






The Effect of Algorithmic Bias on Recommender Systems for Massive Open Online Courses

Ludovico Boratto¹✉, Gianni Fenu², and Mirko Marras²

¹ Data Science and Big Data Analytics Unit, EURECAT, C/ Bilbao 72, 08005 Barcelona, Spain

ludovico.boratto@acm.org

² Department of Mathematics and Computer Science, University of Cagliari, V. Ospedale 72, 09124 Cagliari, Italy
{[fenu](mailto:fenu@unica.it),[mirko.marras](mailto:mirko.marras@unica.it)}@unica.it

Abstract. Most recommender systems are evaluated on how they accurately predict user ratings. However, individuals use them for more than an anticipation of their preferences. The literature demonstrated that some recommendation algorithms achieve good prediction accuracy, but suffer from popularity bias. Other algorithms generate an item category bias due to unbalanced rating distributions across categories. These effects have been widely analyzed in the context of books, movies, music, and tourism, but contrasting conclusions have been reached so far. In this paper, we explore how recommender systems work in the context of massive open online courses, going beyond prediction accuracy. To this end, we compared existing algorithms and their recommended lists against biases related to course popularity, catalog coverage, and course category popularity. Our study remarks even more the need of better understanding how recommenders react against bias in diverse contexts.

Keywords: Recommendation · Algorithmic bias · Learning Analytics

1 Introduction

Recommender systems are reshaping online and online interactions. They learn behavioural patterns from data to support both individuals [44] and groups [22] at filtering the overwhelming alternatives our daily life offers. However, the biases in historical data might propagate in the items suggested to the users, leading to potentially undesired behavior [28]. Therefore, it is important to investigate how various biases are modelled by recommenders and affect their results [40].

Offline experiments on historical data are predominant in the field [25]. However, they often compute prediction accuracy measures that give no evidence on biased situations hidden in the recommended lists [6]. The literature is therefore going one step beyond predictive accuracy. For instance, some recommenders focus on a tiny catalog part composed by popular items, leading to popularity

bias [30, 39, 48]. Others generate a category-wise bias because the rating distribution greatly varies across categories [26]. Historical patterns can promote social biases, such as gender discrimination in publishing [19]. In addition, prediction accuracy might not correlate to online success [14]. Recent movie and book recommenders make good rating predictions, but focus on few popular items and lack in personalization [3]. In contrast, in tourism, higher prediction accuracy corresponds to better perceived recommendations [13]. In view of these context-dependent results, inspired by [31] and recent algorithmic bias studies, assessing how recommenders manage bias in unexplored contexts becomes crucial.

Online education represents an emerging interesting field for this kind of investigation. Large-scale e-learning platforms offering Massive Open Online Courses (MOOCs) have attracted lots of participants and the interaction within them has generated a vast amount of learning-related data. Their collection, processing and analysis have promoted a significant growth of Learning Analytics [46] and have opened up new opportunities for supporting and assessing educational experiences [15, 49]. The market size on this field is expected to grow from USD 2.6 billion in 2018 to USD 7.1 billion by 2023 [38]. These data-driven approaches are being viewed as a potential cure for current educational needs, such as personalization and recommendation [34]. Existing techniques mainly suggest digital educational material (e.g., slides or video-lectures) by leveraging collaborative and content-based filtering [17], while large-scale approaches for online course recommendation have been recently introduced in academia [33] and industry (e.g., Course Talk [2] and Class Central [1]). As these technologies promise to play a relevant role in personalized e-learning, the chance of introducing bias increases and any ignored bias will possibly affect a huge number of people [45]. Entirely removing any bias from algorithms is currently impracticable, but uncovering and mitigating them should be a core objective. In the e-learning recommendation context, this means putting more emphasis on the effects the algorithms have on learners rather than on prediction accuracy [20].

In this paper, we study how recommenders work in the context of MOOCs. We conducted an offline evaluation of different recommendation strategies, which took as input the ratings left by learners after attending MOOCs. We compared the courses recommended by classic and recent methods against data biases, by assessing: (i) how the effectiveness varies when considering algorithms that optimize the rating prediction or the items' ranking, (ii) how course popularity and coverage and concentration biases affect the results, and (iii) how popularity bias in the course categories evaluated by the learners propagates in the recommended lists. These biases might have educational implications (e.g., course popularity bias might affect knowledge diversification, while course category popularity bias might limit learner's multi-disciplinary knowledge). Our results provide evidence on the need to go beyond prediction accuracy, even in the MOOC context.

