Fairness and Popularity Bias in Recommender Systems: an Empirical Evaluation

Cataldo Musto[1], Pasquale Lops[1] and Giovanni Semeraro[1]

[1]*Department of Computer Science, University of Bari Aldo Moro, Bari, Italy*

Abstract

In this paper, we present the results of an empirical evaluation investigating how *recommendation algorithms* are affected by *popularity bias.* Popularity bias makes more popular items to be recommended more frequently than less popular ones, thus it is one of the most relevant issues that limits the *fairness* of recommender systems. In particular, we define an experimental protocol based on two state-of-the-art datasets containing users' preferences on *movies* and *books* and three different recommendation paradigms, i.e., *collaborative filtering, content-based filtering and graph-based algorithms.* In order to evaluate the overall *fairness* of the recommendations we use well-known metrics such as *Catalogue Coverage*, *Gini Index* and *Group Average Popularity* (ΔGAP). The goal of this paper is: *(i)* to provide a clear picture of how recommendation techniques are affected by popularity bias; *(ii)* to trigger further research in the area aimed to introduce methods to mitigate or reduce biases in order to provide *fairer* recommendations.

Keywords

Recommender Systems, Popularity Bias, Fairness

1. Introduction

Recommender Systems (RSs) guide the users in a *personalized* way to interesting or useful objects in domains where a large space of possible options are available [1]. Basically, such systems acquire information about users' needs, interests and preferences and tailor their behavior based on such information, by supporting people in several decision-making tasks [2]. Nowadays, it is acknowledged that RSs have a huge influence on consumers' behaviors. Indeed, many people use these systems to listen to music on *Spotify*, to watch videos on *YouTube* or to buy products on *Amazon.* As shown in [3], such algorithms have a significant impact on both sales volumes and clickthrough rates. As an example, 35% of Amazon's revenues are generated through its recommendation engine[1].

Although RS research traditionally focused on providing users with *accurate* recommendations, that is to say, recommendations that match user interests, recent studies have assessed the importance of additional factors for evaluating the perceived quality and usefulness of recommendation lists. As an example, several works evaluated to what extent a recommendation

AIxIA 2021 Discussion Papers, 20th International Conference Italian Association for Artificial Intelligence, December 1-3 2021, Online, Editors: Viviana Mascari, Matteo Palmonari, Giuseppe Vizzari

cataldo.musto@uniba.it (C. Musto); pasquale.lops@uniba.it (P. Lops); giovanni.semeraro@uniba.it (G. Semeraro)
0000-0001-6089-928X (C. Musto); 0000-0002-6866-9451 (P. Lops); 0000-0001-6883-1853 (G. Semeraro)

[1] http://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers

algorithm is able to expose a user with *diverse*, *novel* or *serendipitous* recommendations [4, 5, 6]. The metrics that allow to quantitatively assess the aforementioned properties of recommendations are typically referred to as *beyond-accuracy metrics* [7, 8]. An example of beyond-accuracy metric which recently gained more and more attention is the *fairness.* Abstractly, by referring to AI methods and techniques, fairness means *to not discriminate against individuals or groups* [9]. As for classification algorithms, a behavior is defined as *fair* if the outcome of the algorithm (*e.g.*, a binary answer to an applicant seeking a loan) is not influenced by personal characteristics of the user, such as gender or race [10].

As for recommendation algorithms, the concept of fairness becomes more complex and *multi-sided* [11], since it can refer to both *users* and *items.* The first sense follows the general definition already introduced for classification algorithms, since a recommender system is fair w.r.t. users if *their personal characteristics do not influence the behavior of the RS.* On the other side, an algorithm is fair w.r.t. items if the recommendation list contains items whose characteristics reflect the preferences of the user. As discussed in [12], if a user has liked 7 romance and 3 action movies, a *fair* recommendation list should contain 70% romance and 30% action movies. Similarly, if a user typically likes *niche* items, that is to say, poorly popular items, her recommendation list should contain a majority of niche items as well.

However, such an *ideal* behavior is far from being real, since several factors *negatively* affect the fairness of recommendation lists. One of the most popular issues that affect the fairness is commonly known as *popularity bias*: indeed, as shown by several studies [12], users mostly provide feedback on *popular* items rather than on niche ones. This introduces a bias towards popular items that tend to be recommended more frequently w.r.t. niche ones, and this is a clear obstacle for the generation of fair recommendation lists.

Even though the problem has been largely discussed in literature [13], to the best of our knowledge the analysis of how the different recommendation paradigms are affected by popularity bias (and consequently provide unfair recommendations) is under-investigated. Accordingly, through this paper we aim to fill in this gap and provide a benchmark for the *fairness* of the popular recommendation paradigms based on the suggestions they provide. In particular, we analyze several implementations of *collaborative filtering*, *content-based* and *graph-based* RSs and we evaluate them in terms of metrics for assessing the fairness of the algorithms, such as *catalogue coverage*, *Gini Index* and *Group Average Popularity.*

The rest of the paper is organized as follows: in Section 2, we briefly introduce other works discussing the impact and the benefits of fairness in recommendation algorithms. Next, in Section 3 we present the recommendation algorithms we evaluated in our experimental protocol described in Section 4. In Section 5 we discuss the results of our benchmark and we sketch the main findings of this work. Finally, Section 6 draws the conclusions and summarizes some ideas for future research in the area.

