt	bias users cnt
AG	0.597	0.667	6.144	0	0.332	1.267	271.956
BAH-KM	0.407	0.810	20.526	0	0.250	1.487	229.895
HDBSCAN	0.787	0.294	3.667	0.192	0.292	1.689	285.978
KM	0.610	0.647	6.778	0	0.338	1.289	275.911
KM-O	0.748	0.333	8.922	0.308	0.362	1.378	289.511

4. Results

In this section, we analyze the results of the experimental tasks with respect to research questions stated at the beginning of Section 3.

4.1. Results for segment discrimination detection

A comparison of the clustering metrics and segment discrimination detection for the synthetic dataset with different unsupervised clustering algorithms is presented in Table 4 and Fig. 2. The cluster quality results are the highest for the two approaches with outlier detection mechanisms (HDBSCAN and KM-O), and HDBSCAN results in the highest Silhouette and lowest DB index values. Our proposed BAH-KM approach results in the highest number of clusters and a lower clustering quality than the other methods. However, this behavior is expected as the clusters are split until a highest bias is detected, and the quality of non-biased segments is not considered for the clustering optimization. Consequently, the BAH-KM method detects a discriminated segment with the lowest metric (though the difference to HDBSCAN is not significant) for the lowest number of users. This means that the discriminated cluster is reduced so that only users with the highest bias are considered.

While the synthetic dataset and the metric contained the discriminated segments by design, similar observations apply to the real-world datasets and recommendation algorithms. Fig. 3 presents the distribution (probability density function) of the NDCG metric for different recommendation algorithms for users in both datasets (MovieLens and Book-Crossing). In both cases, the metric distribution is skewed to the right, which means that there exists a tail of users for whom the recommendations do not perform well. To remedy this, we aim at detecting segments of users for whom the discrimination is most significant.

A comparison of the clustering quality and bias detection for both datasets (averaged for each segmentation method for all recommendation models) is presented in Fig. 4. For all the recommendation models on both datasets, the highest Silhouette results

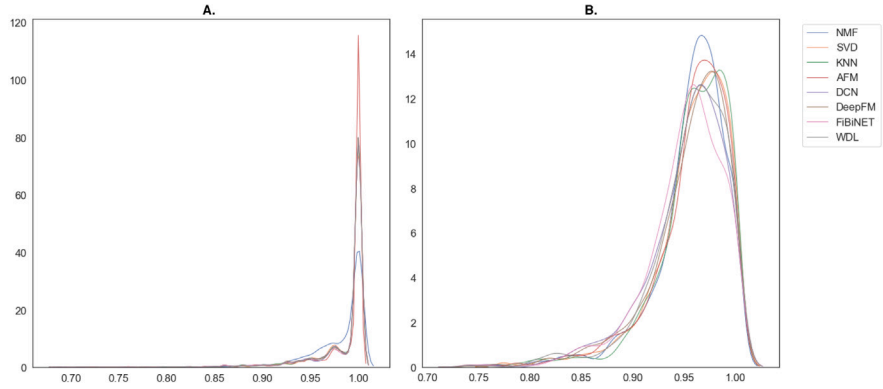

Fig. 3. NDCG metric distribution (probability density function calculated by the kernel density estimation method) for users from Book-Crossing (A) and MovieLens (B) datasets.