The rest of this paper is structured as follows. Section 2 presents the dataset and the recommenders. Section 3 evaluates the prediction and the ranking accuracy of the algorithms. Section 4 uncovers some biases and Sect. 5 discusses their

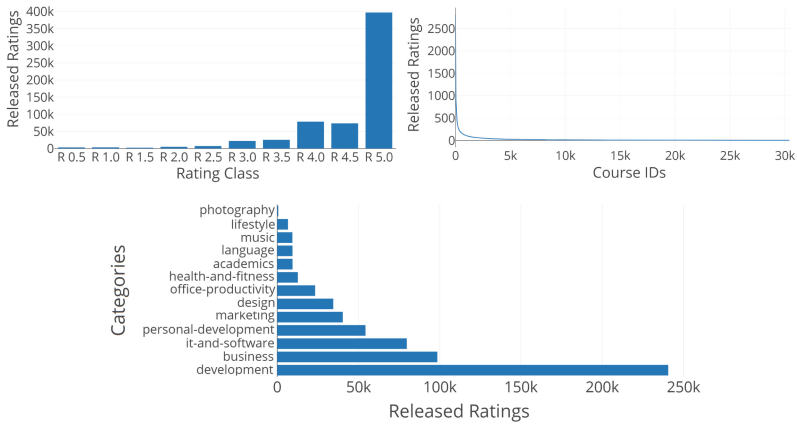


Fig. 1. Sample distributions highlighting bias on the COCO dataset. Ratings per class (top-left). Ratings per course (top-right). Ratings per category (bottom).

impact and their relation with previous studies. Finally, Sect. 6 concludes the paper. The code accompanying this paper is made publicly available¹.

2 Experimental Setup

In our experiments, we leveraged the Java recommendation framework *LibRec* [27] to evaluate several collaborative filtering algorithms on a large-scale online course dataset. Both the dataset and the algorithms are described as follows.

2.1 Dataset

To the best of our knowledge, only one dataset contains both the target MOOCs context and the data size to assess recommendation bias. COCO [16] includes information from one of the most popular course marketplaces for online learning at scale. This public dataset includes 43K courses, distributed into a taxonomy of 15 first-level categories. Over 4M learners provided 6M 5-star ratings and 2M textual reviews. To maintain the evaluation computationally tractable, we took only learners who released at least 10 ratings. The re-sampled dataset includes 37K users, who gave 600K ratings to 30K courses. Figure 1 shows in detail the biases in the dataset towards positive rating classes (common also for learning objects [21]), course popularity and course category popularity.

¹ The code accompanying this paper can be downloaded at <http://bit.ly/2AEban5>.

2.2 Algorithms

We focused on collaborative filtering due to its popularity also in e-learning contexts [9, 17]. We ranged from K-Nearest-Neighbours (KNN) to Learning-to-Rank (LTR) approaches. It should be noted that while we can generate a ranking of the items a user has not evaluated yet by predicting missing ratings, LTR methods are optimized to maximize the ranking quality, generating diverse recommendations against prediction-based algorithms. The algorithms are described below.

Non-Personalized (NP) baselines:

- *Random*: randomly recommending items;
- *MostPop*: recommending the most frequently-consumed items;
- *ItemAvg*: recommending the items with the highest average rating;
- *UserAvg*: recommending the items with the highest user average rating.

Standard Collaborative Filtering (SCF) algorithms:

- *ItemKNN*: item-based collaborative filter (Cosine, K-NN, $k = 100$);
- *UserKNN*: user-based collaborative filter (Cosine, K-NN, $k = 100$).

Matrix Factorization (MF) methods:

- *SVD++*: gradient descent matrix factorization (*LatentFactors* = 40) [36];
- *WRMF*: weighted regular matrix factorization (*LatentFactors* = 40) [29].

Learning-To-Rank (LTR) algorithms:

- *AoBPR*: a variant of BPR manipulating uniform sampling pairs [42];
- *BPR*: bayesian personalized ranking technique for implicit feedback [43];
- *Hybrid*: hybrid integrating diversity and accuracy-focused approaches [50];
- *LDA*: a filtering approach leveraging Latent Dirichlet Allocation [24].

The algorithms were selected as a representative sample as done in other related studies [31]. In what follows, each method is identified by its short name.

3 Comparing Prediction and Ranking Effectiveness

First, we evaluate the recommendation effectiveness, considering metrics that evaluate rating prediction accuracy against those that measure the ranking quality. Like in similar studies [18], we employed a 5-fold cross validation based on a user-sampling strategy. We split the users in five test sets. Each set was the test set of a given fold. In each fold, for each user in the corresponding test set, we selected 5 ratings to be the test ratings, while the rest of their ratings and all the ratings from users not in that test set were the train ratings. Each algorithm was run in both rating prediction and top-10 item ranking mode. We chose top-10 recommendations since they probably get the most attention and 10 is a widely employed cut-off [44]. Root Mean Squared Error (RMSE) evaluated the accuracy of the rating predictions (i.e., the lower the better). Area Under the Curve (AUC), precision, recall, and Normalized Discounter Cumulative Gain (NDCG) [32] measured the recommended list accuracy (i.e., the higher the better).