2. Related Work

The problem of *popularity bias* is connected to the well-known phenomenon of the *long-tail* [14]. This concept, which refers to the way data are distributed and observed, is based on Zipf's law [15, 16] and holds for several scenarios, ranging from wealth distribution to use of terms in a particular language. Zipf's law states that, if a collection of items is ranked by popularity, the second item will have around half the popularity of the first one, and the third item will have about a third of the popularity of the first one, and so on. Accordingly, the long tail theory shows that a tiny amount of objects receives a huge amount of observations (*e.g.,* clicks, likes, purchases. depending on the context), while the majority of the objects (the long tail) receives a smaller amount of observation.

As previously stated, the problem also holds for RSs, since just a few objects receive most of the feedbacks provided by the users. The phenomenon has been largely observed by Jannach et al. [17], who presented a detailed analysis of *what recommenders recommend.* As shown in the article, due to the long tail (and, in turn, to the popularity bias) popular items are more frequently recommended, and this leads to the undesired *blockbuster effect* [18]. It is not by chance that recommending popular items represents a very strong baseline in offline evaluations with respect to accuracy measures [19, 20], Unfortunately, as previously stated, this limits the overall fairness of the recommendation lists. Indeed, as shown in [21], it is important that RSs achieve a good balance between popular and less-popular items.

The nature of the *popularity bias* and the challenges it poses are discussed in several works. This has been done by both analyzing users' rating behavior [22] as well as by proposing new algorithms to control the bias and better reward items in the long tail [21, 13, 23]. Similarly, the concept of fairness in recommendation received a lot of attention [24]. As an example, Zhu et al. [25] proposed an approach to remove discrimination based on demographic features. Similarly, in [26] a method to provide a fair exposure to recommendation items is presented.

In this work we follow the protocol presented in [13], and we focus on the fairness of recommendations with respect to users' expectations. In other terms, we aim to *analyze to what extent the items in the recommendation lists follow the distribution of the actual interests of users* with respect to how many popular items they expect to see in the recommended list. A similar attempt is presented in [27], where an empirical analysis in music domain is carried out, and in [12], where the author proposed the idea of *calibration*: the recommendations should be consistent with the average popularity of the items rated by the users. However, differently from these pieces of work, in this paper we analyze the behavior of different recommendation paradigms, *i.e.*, collaborative filtering, content-based recommender systems and graph-based recommendations, in order to analyze how different algorithms are affected by popularity bias.

3. Recommendation Paradigms

In this section, we briefly introduce the *recommendation paradigms* we analyzed in this work. A thorough analysis of strengths and weaknesses of each group of algorithms goes beyond the scope of the paper, and we suggest to refer to [1] for a complete overview of the topic. In the following, we will introduce the basics of *collaborative filtering* techniques, followed by *content-based* and *graph-based recommender systems.*

3.1. Collaborative Filtering Algorithms

Collaborative filtering (CF) algorithms represent the *most popular* and probably widely available implementation of a recommendation algorithm [28]. The basic idea of CF algorithms is that users who shared the same interests in the past (*e.g.,* viewed the same movies or bought the same books) will also like similar items in the future. Generally speaking, CF systems generate recommendations for the target user based on the preferences expressed by similar users. The concept of *similarity* is based on users' previous behaviors. In a nutshell, if they liked or they bought the same items, they are similar [29]. Such an intuition is concretely implemented by means of a *user-item* matrix, where users are put in the rows, items are put in the columns, and the feedback provided by the user on that item (*e.g.,* bought, rated, viewed, etc.) is encoded at the cross of row and column.

These algorithms have been popularized by the well-known Netflix prize [30], where the winning approach exploited a more sophisticated version of CF [31] based on the factorization of the user-item matrix [32]. As shown in [33], these methods are still very popular [34] and also extended to neural approaches [35]. However, as shown by Lops et al. [36], CF algorithms are strongly affected by *sparsity* issues and cold-start, *i.e.*, they can not provide good recommendations if just a few ratings is available. As a consequence, the research also started investigating *content-based* and *hybrid* approaches [37].

As we will show in the next section, as CF algorithms we considered: *(i)* basic implementations of standard techniques, such as *item-to-item* and *user-to-user* collaborative filtering techniques; *(ii)* matrix factorization (MF) techniques, such as Biased MF, FunkSVD and other methods.

3.2. Content-based Recommender Systems

The *social* nature of collaborative filtering algorithms makes CF poorly suitable when few ratings are available. This issue is completely put aside by content-based recommender systems (CBRS) [5], which typically recommend items that are similar to the ones the user liked in the past. As an example, if a user has positively rated a movie that belongs to the *comedy* genre, then it is likely that the system will suggest other movies labeled with this genre.

Generally speaking, the recommendation process is based on the estimation of *how similar* the recommended item is w.r.t. the profile of the user. Such a similarity, which is based on popular and well-known measures (*e.g.,* cosine similarity, Euclidean distance, etc.), is calculated based on the *attributes* associated to both the item and the profile of the user. Basically, the more the overlap between the attributes, the higher the similarity.