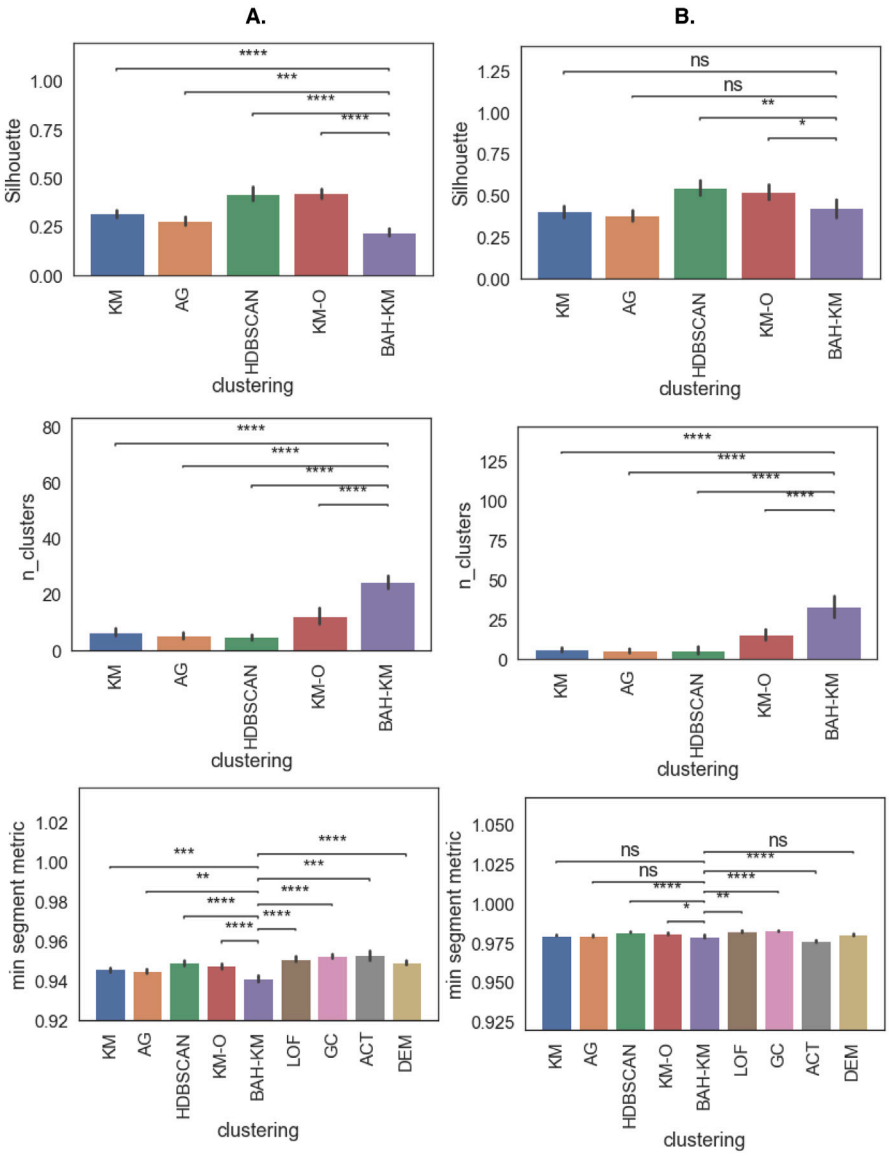

Fig. 4. Clustering quality and cluster bias metrics for MovieLens (A) and Book-Crossing (B) datasets.

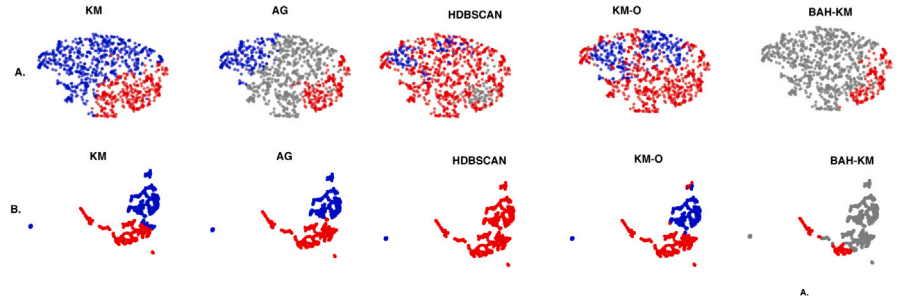

Fig. 5. Examples of UMAP 2D mappings of user embeddings for FiBiNET model for MovieLens (A) and AFM model for Book-Crossing (B). User vectors are colored according to their segment bias detected by different clustering methods for the NDCG metric. Red dots represent the detected potential discrimination (negative bias); blue dots represent segments having significantly higher performance; gray dots represent segments with no statistical difference in terms of the given metric. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

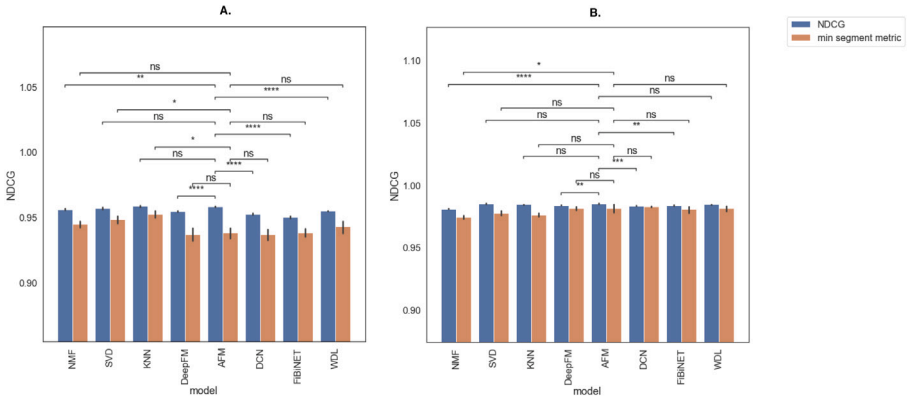

Fig. 6. Recommender metrics and the highest negative bias detected with the Bias Aware Hierarchical K-Means for MovieLens (A) and Book-Crossing (B) datasets. The statistical significance is indicated for the model with the highest average result for each dataset.

are achieved by the outlier-aware methods (HDBSCAN or KM-O). BAH-KM yields significantly lower results in terms of clusters quality compared to other clustering methods for MovieLens, while for Book-Crossing the difference towards KM and AG is not significant. BAH-KM also results in a larger number of segments and the lowest detected discriminated segment metric among all the clustering approaches for MovieLens. However, for the Book-Crossing dataset, the differences between BAH-KM, KM, and AG are not significant, which may be due to the fact that in cases where additional splits do not result in detecting more severe biases, the detected segments may be equivalent to the basic KM method.