Table 1. The accuracy of the algorithms on rating prediction (RMSE) and top-10 ranking (AUC, Precision, Recall, NDCG). The results are sorted by increasing RMSE.

Family	Method	RMSE	AUC	Prec@10	Rec@10	NDCG
MF	SVD++	0.68	0.50	0.005	0.001	0.008
NP	UserAvg	0.70	0.50	0.004	0.007	0.005
SCF	UserKNN	0.71	0.68	0.050	0.101	0.095
SCF	ItemKNN	0.76	0.69	0.051	0.102	0.092
NP	ItemAvg	0.78	0.50	0.005	0.008	0.005
NP	MostPop	1.07	0.60	0.023	0.046	0.038
LTR	BPR	2.08	0.69	0.054	0.109	0.094
LTR	AoBPR	2.34	0.69	0.054	0.108	0.094
NP	Random	2.36	0.50	0.004	0.008	0.005
LTR	LDA	4.11	0.66	0.042	0.085	0.074
LTR	Hybrid	4.11	0.55	0.018	0.037	0.029
MF	WRMF	4.12	0.71	0.062	0.124	0.114

Table 1 shows the results. The best ones are printed in bold in case they were significantly different from all others. In this paper, we used paired two-tailed Student’s t-tests with a $p = 0.05$ significance level. The MF approach SVD++ significantly outperformed all the other schemes. However, the rather simple non-personalized UserAvg yielded comparable accuracy to SVD++ and was better than other computationally expensive schemes like ItemKNN and BPR. The latter was significantly better than the other LTR approaches. ItemAvg, which simply considers an item’s average rating, achieved results in line with ItemKNN. The WRMF method performed, somewhat surprisingly, worse than a lot of the traditional ones. The ranking of the algorithms on RMSE is not consistent with respect to other contexts [31]. This confirms that the dataset characteristics like size, sparsity, and rating distributions can greatly affect the recommendation accuracy [5]. The results on item ranking led to a completely different algorithm ranking. BPR and AoBPR achieved the best performance together with WRMF and KNN. Except Hybrid, the LTR methods performed consistently better than MostPop. In line with the results in [35] for learning object recommendation, MostPop performed quite poorly, probably due to the wide range of categories included in the dataset. Although Item-KNN is rather simple, it performed much better than almost all the NP baselines and reached results comparable to LTR schemes. SVD++ led to mediocre results, while it was the best method in rating prediction. In contrast, WRMF achieved the highest accuracy in this setup.

While the accuracy of some algorithms is almost equal, the top-10 lists greatly varied. In view of these differences, we calculated the average overlap of courses recommended by each pair of algorithms to the same user (Fig. 2). The overlap is low, except for (WRMF, UserKNN), (UserKNN, LDA), and (AoBPR, BPR),

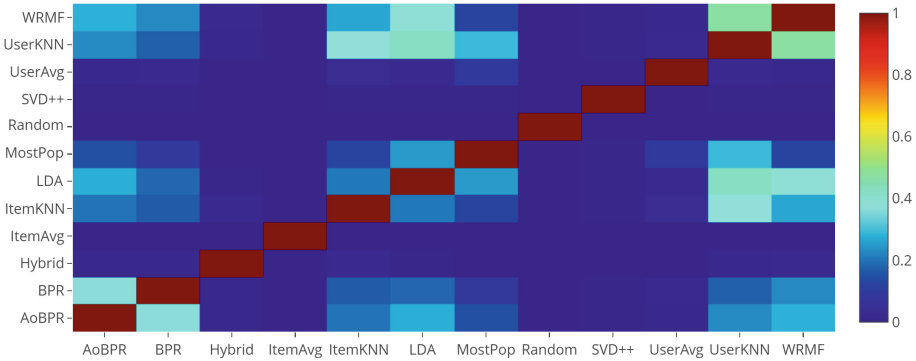


Fig. 2. The average overlap per user between the top-10 lists recommended by each pair of algorithms. The hotter the color of the rectangle the higher the overlap.

where the agreement was above 35%. Hybrid and SVD++ recommended courses which are not typically proposed by the other algorithms, for example. Since MostPop works well and has some similar recommendations with respect to other algorithms, it is possible that they also tend to recommend popular courses.