In some cases, attributes are simple keywords that are extracted from the item descriptions, such as the content of a news or the plot of a movie. However, more sophisticated approaches that exploit more accurate and advanced techniques based on natural language processing also exist. As stated in [5], *semantics-aware techniques* which learn a representation of the items based on the meaning of the attributes (rather than on simple keywords) recently gained attention thank to the good accuracy they provide [38]. As an example, Ozsoy et al. [39] proposed the use of Word2Vec to learn word embeddings representing items and user profiles. Moreover, in [40] Doc2Vec is used to learn an embedding representing a news article, based on the text and the title of the news, while FastText is used in [41] in a content-based recommendation scenario. Other shreds of evidence concerning semantics-aware recommendation methods exploiting word embeddings [42, 43] definitely confirm these claims.

As for CBRS, in this paper we will both take into account: *(i)* early CBRS implementations, based on a vector space representation of users and items with TF-IDF weighting; *(ii)* semantics-aware methods, i.e., based on Doc2Vec [44], Word2Vec [45], and LSI.

3.3. Graph-based Recommender Systems

Graphs provide a very *natural and straightforward* representation model to encode all the entities involved in the recommendation process. Indeed, *users*, *items* and *attributes* can be all modeled as *nodes*, while an edge can be created whenever a user likes a particular item or an item is described by a particular attribute (*e.g.*, genre, directory, etc.).

Based on this intuition, several approaches exploiting a graph-based representation have been proposed in literature. Generally speaking, these approaches typically fall into the class of *hybrid* recommender systems, since different entities are modeled in the same graph. In a nutshell, the approaches presented in the area of graph-based recommendations can be roughly split into two classes: *(i)* approaches that exploit spreading activation techniques; *(ii)* approaches inspired by PageRank (PR) and random walk [46].

The use of spreading activation for recommendations purposes is investigated from the early 2000s [47] and is still adopted [48, 49] thanks to the good predictive accuracy it provides. As for the use of PR and random walk, one of the early work in the area is due to Hotho et al. [50], who used PR for tag recommendation [51]. Similar intuitions were proposed in other domains as well [52, 53]. Recently, hybrid approaches combining graph-based representations and deep learning also emerged [54, 55].

However, in this work we only focused on PR and Personalized PageRank (PPR) run over the simple graph-based data model, without any other processing and without the application of any other algorithm. This choice is motivated by the findings emerging from previous research [56], where it is shown that recommendation strategies based on PPR can provide state-of-the-art recommendation accuracy.

4. Experimental Protocol

In the current work we follow the protocol presented in [13]. In particular, we focus on the fairness of recommendations with respect to users' expectations. In other terms, we aim to *analyze to what extent the items in the recommendation lists follow the distribution of the actual interests of users* with respect to how many popular items they expect to see in the recommended list.

Datasets. To carry out the experiments, we exploited two state-of-the-art datasets which are commonly used to evaluate RS performance. In particular, we used MovieLens-1M, focusing on movie recommendations, and GoodBooks, focusing on book recommendations. Statistics of the datasets are provided in Table 1. As shown in the table, GoodBooks contains more ratings and it is more unbalanced towards positive opinions, but it is more sparse as well (*i.e.*, a higher amount of non-voted items).

Algorithms. As recommendation algorithms we exploited some available implementations of collaborative filtering, content-based and graph-based techniques. As for CF, we used the im-

	MovieLens-1M	GoodBooks
Users	6,040	53,424
Items	3,883	10,000
Ratings	1,000,209	6,000,000
%Positive	57.51%	68.97%
Sparsity	96.42%	99.82%

Table 1
Statistics of the datasets

plementations available in LensKit[2] of user-to-user CF, item-to-item CF and matrix factorization techniques such as FunkSVD and Implicit MF. As for CBRS, we used the implementations available in Gensim[3] of basic TF-IDF recommender system as well as some implementation of the embedding-based methods Word2Vec, Doc2Vec and LSI. Finally, as for PageRank, we exploited NetworkX library[4] that included an implementation of both PageRank (PR) and Personalized PageRank (PPR). For all the algorithms default parameters were used. In particular, as for CF algorithm the number of neighbors is set to 100, while the latent factors of MF algortithms are set to 50. As for PR and PPR, we used 0.85 as damping factor. As future work, we will perform further experiments with different parameter settings for the algorithms.

Data Models. As for CF algorithms, no particular processing was needed since all the available ratings were used to build the user-item matrix or to learn the factorization models. As for CBRS, to feed content-based recommendation algorithms, we used tags, structured descriptive attributes of the items (*i.e.*, actor, director, author, genre, etc.) as well as unstructured features obtained by processing textual content (*i.e.*, description of the book and plot of the movie) through natural language processing libraries. When embedding methods such as Word2Vec are used, we exploited pre-trained embeddings. Finally, as for PR and PPR, structured properties were used as attributes of the items and encoded in the graph.

Evaluation Metrics. Metrics were calculated on the top-10 recommendation list returned by each algorithm for each user, and finally averaged over all the users. As evaluation metrics, we adopted standard methods used to evaluate the fairness of the algorithms. In particular, we adopted: *(i)* catalogue coverage; *(ii)* Gini Index; *(iii)* ΔGAP.