The detailed results of the segment bias detection for both datasets for particular models are presented in Table 5. The BAH-KM approach detects the lowest bias segments compared to both the unsupervised clustering methods and the baseline approaches for 6 out of 8 models for the MovieLens dataset (KM detected a lower segment metric for SVD — 0.946 vs. 0.948 and AG for KNN — 0.949 vs. 0.953), and it tends to detect segments of smaller sizes. For Book-Crossing, BAH-KM detects more severe biases than the other unsupervised clustering methods for KNN and SVD algorithms. However, the ACT baseline detects more severe disparities than the unsupervised clustering methods for other recommendation algorithms, which indicates that the bias related to the user activity may be more important for this dataset. We also note that for both the datasets, there are cases when one of the other baseline approaches (DEM, LOF, GC) detects biases in terms of NDCG; however, the minimum segment metric is higher than for the unsupervised methods.

While the main focus of this research is to detect the potential group discrimination (negative biases), information about users for whom the recommendations perform significantly better (positive bias) may also be beneficial. Our approach enables an analysis of both cases. Fig. 5 shows a visualization of examples of discriminated segments as well as segments with a significantly higher metric for different clustering algorithms. While all the methods detect a similar group of discriminated users, BAH-KM algorithm detects the segments with the smallest size, which complies with the previous observations.

Fig. 6 shows the average NDCG results and the results of the most discriminated segment detected with BAH-KM for each of the recommendation models for both the datasets. We note that while the difference between the average result for AFM and KNN methods on MovieLens dataset is not significant, the AFM model results in more severe discrimination of one of the segments. For Book-Crossing, the results for the lowest segment metric are comparable for all the methods.

As presented in Fig. 7, the metric value for segments with negative biases detected by the BAH-KM algorithm is in general lower for small segments than for the larger ones. This may indicate that the minority groups may suffer from more severe disparities than the larger segments.

Table 5
Results of discriminated segment detection for different recommendation models and user segmentation methods for MovieLens and Book-Crossing datasets.