4 Uncovering Bias in Course Recommendation Rankings

This section includes the experimental comparison of the algorithms and their recommended lists against different bias causes: course popularity and the related coverage and concentration, and course category popularity.

4.1 Interacting with Course Popularity Bias

Even though it is often assumed that recommending what is popular helps high-quality content emerge, popularity can bias future success without reflecting that hidden quality. First, there could be social influence among learners and a lack of independence. Second, engagement and popularity metrics could be subjected to manipulation by fake reviews or social bots. Third, the cost of learning how to evaluate quality could lead to courses with boundless popularity irrespective of differences in quality. Fourth, long-tail courses could be often desirable for more personalized recommendations and knowledge diversification among learners, and are important for generating a better understanding of learners’ preferences. Moreover, long-tail recommendation can drive markets and social good. Suffering from popularity bias could impede novel courses from rising to the top and the market could be dominated by a few large institutions or well-known teachers. With this in mind, we explored how the course popularity in data influences the algorithms. We evaluated how popular are the courses provided by an algorithm, in order to assess its capability to suggest relevant but not popular ones.

Table 2. The popularity of the recommended items based on the average rating and the average number of ratings. The algorithms are sorted by decreasing average rating.

Family	Algorithm	Avg./Std. Dev. rating	Avg./Std. Dev. number of ratings
MF	SVD++	4.76/0.21	134/267
NP	MostPop	4.71/0.07	1545/588
NP	ItemAvg	4.70/0.42	15/3
MF	WRMF	4.68/0.17	404/393
LTR	LDA	4.64/0.14	586/515
SCF	UserKNN	4.63/0.21	192/296
NP	UserAvg	4.60/0.20	341/524
LTR	AoBPR	4.58/0.25	71/152
SCF	ItemKNN	4.55/0.23	88/168
LTR	BPR	4.55/0.27	67/144
NP	Random	4.47/0.58	20/73
LTR	Hybrid	4.44/0.72	11/57

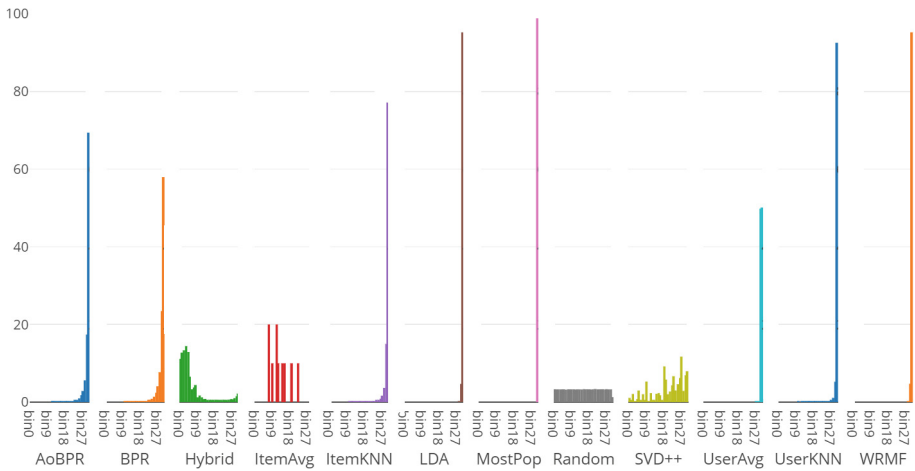


Fig. 3. The distribution of the recommended courses with respect to all the courses in the catalog grouped into 31 bins with 1000 courses each. X-axis shows the bins ranked by increasing popularity in the dataset. Y-axis shows the percentage of the recommended courses belonging to each bin.

Table 2 presents the popularity of the recommended courses as the number of ratings they received. MostPop has, by design, the highest average popularity, since it recommends best sellers. The recommended courses received about 1,500 ratings on average. LDA and WRMF also showed a popularity bias, with

586 and 404 ratings per recommended course, respectively. On the other hand, some algorithms are not biased towards course popularity. SVD++, ItemKNN, AoBPR, and BPR recommended a lot of courses from the long tail. Interestingly, only Hybrid recommended niche and unpopular courses, and its average number of ratings (11) is lower than the average number of ratings per course in the catalog (20). NP baselines achieved a good trade-off between popular and less popular courses. To obtain a detailed picture, we sorted the courses according to the number of ratings in the dataset and organized them in bins of 1000 courses (Fig. 3); the first bin contains the least rated courses, while subsequent ones consider courses of increasing popularity. Except Hybrid, Random, and SVD++, all the algorithms often recommended courses from the bin of the most popular ones (*bin30*). In BPR, course popularity seems to be directly related with the chance of being recommended. SVD++ and Hybrid seem to be good options to recommend niche courses. Interestingly, Hybrid tends to recommend more unpopular courses than popular ones. For ItemAvg, the plot is a rough indicator, since its histogram is based on a small number of recommended courses.