In the following, we briefly introduce the different metrics:

1. *Catalogue Coverage* measures the amount of items in the catalogue which are recommended to at least one user, and it is obtained by merging all the recommendation lists produced for all the users by an algorithm and by counting the amount of different items contained in the merged list. Of course, the higher the coverage, the higher the *fairness of the algorithm.*, since a larger number of the items available in the catalogue are included in the recommendation lists.
2. *Gini Index* measures how unbalanced (in terms of frequency) is the distribution of the recommendations to all the users. This metric assumes values in the range [0,1], where 0 indicates a balanced (and more *fair*) distribution of the recommendations, while 1

[2] https://lenskit.org/
[3] https://radimrehurek.com/gensim/
[4] https://networkx.org/

represents the worst value (not balanced recommendations), i.e. recommendations concentrated on a single item.

3. The Group Average Popularity (GAP) measures the average popularity of the items in a certain group. In our case, we define $GAP(g)_p$, which measures the average popularity of the items in the user profiles p of a specific group g and $GAP(g)_r$, which measures the average popularity of the items in the recommendation list r of a specific group g. Popularity is calculated as the amount of ratings expressed by the users on a particular item. Based on the protocol presented in [13], three different groups of users are defined: *blockbuster* buster (whether they majority of the items liked by the user are in the top-20% most rated items), *niche* users (majority of liked items in the less-20% most rated items) and *diverse* users (the remaining).
   For each algorithm and user group, we are interested in the change in GAP (*i.e.*, ΔGAP), which shows how the popularity of the recommended items differs from the expected popularity of the items in the user profiles. Formally:
   $$\Delta GAP(g) = \frac{GAP(g)_r - GAP(g)_p}{GAP(g)_p} \tag{1}$$
   The interpretation of such metric is straightforward. $\Delta GAP = 0$ would indicate fair recommendations in terms of item popularity, where fair means that the average popularity of the recommendations a user receives matches the average popularity in the user's profile. Conversely, if ΔGAP is higher than 0, the algorithm *overestimates* the popularity *required* by the user, based on her previous likes. Conversely, if ΔGAP is lower than 0 an underestimation occurs.

5. Results

In this section we present the results of our experiments and we comment the findings emerging for each evaluation metric and for each dataset.

5.1. Catalogue Coverage and Gini Index

Results concerning the evaluation of *catalogue coverage* are presented in Table 2. Beyond the recommendation paradigms we previously introduced, we also evaluate two *baseline* recommendation algorithms, *i.e.*, random algorithm and popularity-based algorithm. The first provides each user with a set of randomly generated recommendations, while the second one provides all the users with a set of items randomly picked among the most popular ones. In our setting, they represent the *upper* and the *lower* bounds of our experiment, since random algorithm provides with the maximum coverage of the catalogue, while a popularity-based algorithm, by definition, is the one that is mostly affected by popularity bias.

As shown in the Table 2, different outcomes emerged for the different datasets. As for *MovieLens 1M*, Biased MF emerged as the technique that is able to better cover the whole catalogue of items, while Word2Vec emerged as best technique on *GoodBooks*. These results can be explained in light of the different characteristics of the datasets. As shown in Table 1,

		MovieLens 1M		*Goodbooks*	
Paradigm	*Technique*	Catalogue Cov.	Coverage %	Catalogue Cov.	Coverage %
Baseline	Random	3,688	94.98%	10,000	100%
	Popular	67	1.63%	43	0.43%
CF	User-to-User CF	296	7.62%	2,210	22.10%
	Item-to-Item-CF	471	12.13%	**2,819**	**28.19%**
	Biased MF	**547**	**14.09%**	1,830	18.30%
	FunkSVD	276	7.11%	546	5.46%
CBRS	TF-IDF	444	11.43%	2,622	26.22%
	Word2Vec	**492**	**12.67%**	**3,081**	**30.81%**
	Doc2Vec	476	12.26%	2,987	29.87%
	LSI	443	11.41%	2,799	27.99%
Graphs	PR	15	0.38%	11	0.11%
	PPR	**36**	**0.92%**	**22**	**0.22%**

Table 2
Results of the experiments concerning *Catalogue Coverage*. The best-performing technique for each paradigm is emphasized in **bold**, while the overall best-performing techniques for each dataset is also underlined.

MovieLens has a lower sparsity than GoodBooks, that is to say, a higher percentage of items is known (and rated) by the users. Accordingly, a less sparse matrix leads to a better coverage of the catalogue of items, thus it is not surprising the *collaborative filtering* techniques obtain the best results on MovieLens. Conversely, when the sparsity is higher, CF techniques are not able to cover (recommend) a sufficient portion of the catalogue and content-based methods emerged as more effective and more stable. Indeed, in this case Word2Vec obtained the best overall results.

By also comparing *standard* techniques such as User-to-User CF or TF-IDF content-based recommendations with more *advanced* strategies, it emerges that the adoption of more sophisticated models based on matrix factorization or on semantics-aware word embedding techniques leads to a slight improvement of the *catalogue coverage*. As for content-based techniques, this holds for both the datasets. Indeed, both Word2Vec and Doc2Vec provide a larger coverage w.r.t. standard TF-IDF-based recommendations. As for collaborative filtering, the role of the *sparsity* emerged again, since a higher sparsity (as on *GoodBooks*) leads to a decrease in terms of coverage when matrix factorization techniques are adopted. This means that when most of the ratings are unknown, factorization techniques are not able to learn the relationships between latent features and cover just a little portion of the catalogue.