Model	Clustering	MovieLens			Book-Crossing		
		min cluster NDCG	discr. seg. cnt	discr. users cnt	min cluster NDCG	discr. seg. cnt	discr. users cnt
AFM	ACT	0.956	0.10	47	0.978	1	1061.1
	AG	0.942	1.20	176.4	0.98	1.1	788.5
	BAH-KM	0.938	1.6	131.3	0.982	1.7	1573.4
	DEM	0.953	0.25	64.9	0.983	0.45	438.2
	GC	0.956	0.20	122.1	0.984	0.8	1152.2
	HDBSCAN	0.956	1	612	0.983	1.2	1679.2
	KM	0.943	1.10	149.8	0.980	1.1	850.7
	KM-O	0.945	0.90	187.5	0.981	1.	815.6
	LOF	0.955	0.40	197.8	0.984	0.3	317.8
DCN	ACT	0.949	0.10	47.1	0.976	1.	1061.1
	AG	0.943	1.20	374.5	0.981	1.	1150.4
	BAH-KM	0.937	1.90	300.6	0.983	1.	1999.8
	DEM	0.947	0.25	68.2	0.980	0.4	360.9
	GC	0.95	0.40	225.8	0.982	0.6	909.7
	HDBSCAN	0.945	1.	361.3	0.983	1.	1999.8
	KM	0.945	1.30	388.9	0.981	1.	1031.9
	KM-O	0.944	0.50	97.5	0.981	0.7	660.9
	LOF	0.95	0.20	94.1	0.982	0.4	568.3
DeepFM	ACT	–	0.	0.	0.977	1.	1061.1
	AG	0.945	1.	258.2	0.981	1.	1105.4
	BAH-KM	0.937	1.80	243.2	0.981	1.1	1479.5
	DEM	0.95	0.20	54.6	0.980	0.3	150.8
	GC	0.953	0.40	263.4	0.982	0.6	830.4
	HDBSCAN	0.951	1.	469.9	0.983	1.	1974.8
	KM	0.946	0.90	216.2	0.981	1.	1103.4
	KM-O	0.949	0.80	233.4	0.980	0.8	704.5
	LOF	0.95	0.50	178.8	0.983	0.2	315.8
FiBiNET	ACT	0.948	0.10	47.	0.976	1.	1061.1
	AG	0.945	1.	390.9	0.980	1.	991.3
	BAH-KM	0.938	1.10	244.9	0.981	1.3	1635.6
	DEM	0.945	0.35	92.250	0.980	0.4	359.5
	GC	0.948	0.80	536.1	0.982	0.4	513.6
	HDBSCAN	0.945	1.	353.9	0.983	1.	1999.3
	KM	0.943	1.	311.4	0.980	1.	937.7
	KM-O	0.945	1.	320.	0.981	0.8	728.3
	LOF	0.946	0.60	291.8	0.983	0.4	638.8
KNN	ACT	0.956	0.20	131.9	0.978	1.	1021.3
	AG	0.949	0.50	42.5	0.981	1.1	508.8
	BAH-KM	0.953	0.50	118.6	0.976	2.	247.667
	DEM	0.953	0.20	54.4	0.981	0.150	125.350
	GC	0.957	0.70	536.7	0.983	0.4	638.
	HDBSCAN	0.954	0.40	98.5	0.982	1.	938.8
	KM	0.951	0.60	66.8	0.980	1.	406.8
	KM-O	0.952	0.50	126.4	0.981	0.8	384.2
	LOF	0.957	0.30	235.5	0.983	0.4	409.2
NMF	ACT	0.955	0.10	47.	0.972	1.	1021.3
	AG	0.949	0.80	141.6	0.976	0.8	269.9
	BAH-KM	0.945	1.20	129.7	0.974	1.4	199.
	DEM	0.952	0.30	81.7	0.978	0.450	304.750
	GC	0.953	0.30	197.8	0.980	0.2	134.9
	HDBSCAN	0.946	0.40	36.6	0.980	1.	1402.8
	KM	0.95	1.10	226.6	0.979	0.7	372.1
	KM-O	0.951	0.80	253.4	0.979	0.4	408.2
	LOF	0.948	0.10	9.4	0.980	0.1	163.6
SVD	ACT	0.953	0.20	131.9	0.979	1.	1021.3
	AG	0.947	0.20	14.	0.979	0.8	215.1
	BAH-KM	0.948	0.90	131.9	0.978	1.3	195.
	DEM	0.95	0.15	40.8	0.982	0.2	149.150
	GC	0.954	0.50	339.	0.984	0.8	1354.6
	HDBSCAN	0.952	0.30	65.7	0.982	0.5	550.5
	KM	0.946	0.30	29.7	0.979	0.7	119.7
	KM-O	0.952	0.40	93.7	0.983	0.6	534.4

(*continued on next page*)

Table 5 (*continued*).

Model	Clustering	MovieLens			Book-Crossing		
		min cluster NDCG	discr. seg. cnt	discr. users cnt	min cluster NDCG	discr. seg. cnt	discr. users cnt
	LOF	0.953	0.50	169.5	0.984	0.4	469.7
WDL	ACT	–	0.	0.	0.977	1.	1061.1
	AG	0.947	1.	384.	0.981	1.	1071.8
	BAH-KM	0.943	1.50	388.375	0.981	1.222	1315.333
	DEM	0.949	0.35	81.250	0.981	0.350	282.050
	GC	0.952	0.60	329.3	0.984	0.5	824.6
	HDBSCAN	0.95	1.10	460.6	0.983	1.	1817.9
	KM	0.947	1.30	437.7	0.981	1.	1079.4
	KM-O	0.949	0.90	291.1	0.982	0.8	759.
	LOF	0.953	0.40	263.6	0.983	0.2	339.

Table 6
Average AUC–ROC for post-hoc segment prediction.

Dataset	NMF	SVD	KNN	DeepFM	AFM	DCN	FiBiNET	WDL
MovieLens	0.739	0.555	0.536	0.739	0.854	0.730	0.817	0.776
Book-Crossing	0.599	0.486	0.578	0.781	0.86	0.9	0.792	0.847

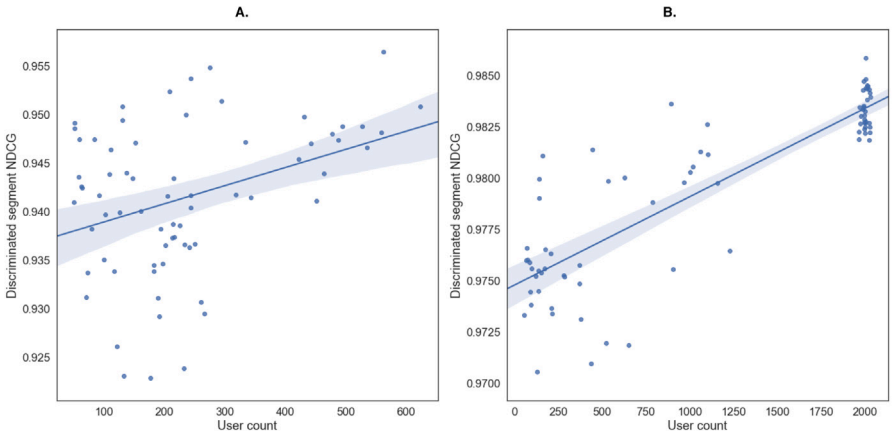

Fig. 7. Correlation between the number of users (X) and the metric value (Y) in the discriminated segments detected with BAH-KM algorithm for MovieLens (A) and Book-Crossing (B) datasets.