Receiving a lot of ratings does not imply people liked a course. The correlation between number of ratings and average rating is weak, 0.11. Therefore, we measured the average rating of a course as another popularity indicator. It does not tell if the course is really liked by a large number of people, but it can help to see if some algorithms tend to concentrate on highly-rated and probably less-known courses. Table 2 shows that a lot of algorithms recommend courses that were rated, on average, above 4.44 (the global average is 4.47). Furthermore, some algorithms (i.e., SVD++, PopRank, and WRMF) recommended a lot of courses with a high average rating, and low-rated courses are rarely recommended. LDA focuses on high-rated courses (4.64) and is significantly different from other LTR methods. For algorithms not optimized for rating prediction, the average rating is comparably low and closer to the global average. This means that they do not take the average rating into account and recommended also low-rated courses. These algorithms might recommend controversial courses. The average rating of the MostPop recommendations is 4.71, so well-known courses are also top-rated.

4.2 Exploring Bias on Catalog Coverage and Concentration

To check if the recommender system is guiding users to long-tail or niche courses, we should count how many courses in the catalog are recommended. Hence, we looked at the course space coverage and concentration effects of the algorithms.

We counted the number of different courses appearing in the lists (Table 3). The results show that the coverage can be quite different across the algorithms. Except Random, only Hybrid recommend more courses than all other techniques, almost half of the whole catalog. This is in line with the idea behind Hybrid: balancing diversity and rating prediction accuracy. However, in our context, we found it achieved good diversity, but low prediction accuracy. Other LTR approaches provided a coverage of around 20%, except LDA (1%). KNN methods showed a limited catalog coverage, confirming the results in [47] for learning

Table 3. The catalog coverage per algorithm out of 30.399 courses. GINI indexes are computed for the ratings per course distributions in the recommended lists.

Family	Algorithm	Coverage	Catalog percentage	Gini index
NP	Random	30399	100.00	0.16
LTR	Hybrid	12735	41.90	0.77
LTR	BPR	6514	21.43	0.85
LTR	AoBPR	5857	19.27	0.89
SCF	ItemKNN	4653	15.31	0.89
SCF	UserKNN	1183	3.89	0.89
MF	SVD++	1121	3.68	0.88
MF	WRMF	457	1.50	0.68
LTR	LDA	200	0.65	0.64
NP	MostPop	29	0.09	0.63
NP	UserAvg	14	0.04	0.17
NP	ItemAvg	12	0.04	0.28

objects. In contrast to the learning object scenario [37], the algorithms performing best on prediction accuracy are not the best ones also for the catalog coverage. These differences went unnoticed if only the accuracy was considered.

Catalog coverage does not reveal how often each course was recommended. Thus, we captured inequalities with respect to how frequently the courses appeared. For each course suggested by an algorithm, we counted how often it is contained in the lists of that algorithm. The courses are sorted in descending order, according to the times they appeared in the lists, and grouped in bins of 10 courses. *Bin1* contains the most recommended courses. Figure 4 shows the four bins (out of 3040) with the 40 most frequently recommended courses. The Y-axis shows the percentage of recommendations the algorithm has given for the courses in the corresponding bin with respect to the total number of suggestions provided by that algorithm. While SVD++ and ItemKNN recommended a number of different courses, most of them were rarely proposed. BPR, AoBPR, and WRMF, which had a good catalog coverage, provided about 20% of the courses from the 40 most often recommended ones. In Table 3, we show the Gini index to observe the inequality with respect to how often certain courses are recommended, where 0 means equal distribution and 1 corresponds to maximal inequality [25]. Except for the NP baselines, Hybrid and BPR have the weakest concentration bias. Compared to BPR, Hybrid’s Gini index is significantly lower, showing a more balanced distribution of recommendations among courses.