Finally, an interesting behavior also emerged for graph-based techniques, which emerged as the paradigm that is more prone to *popularity bias*. Indeed, PR recommends just a tiny portion of the catalogue of items on both the datasets, and the adoption of a *personalized* variant as PPR does not significantly improve the overall behavior. To conclude, we can state that this first experiment provided us with interesting findings, since the results showed the importance of adopting more sophisticated techniques based on artificial intelligence as well as the fundamental role of sparsity in the selection of the most effective algorithm.

However, it should be pointed out that the overall catalogue coverage of all the algorithms

Paradigm	*Technique*	*Dataset*	
		MovieLens-1M	*GoodBooks*
Baseline	Random	0.185	0.334
	Popular	0.995	0.998
CF	User-to-User CF	0.989	**0.973**
	Item-to-Item-CF	0.990	0.986
	Biased MF	**0.984**	0.987
	FunkSVD	0.997	0.998
CBRS	TF-IDF	0.990	0.961
	Word2Vec	**0.985**	**0.956**
	Doc2Vec	0.987	0.959
	LSI	0.988	0.958
Graphs	PR	0.996	0.999
	PPR	**0.995**	**0.998**

Table 3

Results of the experiments concerning *Gini Index*. The best-performing technique for each paradigm is emphasized in **bold**, while the overall best-performing techniques for each dataset is also underlined.

is not particularly satisfying, since the best-performing algorithm obtained around 14% on *MovieLens* and around 30% on *GoodBooks*. Accordingly, a huge part of the catalogue is still out of the recommendation lists of the users. These experimental outcomes further strengthen the idea of developing strategies to *mitigate* popularity bias and include a larger number of items of the *long tail* in the recommendation lists.

Next, results concerning the evaluation of *Gini Index* are reported in Table 3. Due to space reasons, we can't provide a thorough discussion of the findings emerged by this evaluation metric. However the outcomes follow those already discussed for *catalogue coverage*, since Word2Vec and Biased MF emerged as best-performing techniques on *GoodBooks* and *MovieLens*, respectively. As we already noted for catalogue coverage, CF techniques tend to perform better when the sparsity is lower, while CBRS appeared as more effective when a lower number of ratings is available. Overall, we note again that all the scores are very close to 1. As we explained in the previous section, this means that recommendation lists are very concentrated on a small portion of (popular) items, thus all the algorithms emerged again as very prone to *popularity bias*. This leaves a lot of room for work to develop novel methods and strategies to mitigate this bias and return more balanced recommendation lists.

5.1.1. Group Average Popularity and ΔGAP.

Finally, Figure 1 and Figure 2 2 show the behavior of the different algorithms in terms of ΔGAP. As previously stated, this value shows to what extent the items in the recommendation lists follow the distribution of the items in the user profile in terms of popularity. Values close to 0 represent the ideal behavior, while higher and lower number represent and over-estimation and an under-estimation of the average popularity.

As shown in the figures, the findings of this analysis mostly follow those previously discussed in terms of *catalogue coverage* and *Gini Index*. As for MovieLens data, Biased MF, which already emerged as the technique able to cover the largest part of the catalogue of items,

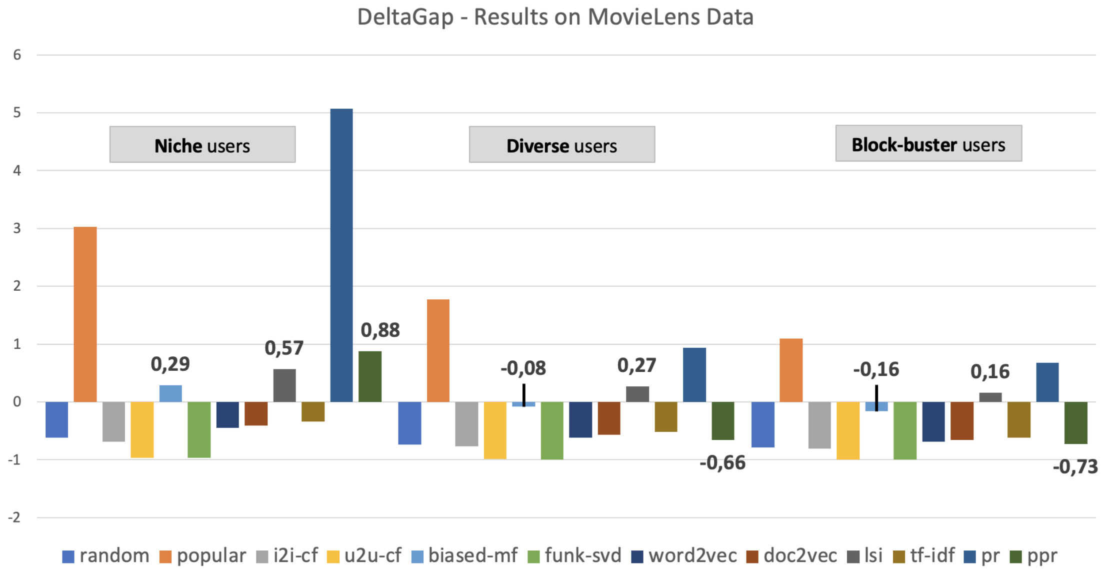

Figure 1: Comparison in terms of ΔGAP on MovieLens-1M data. To improve the readability, the score obtained by the best-performing technique for each paradigm is explicitly reported in the plot.

obtained the overall best results on all the different categories of users (*i.e.*, niche, diverse and blockbusters). As for content-based methods, in this case the overall best results are obtained by LSI, which slightly overcame the basic TF-IDF on all the groups. Finally, as already noted for the previous analyses, graph-based algorithms (in particular in their non-personalized variant) do not perform well, since they are not able to return a list of recommendations that reflects the average popularity of the items in the profile of the user. Overall, we can state that we obtained consisted findings w.r.t. those we previously presented, since the lower sparsity of the data allows collaborative algorithms to generate recommendations that reflect the interests of the users.