4.2. Segment prediction results

An evaluation of the segment prediction quality for different recommendation models (measured as AUC–ROC for the binary task of segment prediction) is presented in Table 6.

In general, the AUC–ROC metric is higher for the Deep Learning recommenders than for the standard CF approaches, with the highest score for DCN on Book-Crossing (AUC–ROC 0.9) and AFM for MovieLens (0.854). However, for the standard CF models, the quality of predictions is sometimes low, which indicates that there are some additional aspects, not described by the available features, which are leading to discrimination.

As examples of detected disparities, some explanations of SHAP analysis for discriminated segments for different algorithms on MovieLens dataset are presented in Fig. 8. In both cases, users in particular segments are less interested in drama movies than an average person. For the movie dataset, the DCN model tends to discriminate the segment of female users who highly rated the old film-noir movies, while the AFM model has worse performance for the fans of action, thriller, and adventure movies. This analysis indicates what types of users are not satisfied with the recommendations and enables taking appropriate remedial actions.

4.3. Results summary and implications

An analysis of the experimental results shows that applying unsupervised clustering methods is helpful in detecting biases in the recommendation algorithms performance for different user groups. A comparison with commonly used segments constructed from arbitrarily selected sensitive user attributes shows that automatically detected segments may represent more significant potential disparities without requiring any a priori assumptions.

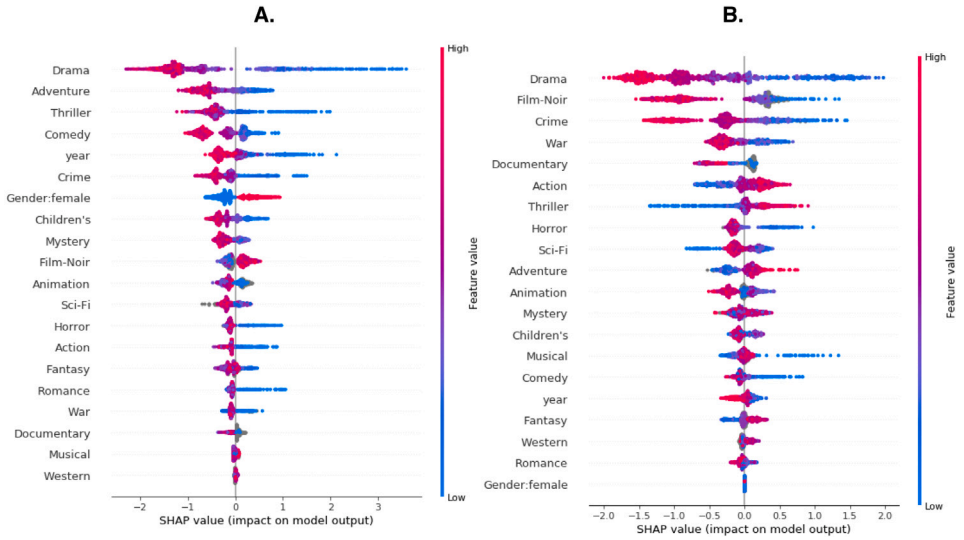

Fig. 8. Examples of the impact of features on classifier predictions for discriminated segments detected with BAH-KM for DCN (A) and AFM (B) models for MovieLens dataset generated with SHAP method. The descriptive features are ranked in a descending order and plotted for each data point (user) — the horizontal location shows whether the effect of that value is associated with a higher or lower prediction, while the color shows the feature value (high — red and low — blue). A. Positive values of *female* feature have a positive impact on the assignment to the discriminated cluster, so does the interest in *film-noir*, while positive values of features *drama, adventure and thriller*, as well as recent production year, have a negative impact. B. Positive values of interest in *action, thriller* and *sci-fi* movies have a positive impact on the assignment to the discriminated cluster, in contrast to a negative impact of *drama, film-noir* and *crime* features. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

As for the proposed BAH-KM approach, while the clustering quality in terms of cluster density and separateness is lower than for the other clustering methods (due to a relatively large number of non-biased clusters produced), it detects the most severely biased segments in most of the analyzed recommendation settings for both synthetic and real-world datasets.

Additionally, the segment descriptions generated from the interpretable user features enable identifying the hidden disparities.

The discussion on the potential limitations and implications of the proposed method is presented in the following sections.