4.3 Exposing Course Category Popularity Bias

E-learning recommender systems are often equipped with a taxonomy that associates each course with one or more categories. This attribute does not imply the

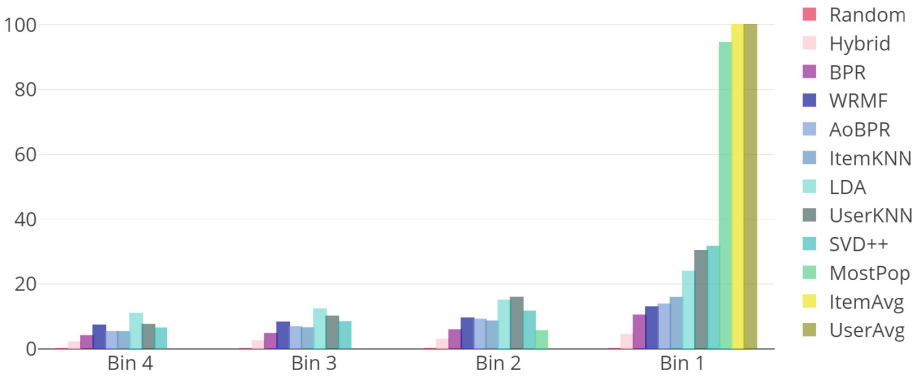


Fig. 4. The distribution of the number of recommendations for the 40 most recommended courses for each algorithm, grouped into 4 bins with 10 courses each. Each coloured column in X-axis is associated to an algorithm. For each algorithm within a bin, Y-axis shows the percentage of recommendations for the courses in the corresponding bin with respect to the total recommendations provided by that algorithm.

quality of a course, but the distribution of the number of ratings can greatly vary across categories. Nonetheless it is natural, given by the heterogeneity of users and courses, it makes aggregated ratings commonly used by algorithms incomparable across categories and thus prone to bias issues. The course category popularity bias inherits large part of the drawbacks held by global popularity bias, and could even influence how learners perceive the recommendations as useful for deepening the knowledge in a preferred category or for fostering a multi-disciplinary knowledge in unexplored categories. Therefore, we focused on the popularity of the category to which courses belong and how the popularity bias affecting course categories in data propagates in the recommended lists.

We counted how many different course categories appeared in the lists. UserAvg exhibited only 3 out of 13 different categories, while MostPop and ItemAvg recommended 5 and 8 categories, respectively. Except for LDA (10 categories), all the other algorithms provided a full coverage on categories. To obtain a clear picture, we sorted the 13 categories according to their increasing number of ratings in the dataset. *Bin12* represents the most popular category. For each algorithm, we counted how many recommendations per category were provided in the recommended lists. Figure 5 shows the distribution of the recommendations per category. BPR, ItemKNN, LDA, and WRMF showed a bias to the most popular category. More than 50% of their recommendations came from it. Hybrid and SVD++ offered a more uniform distribution across categories.

In this context, it was also important to measure how much each algorithm reinforces or reduces the bias to a given category. Figure 6 shows the bias related to course category popularity. Each rectangle shows the increment/decrement on the recommended courses per category with respect to the ratings per category in the dataset. Considering that “development” is the most popular category in COCO, when producing recommendations, MostPop reinforces its popularity by 50%.

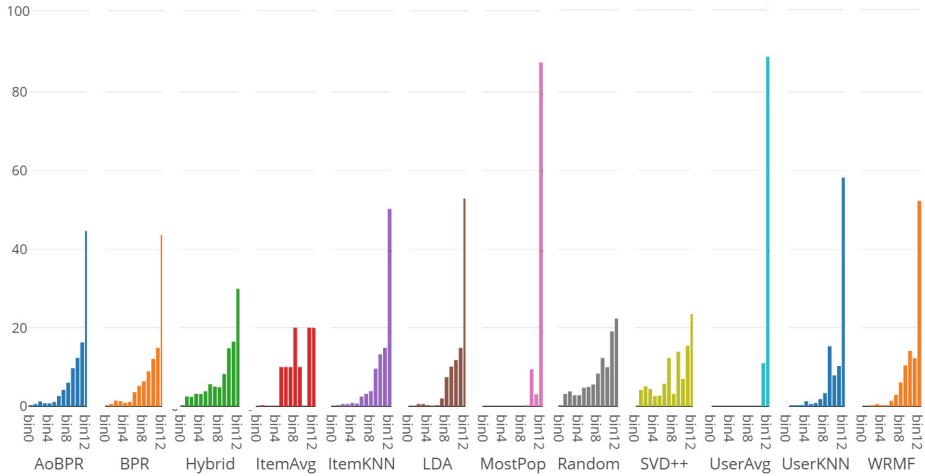


Fig. 5. The distribution of the recommended courses with respect to the course categories. X-axis shows the category bins ranked by their increasing popularity in the dataset. Y-axis shows the relative frequency of the recommended items for each bin.