As for the general behavior of all the algorithms, it should be pointed out that all the strategies provide a slight *under-estimation* of the average popularity, that is to say, recommended items are less popular than those the user liked. Generally speaking, this is an encouraging behavior, since it is likely that less popular items are included in the recommendation lists. Of course, algorithms that are particularly prone to *popularity bias* (*i.e.*, popularity-based algorithms and PageRank) do not follow this trend, since their recommendations over-estimate the average popularity required by the user.

As for *GoodBooks* data, the overall best results are obtained by content-based recommendations exploiting Word2Vec. This reflects again the behavior we already noted in terms of Gini Index and catalogue coverage. In this case, characterized by a higher sparsity of the data, content-based techniques obtained better results w.r.t CF counterparts, on average. Moreover, differently to what expected, FunkSVD and PPR, that do not perform particularly well on the previous analysis, showed their ability to return a recommendation list in terms of ΔGAP.

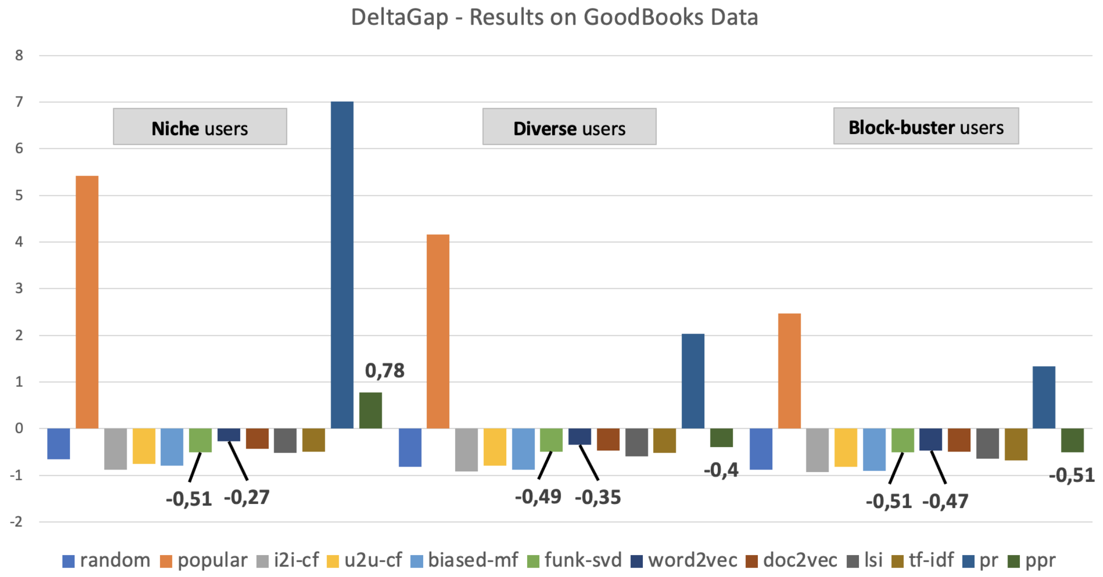

Figure 2: Comparison in terms of ΔGAP on GoodBooks data. To improve the readability, the score obtained by the best-performing technique for each paradigm is explicitly reported in the plot.

However, CBRS based on more advanced representations, such as Word2Vec and Doc2Vec, still beat other algorithms on these data.

6. Conclusions

In this paper, we presented the results of an empirical evaluation investigating how *recommendation algorithms* are affected by *popularity bias*. We considered two state-of-the-art datasets for *movie* and *book* recommendations and several implementations of the three principal recommendation paradigms, i.e., *collaborative filtering, content-based filtering and graph-based algorithms*. We used well-known metrics such as *Catalogue Coverage*, *Gini Index* and *Group Average Popularity* (ΔGAP) in order to discuss how different recommendation techniques are affected by popularity bias.

As shown in the paper, all the algorithms are *strongly* affected by popularity bias, since just a small portion of the available items is included in the recommendation lists. This is a common behavior that does not depend on the particular paradigm which is used to generate recommendations. Accordingly, this work confirms the need for novel and more effective strategies to mitigate *popularity bias*. As for the adherence of the items in the recommendation lists to those in the user profiles in terms of average popularity, it emerged that content-based techniques are more suitable when the sparsity of the data is higher, while collaborative filtering obtained better results with less sparse data. Finally, graph-based techniques did not perform particularly well in any of the experimental settings discussed in this work.