4.3.1. Limitations

In our proposed BAH-KM approach, we construct clusters by maximizing the bias detected for metric M, which makes clustering dependent on the metric. For instance, a group of users may be discriminated in terms of the mean squared error of their rating prediction or the diversity of recommended items, while the performance in terms of click prediction may be comparable to others. Alternatively, in the standard unsupervised clustering approaches, clusters formed by maximizing the cluster quality are considered the ground truth about the user representation characteristics and discrimination detection is applied on the top of these detected clusters. This approach may be used to detect discrimination for different evaluation metrics on the same groups of users; however, BAH-KM method detects more severe biases than the other methods in most cases.

We note that while the discrimination detection stage may be applied to any representation type, segment descriptions are generated based on additional metadata (*descriptive features*), and their quality is dependent on the availability of this information. This aspect may constitute a potential limitation of our approach for some domains, as it assumes that the set of descriptive features is sufficient for explaining the latent characteristics of user behaviors; hence it makes sense only if enough metadata is available. In cases where additional information cannot be provided, we suggest using item names instead. In such cases, descriptions will be less general, which may require more cognitive effort to analyze, but they may still be useful for discovering latent characteristics and biases.

Additionally, while the use of the post-hoc explainer model and Shapley values analysis is useful in describing and understanding what groups of users are discriminated, it only shows correlations and descriptive statistics of features with the output rather than causal relationships (Rudin, 2018).

4.3.2. Implications — addressing the discrimination problem

We would like to point out that though our method is capable of detecting groups of users who are potentially discriminated by a given algorithm, how to address their needs should be decided by the system owner or the content provider according to a particular application. We took a universal model and metric-agnostic approach, which may be applied to different business applications. The reasons for the discrimination may also be diverse — for instance, as observed in Misztal-Radecka et al. (2019), the pool of available items may not address the diversity of user needs, or some algorithmic bias may be observed (Baeza-Yates, 2018; Bolukbasi et al., 2016; Caliskan et al., 2017; Graells-Garrido et al., 2015).

The most straightforward way to incorporate bias detection in the recommendation design process is to include information about the discriminated segments in the model and the parameter selection process. For instance, as shown in Fig. 6, while AFM yields comparable results to KNN for MovieLens, it also results in a lower metric for the most discriminated segment. Hence, selecting a method for which disparity is lower may result in more fair recommendations.

In some cases, when the set of users is diverse, various approaches work for particular types of users, and it may be beneficial to combine different methods for different individuals. For instance, if there is a group of users with specific tastes for whom the global CF recommender does not perform well, it may be combined with a content-based strategy that may be more accurate for this group. We also plan to incorporate debiasing techniques in our approach, such as adding fairness objectives (Yao & Huang, 2017) to the model training and selection processes.

Finally, the segment descriptions generated from the interpretable user features provide insights into the characteristics of potentially discriminated users, and enable taking remedial actions to address their needs, such as increasing the items diversity or the diversity of users in the training set.

5. Conclusions and future work

We presented an approach to detecting and describing potentially discriminated groups of users for a given recommendation algorithm. We proposed an unsupervised Bias-Aware Hierarchical Clustering method that automatically detects these segments, hence does not require any a priori information about existing disparities. Additionally, we applied a post-hoc explanation technique to provide human-understandable descriptions for each of the discriminated segments to help in identifying the potential algorithmic biases. The experimental evaluation on two distinct real-world, as well as a synthetic dataset, showed that our method is capable of detecting under-performing segments and describing their characteristics for different recommendation algorithms.

In future work, we plan to focus on addressing the detected discrimination problem for particular recommendation algorithms to better address the needs of unsatisfied groups of users. We are also experimenting on other datasets from different domains and various other recommendation algorithms, including top-N and online recommendations, to analyze potential algorithmic biases in various recommendation settings. Additionally, as our method is model and metric agnostic, it may be generalized to detect disparities with respect to recommended objects as well.

CRediT authorship contribution statement

Joanna Misztal-Radecka: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Visualization. **Bipin Indurkhya:** Conceptualization, Writing - review & editing, Supervision, Project administration.

Acknowledgment

The research presented in this paper was supported by funds from the Polish Ministry of Science and Higher Education assigned to the AGH University of Science and Technology.

References

Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms* (pp. 1027–1035). USA: Society for Industrial and Applied Mathematics.

Baeza-Yates, R. (2018). Bias on the web. *Communications of the ACM*, *61*, 54–61. http://dx.doi.org/10.1145/3209581.

Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and machine learning*. fairmlbook.org, http://www.fairmlbook.org.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems (vol. 29)* (pp. 4349–4357). Curran Associates, Inc.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on management of data* (pp. 93–104). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/342009.335388.