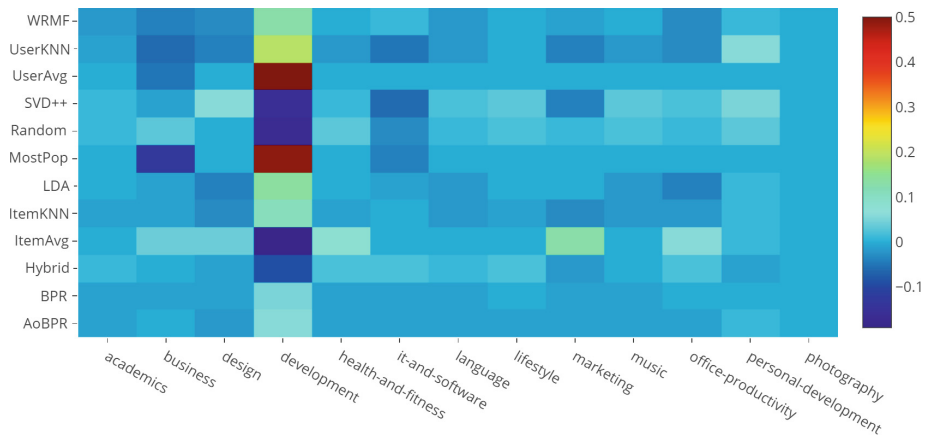


Fig. 6. The reinforcement produced by algorithms with respect to course categories. The hotter the color the higher the reinforcement of that algorithm to that category.

On the other hand, Hybrid and SVD++ caused a 10% popularity reduction in courses of this category. Hence, their recommendations can potentially meet the needs of those not interested only in “development” courses.

5 Discussion

The differences in terms of RMSE or NDCG between some algorithms are very small. For instance, the best-performing techniques, SVD++ and UserAvg, have a difference of 0.02 in RMSE and the same happens for ItemKNN and ItemAvg or LDA, Hybrid, and WRMF. The algorithm ranking based on their item ranking accuracy was quite different, going in contrast with the observations made by [31] for movies (i.e., SVD++ was the best algorithm for both prediction and ranking). However, the analysis regarding catalog coverage and concentration showed that the differences between the algorithms can be more marked and contrasting with respect to the ones reached for prediction accuracy. If the goal is to point learners to different areas of the course catalog, the choice should not be based on accuracy alone. In fact, Hybrid did not perform well on prediction accuracy, but covered half of the catalog and is less influenced by concentration bias. In contrast, SVD++ had a better prediction accuracy, but recommended only 4% of the courses. AoBPR, BPR, and ItemKNN significantly outperformed SVD++ in catalog coverage, even though they achieved a poor accuracy on rating prediction. KNN methods tended to reinforce the bias towards popular courses, as shown by [10] for music. Considering course categories, Hybrid achieved the best trade-off between catalog coverage and category distribution in the recommended lists. While SVD++ demonstrated a low catalog coverage, the recommended courses were more uniformly distributed among categories w.r.t. BPR, AoBPR, and ItemKNN. The latter suggested lots of courses from the most popular category. This went unnoticed if we only considered the catalog coverage. Overall, no algorithm was better than the others, but we observed that Hybrid and SVD++ reached the best trade-off across all the dimensions.

The recommender system community has long been interested in this social dimension of recommendation; similarly to us, representative studies that highlight algorithmic bias analyzed accuracy, catalog coverage, concentration, and popularity bias on several algorithms in the contexts of movies, music, books, social network, hotels, games, and research articles [4, 12, 30, 31, 41]. However, some of the algorithms they analyzed showed a different behavior with respect to the one the same algorithms showed in our context. Category-wise biases have been studied on movies data [26]. Differently from them, we went in-depth on the distribution of the recommended courses with respect to the course categories and highlighted the reinforcement generated by the algorithms on popular categories. Conversely, other works analyzed fairness on users' attributes, such as gender on books and gender with age on movies and music [18, 19]. Popularity and diversity biases at user profile level have been recently considered [11].

6 Conclusions and Future Work

In this paper, we analyzed existing recommendation algorithms in terms of their predictive accuracy on course ratings and rankings in the context of MOOCs. Then, through a series of experiments, we demonstrated that, despite comparably minor differences with respect to accuracy, the algorithms can be quite

different on which courses they recommend. Moreover, they can exhibit possibly undesired biases and consequent educational implications. Offline analysis cannot replace user studies, but our work can provide a better understanding on how generalizable state-of-the-art recommenders are to new contexts, in our case to MOOCs. Furthermore, it can foster more learner-oriented evaluations of the recommenders applied to MOOCs, going beyond classical prediction accuracy.

In next steps, we plan to investigate more algorithms, such as content-based recommenders, and exploit the semantics of content, such as course descriptions or learners' reviews [7, 8, 23]. Moreover, we will consider other types of bias related to demographic attributes and user profiles, as examples. Then, we will design context-specific countermeasures to the biases we have uncovered.