As future work, we will extend this analysis by also considering novel approaches based on deep learning techniques (e.g., complex architectures [57, 58], pre-trained embedding such as BERT [59], etc.) and based on different groups of features (e.g., Linked Open Data, as in [60]), in order to further validate the behavior of the different paradigms.

Acknowledgments

We would like to thank Gianluca Messinese for the huge implementation effort in the development of the framework. Many thanks also to Francesco Cilardi, Martina Pisani, Antonio Valletta and Marco Zaccheo for running part of the experiments that are included in this work.

References

[1] D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, Recommender systems: an introduction, Cambridge University Press, 2010.

[2] P. Resnick, H. R. Varian, Recommender systems, Communications of the ACM 40 (1997) 56–58.

[3] D. Lee, K. Hosanagar, Impact of recommender systems on sales volume and diversity (2014).

[4] P. Castells, N. J. Hurley, S. Vargas, Novelty and diversity in recommender systems, in: Recommender systems handbook, Springer, 2015, pp. 881–918.

[5] M. de Gemmis, P. Lops, G. Semeraro, C. Musto, An investigation on the serendipity problem in Recommender Systems, Information Processing and Management 51 (2015) 695 – 717. URL: http://www.sciencedirect.com/science/article/pii/S0306457315000837. doi:http://dx.doi.org/10.1016/j.ipm.2015.06.008.

[6] D. Kotkov, S. Wang, J. Veijalainen, A survey of serendipity in recommender systems, Knowledge-Based Systems 111 (2016) 180–192.

[7] M. Kaminskas, D. Bridge, Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems, ACM Transactions on Interactive Intelligent Systems (TiiS) 7 (2016) 1–42.

[8] P. Lops, F. Narducci, C. Musto, M. de Gemmis, M. Polignano, G. Semeraro, Recommendations biases and beyond-accuracy objectives in collaborative filtering, in: S. Berkovsky, I. Cantador, D. Tikk (Eds.), Collaborative Recommendations - Algorithms, Practical Challenges and Applications, WorldScientific, 2018, pp. 329–368.

[9] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM Computing Surveys (CSUR) 54 (2021) 1–35.

[10] P. Gajane, M. Pechenizkiy, On formalizing fairness in prediction with machine learning, arXiv preprint arXiv:1710.03184 (2017).

[11] R. Burke, Multisided fairness for recommendation, arXiv preprint arXiv:1707.00093 (2017).

[12] H. Steck, Item popularity and recommendation accuracy, in: B. Mobasher, R. D. Burke, D. Jannach, G. Adomavicius (Eds.), Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011, ACM, 2011, pp. 125–132. doi:10.1145/2043932.2043957.

[13] H. Abdollahpouri, M. Mansoury, R. Burke, B. Mobasher, The unfairness of popularity bias in recommendation, arXiv preprint arXiv:1907.13286 (2019).

[14] C. Anderson, The long tail, Nieuw Amsterdam, 2013.

[15] G. K. Zipf, The Psychobiology of Language, Houghton-Mifflin, 1935.

[16] G. K. Zipf, Human Behavior and the Principle of Least Effort, Addison-Wesley, 1949.

[17] D. Jannach, L. Lerche, I. Kamehkhosh, M. Jugovac, What recommenders recommend: an analysis of recommendation biases and possible countermeasures, User Modeling and User-Adapted Interaction 25 (2015) 427–491.

[18] D. Fleder, K. Hosanagar, Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity, Management Science 55 (2009) 697–712.

[19] A. Bellogín, P. Castells, I. Cantador, Statistical biases in information retrieval metrics for recommender systems, Inf. Retr. Journal 20 (2017) 606–634. doi:10.1007/s10791-017-9312-z.

[20] P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in: X. Amatriain, M. Torrens, P. Resnick, M. Zanker (Eds.), Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010, ACM, 2010, pp. 39–46. doi:10.1145/1864708.1864721.

[21] H. Abdollahpouri, R. Burke, B. Mobasher, Controlling popularity bias in learning-to-rank recommendation, in: P. Cremonesi, F. Ricci, S. Berkovsky, A. Tuzhilin (Eds.), Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017, ACM, 2017, pp. 42–46. doi:10.1145/3109859.3109912.

[22] Y.-J. Park, A. Tuzhilin, The long tail of recommender systems and how to leverage it, in: Proceedings of the 2008 ACM conference on Recommender systems, 2008, pp. 11–18.

[23] H. Abdollahpouri, M. Mansoury, R. Burke, B. Mobasher, Addressing the multistakeholder impact of popularity bias in recommendation through calibration, arXiv preprint arXiv:2007.12230 (2020).

[24] S. Yao, B. Huang, Beyond parity: Fairness objectives for collaborative filtering, arXiv preprint arXiv:1705.08804 (2017).

[25] Z. Zhu, X. Hu, J. Caverlee, Fairness-aware tensor-based recommendation, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 1153–1162.

[26] W. Liu, R. Burke, Personalizing fairness-aware re-ranking, arXiv preprint arXiv:1809.02921 (2018).

[27] D. Kowald, M. Schedl, E. Lex, The unfairness of popularity bias in music recommendation: a reproducibility study, Advances in Information Retrieval 12036 (2020) 35.

[28] M. D. Ekstrand, J. T. Riedl, J. A. Konstan, Collaborative filtering recommender systems, Now Publishers Inc, 2011.