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, *356*(6334), 183–186. http://dx.doi.org/10.1126/science.aal4230, arXiv:https://science.sciencemag.org/content/356/6334/183.full.pdf, URL https://science.sciencemag.org/content/356/6334/183.

Campello, R. J. G. B., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, & G. Xu (Eds.), *Advances in knowledge discovery and data mining* (pp. 160–172). Berlin, Heidelberg: Springer Berlin Heidelberg.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. arXiv:1603.02754, CoRR, abs/1603.02754, URL http://arxiv.org/abs/1603.02754.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., et al. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems* (pp. 7–10). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2988450.2988454.

Chu, W., Park, S.-T., Beaupre, T., Motgi, N., Phadke, A., Chakraborty, S., et al. (2009). A case study of behavior-driven conjoint analysis on yahoo!: Front page today module. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1097–1104). New York, NY, USA: ACM, http://dx.doi.org/10.1145/1557019.1557138, URL http://doi.acm.org/10.1145/1557019.1557138.

Das, A. S., Datar, M., Garg, A., & Rajaram, S. (2007). Google news personalization: Scalable online collaborative filtering. In *Proceedings of the 16th international conference on world wide web* (pp. 271–280). New York, NY, USA: ACM, http://doi.acm.org/10.1145/1242572.1242610, URL http://portal.acm.org/citation.cfm?id=1242572.1242610.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*(2), 224–227. http://dx.doi.org/10.1109/TPAMI.1979.4766909.

Dmitriev, P., Gupta, S., Kim, D. W., & Vaz, G. (2017). A dirty dozen: Twelve common metric interpretation pitfalls in online controlled experiments. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1427–1436). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3097983.3098024.

Eskandanian, F., Sonboli, N., & Mobasher, B. (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* (pp. 225–233). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3320435.3320464.

Gajane, P. (2017). On formalizing fairness in prediction with machine learning. arXiv:1710.03184, CoRR, abs/1710.03184, http://arxiv.org/abs/1710.03184.

Ghazanfar, M. A., & Prügel-Bennett, A. (2014). Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. *Expert Systems with Applications*, *41*(7), 3261–3275.

Graells-Garrido, E., Lalmas, M., & Menczer, F. (2015). First women, second sex: Gender bias in wikipedia. In *Proceedings of the 26th ACM conference on hypertext & social media* (pp. 165–174). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2700171.2791036.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Computing Surveys*, *51*, http://dx.doi.org/10.1145/3236009.

Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). DeepFM: A factorization-machine based neural network for CTR prediction. arXiv:1703.04247, CoRR, abs/1703.04247, URL http://arxiv.org/abs/1703.04247.

Harper, F. M., & Konstan, J. A. (2015). The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, *5*(4), 19:1–19:19. http://dx.doi.org/10.1145/2827872, URL http://doi.acm.org/10.1145/2827872.

Hennig, C. (2019). Cluster validation by measurement of clustering characteristics relevant to the user. In *Data analysis and applications (vol. 1)* (pp. 1–24). John Wiley & Sons, Ltd, http://dx.doi.org/10.1002/9781119597568.ch1, Ch. 1. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119597568.ch1.

Huang, T., Zhang, Z., & Zhang, J. (2019). FiBiNET: Combining feature importance and bilinear feature interaction for click-through rate prediction. arXiv:1905.09433, CoRR, abs/1905.09433, URL http://arxiv.org/abs/1905.09433.

Hug, N. (2020). Surprise: A Python library for recommender systems. *Journal of Open Source Software*, *5*(52), 2174. http://dx.doi.org/10.21105/joss.02174.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by nonnegative matrix factorization. *Nature*, *401*, 788–791.

Legislation, E. U. (2009). Equal treatment of persons. URL https://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX%3A52008PC0426.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., et al. (2019). Explainable AI for trees: From local explanations to global understanding. arXiv preprint arXiv:1905.04610.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems (vol. 30)* (pp. 4765–4774). Curran Associates, Inc.

Luong, B. T., Ruggieri, S., & Turini, F. (2011). K-NN as an implementation of situation testing for discrimination discovery and prevention. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 502–510). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2020408.2020488.

McInnes, L., Healy, J., & Astels, S. (2017). Hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, *2*(11), http://dx.doi.org/10.21105/joss.00205.

McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv:1802.03426.

Misztal-Radecka, J., & Indurkhya, B. (2020). Getting to know your neighbors (KYN). Explaining item similarity in nearest neighbors collaborative filtering recommendations. In *Adjunct publication of the 28th ACM conference on user modeling, adaptation and personalization* (pp. 59–64). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3386392.3397599.