[29] X. Ning, C. Desrosiers, G. Karypis, A comprehensive survey of neighborhood-based recommendation methods, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer, 2015, pp. 37–76. doi:10.1007/978-1-4899-7637-6_2.

[30] A. Tuzhilin, Y. Koren, J. Bennett, C. Elkan, D. Lemire, Large-scale recommender systems and the netflix prize competition, in: KDD Proceedings, 2008, pp. 1–34.

[31] G. Takács, I. Pilászy, B. Németh, D. Tikk, Scalable collaborative filtering approaches for large recommender systems, The Journal of Machine Learning Research 10 (2009) 623–656.

[32] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (2009) 30–37.

[33] X. Su, T. M. Khoshgoftaar, A survey of collaborative filtering techniques, Advances in artificial intelligence 2009 (2009).

[34] Y. Koren, R. Bell, Advances in collaborative filtering, Recommender systems handbook (2015) 77–118.

[35] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th international conference on world wide web, 2017, pp. 173–182.

[36] P. Lops, C. Musto, F. Narducci, G. Semeraro, Semantics in Adaptive and Personalised Systems, Springer, 2019.

[37] R. Burke, Hybrid web recommender systems, The adaptive web (2007) 377–408.

[38] P. Lops, M. de Gemmis, G. Semeraro, C. Musto, F. Narducci, M. Bux, A semantic content-based recommender system integrating folksonomies for personalized access, in: Web Personalization in Intelligent Environments, Springer, 2009, pp. 27–47.

[39] M. G. Ozsoy, From word embeddings to item recommendation, arXiv preprint arXiv:1601.01356 (2016).

[40] D. Khattar, V. Kumar, M. Gupta, V. Varma, Neural content-collaborative filtering for news recommendation., NewsIR@ ECIR 2079 (2018) 45–50.

[41] M. G. Ozsoy, Utilizing fasttext for venue recommendation, arXiv preprint arXiv:2005.12982 (2020).

[42] C. Musto, G. Semeraro, P. Lops, M. De Gemmis, F. Narducci, Leveraging social media sources to generate personalized music playlists, in: International Conference on Electronic Commerce and Web Technologies, Springer, 2012, pp. 112–123.

[43] C. Musto, G. Semeraro, P. Lops, M. de Gemmis, Random indexing and negative user preferences for enhancing content-based recommender systems, in: International Conference on Electronic Commerce and Web Technologies, Springer, 2011, pp. 270–281.

[44] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International conference on machine learning, PMLR, 2014, pp. 1188–1196.

[45] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[46] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: bringing order to the web. (1999).

[47] A. I. Kovacs, H. Ueno, Recommending in context: A spreading activation model that is independent of the type of recommender system and its contents, in: Proc. 2nd International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces (WPRSIUI 06), Citeseer, 2006.

[48] Z. Bahramian, R. A. Abbaspour, C. Claramunt, A context-aware tourism recommender system based on a spreading activation method, International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 42 (2017).

[49] S. Papneja, K. Sharma, N. Khilwani, Context-aware personalized content recommendation using ontology based spreading activation, International Journal of Information Technology 10 (2018) 133–138.

[50] A. Hotho, R. Jäschke, C. Schmitz, G. Stumme, K.-D. Althoff, Folkrank: A ranking algorithm

for folksonomies, in: LWA, volume 1, 2006, pp. 111–114.

[51] C. Musto, F. Narducci, M. De Gemmis, P. Lops, G. Semeraro, Star: a social tag recommender system, Proceedings of the ECML/PKDD Discovery Challenge (2009) 215–227.

[52] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, M. Aly, Video suggestion and discovery for YouTube: taking Random Walks through the view graph, in: Proceedings of the 17th International Conference on World Wide Web, ACM, 2008, pp. 895–904.

[53] T. Bogers, Movie recommendation using Random Walks over the contextual graph, in: Proc. of the 2nd Intl. Workshop on Context-Aware Recommender Systems, 2010.

[54] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, S. Wang, Learning graph-based poi embedding for location-based recommendation, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 15–24.

[55] X. Wang, X. He, Y. Cao, M. Liu, T.-S. Chua, KGAT: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 950–958.

[56] C. Musto, P. Lops, M. de Gemmis, G. Semeraro, Semantics-aware recommender systems exploiting linked open data and graph-based features, Knowledge-Based Systems 136 (2017) 1–14.

[57] C. Musto, C. Greco, A. Suglia, G. Semeraro, Ask me any rating: A content-based recommender system based on recurrent neural networks., in: IIR, 2016.

[58] C. Musto, T. Franza, G. Semeraro, M. de Gemmis, P. Lops, Deep content-based recommender systems exploiting recurrent neural networks and linked open data, in: Adjunct Publication of the 26th conference on user modeling, adaptation and personalization, 2018, pp. 239–244.

[59] M. Polignano, C. Musto, M. de Gemmis, P. Lops, G. Semeraro, Together is better: Hybrid recommendations combining graph embeddings and contextualized word representations, in: Fifteenth ACM Conference on Recommender Systems, 2021, pp. 187–198.

[60] P. Basile, C. Musto, M. de Gemmis, P. Lops, F. Narducci, G. Semeraro, Aggregation strategies for linked open data-enabled recommender systems, 11th ESWC (2014).