Misztal-Radecka, J., Indurkhya, B., & Smywiński-Pohl, A. (2020). Meta-User2Vec model for addressing the user and item cold-start problem in recommender systems. http://dx.doi.org/10.1007/s11257-020-09282-4,

Misztal-Radecka, J., Rusiecki, D., Żmuda, M., & Bujak, A. (2019). Trend-responsive user segmentation enabling traceable publishing insights. A case study of a real-world large-scale news recommendation system. In *CEUR Workshop Proceedings*: *vol. 2554*, *Proceedings of the 7th international workshop on news recommendation and analytics in conjunction with the 13th ACM conference on recommender systems* (pp. 53–62). CEUR-WS.org, URL http://ceur-ws.org/Vol-2554/paper_08.pdf.

Nasiriani, N., Squicciarini, A., Saldanha, Z., Goel, S., & Zannone, N. (2019). Hierarchical clustering for discrimination discovery: A top-down approach. In *2019 IEEE second international conference on artificial intelligence and knowledge engineering* (pp. 187–194).

Nurma Sari, J., Nugroho, L., Ferdiana, R., & Santosa, P. (2016). Review on customer segmentation technique on ecommerce. *Advanced Science Letters*, *22*, 3018–3022. http://dx.doi.org/10.1166/asl.2016.7985.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Model-agnostic interpretability of machine learning. arXiv:1606.05386, CoRR, abs/1606.05386, URL http://arxiv.org/abs/1606.05386.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. http://dx.doi.org/10.1016/0377-0427(87)90125-7, URL http://www.sciencedirect.com/science/article/pii/0377042787901257.

Rudin, C. (2018). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. arXiv:1811.10154.

Sacharidis, D. (2019). Top-n group recommendations with fairness. In *Proceedings of the 34th ACM/SIGAPP symposium on applied computing* (pp. 1663–1670). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3297280.3297442.

Sánchez, P., & Bellogín, A. (2019). Attribute-based evaluation for recommender systems: Incorporating user and item attributes in evaluation metrics. In *Proceedings of the 13th ACM conference on recommender systems* (pp. 378–382). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3298689.3347049.

Sarwar, B. M., Karypis, G., Konstan, J., & Reidl, J. (2002). Recommender systems for large-scale E-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the 5th international conference on computer and information technology*.

Serbos, D., Qi, S., Mamoulis, N., Pitoura, E., & Tsaparas, P. (2017). Fairness in package-to-group recommendations. In *Proceedings of the 26th international conference on world wide web* (pp. 371–379). Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, http://dx.doi.org/10.1145/3038912.3052612.

Shen, W. (2018). DeepCTR: Easy-to-use,modular and extendible package of deep-learning based CTR models. *GitHub Repository*, https://github.com/shenweichen/deepctr.

Singh, J., & Anand, A. (2018). Posthoc interpretability of learning to rank models using secondary training data. arXiv:1806.11330, CoRR, abs/1806.11330, URL http://arxiv.org/abs/1806.11330.

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. In *KDD workshop on text mining*.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, *2009*, http://dx.doi.org/10.1155/2009/421425.

Tintarev, N., & Masthoff, J. (2011). Designing and evaluating explanations for recommender systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 479–510). Springer US, http://dx.doi.org/10.1007/978-0-387-85820-3_15.

Tsintzou, V., Pitoura, E., & Tsaparas, P. (2018). Bias disparity in recommendation systems. arXiv:1811.01461, CoRR, abs/1811.01461, URL http://arxiv.org/abs/1811.01461.

Wang, R., Fu, B., Fu, G., & Wang, M. (2017). Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3124749.3124754.

Wang, Y., Wang, L., Li, Y., He, D., Liu, T., & Chen, W. (2013). A theoretical analysis of NDCG type ranking measures. arXiv:1304.6480, CoRR, abs/1304.6480, URL http://arxiv.org/abs/1304.6480.

Xiao, L., Min, Z., Yongfeng, Z., Zhaoquan, G., Yiqun, L., & Shaoping, M. (2017). Fairness-aware group recommendation with Pareto-efficiency. In *Proceedings of the eleventh ACM conference on recommender systems* (pp. 107–115). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3109859.3109887.

Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., & Chua, T.-S. (2017). Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence* (pp. 3119–3125). http://dx.doi.org/10.24963/ijcai.2017/435.

Yao, S., & Huang, B. (2017). Beyond parity: Fairness objectives for collaborative filtering. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems (vol. 30)* (pp. 2921–2930). Curran Associates, Inc.

Zhang, Y., & Chen, X. (2020). Explainable recommendation: A survey and new perspectives. *Foundations and Trends in Information Retrieval*, *14*(1), 1–101. http://dx.doi.org/10.1561/1500000066.

Zheng, Y., Agnani, M., & Singh, M. (2017). Identifying grey sheep users by the distribution of user similarities in collaborative filtering. In *Proceedings of the 6th Annual Conference on Research in Information Technology* (pp. 1–6). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3125649.3125651.

Ziegler, C.-N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on world wide web* (pp. 22–32). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/1060745.1060754.