Acknowledgments. Mirko Marras gratefully acknowledges Sardinia Regional Government for the financial support of his PhD scholarship (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2014–2020, Axis III “Education and Training”, TG 10, PoI 10ii, SG 10.5).

This work has been partially supported by the Italian Ministry of Education, University and Research under the programme “Smart Cities and Communities and Social Innovation” during “ILEARN TV, Anytime, Anywhere” Project (DD n.1937 05.06.2014, CUP F74G14000200008 F19G14000910008), and by the Agència per a la Competitivitat de l'Empresa, ACCIÓ, under “AlgoFair” Project.

References

1. Class Central. <https://www.class-central.com/>. Accessed 17 Jan 2019
2. Coursetalk. <https://www.coursetalk.com/>. Accessed 17 Jan 2019
3. Abdollahpouri, H., Burke, R., Mobasher, B.: Controlling popularity bias in learning-to-rank recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 42–46. ACM (2017)
4. Adamopoulos, P., Tuzhilin, A., Mountanos, P.: Measuring the concentration reinforcement bias of recommender systems. *rN (i)* **1**, 2 (2015)
5. Adomavicius, G., Bockstedt, J., Curley, S., Zhang, J.: De-biasing user preference ratings in recommender systems. In: Joint Workshop on Interfaces and Human Decision Making in Recommender Systems, p. 2 (2014)
6. Bellogín, A., Castells, P., Cantador, I.: Statistical biases in information retrieval metrics for recommender systems. *Inf. Retrieval J.* **20**(6), 606–634 (2017)
7. Boratto, L., Carta, S., Fenu, G., Saia, R.: Using neural word embeddings to model user behavior and detect user segments. *Knowl. Based Syst.* **108**, 5–14 (2016)
8. Boratto, L., Carta, S., Fenu, G., Saia, R.: Semantics-aware content-based recommender systems: design and architecture guidelines. *Neurocomputing* **254**, 79–85 (2017)
9. Cechinel, C., Sicilia, M.Á., SáNchez-Alonso, S., GarcíA-Barriocanal, E.: Evaluating collaborative filtering recommendations inside large learning object repositories. *Inf. Process. Manag.* **49**(1), 34–50 (2013)
10. Celma, Ò., Cano, P.: From hits to niches? Or how popular artists can bias music recommendation and discovery. In: Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition, p. 5. ACM (2008)

11. Channamsetty, S., Ekstrand, M.D.: Recommender response to diversity and popularity bias in user profiles. In: Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2017, Marco Island, Florida, USA, 22–24 May 2017, pp. 657–660 (2017). <https://aaai.org/ocs/index.php/FLAIRS/FLAIRS17/paper/view/15524>
12. Collins, A., Tkaczyk, D., Aizawa, A., Beel, J.: Position bias in recommender systems for digital libraries. In: Chowdhury, G., McLeod, J., Gillet, V., Willett, P. (eds.) iConference 2018. LNCS, vol. 10766, pp. 335–344. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78105-1_37
13. Cremonesi, P., Garzotto, F., Turrin, R.: User-centric vs. system-centric evaluation of recommender systems. In: Kotz, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (eds.) INTERACT 2013. LNCS, vol. 8119, pp. 334–351. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40477-1_21
14. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 39–46. ACM (2010)
15. Dessi, D., Fenu, G., Marras, M., Recupero, D.R.: Bridging learning analytics and cognitive computing for big data classification in micro-learning video collections. *Comput. Hum. Behav.* **92**, 468–477 (2018)
16. Dessi, D., Fenu, G., Marras, M., Reforgiato Recupero, D.: COCO: semantic-enriched collection of online courses at scale with experimental use cases. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S. (eds.) WorldCIST’18 2018. AISC, vol. 746, pp. 1386–1396. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-77712-2_133
17. Drachsler, H., Verbert, K., Santos, O.C., Manouselis, N.: Panorama of recommender systems to support learning. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 421–451. Springer, Boston (2015). https://doi.org/10.1007/978-1-4899-7637-6_12
18. Ekstrand, M.D., et al.: All the cool kids, how do they fit in? Popularity and demographic biases in recommender evaluation and effectiveness. In: Conference on Fairness, Accountability and Transparency, pp. 172–186 (2018)
19. Ekstrand, M.D., Tian, M., Kazi, M.R.I., Mehrpouyan, H., Kluver, D.: Exploring author gender in book rating and recommendation. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 242–250. ACM (2018)
20. Erdt, M., Fernández, A., Rensing, C.: Evaluating recommender systems for technology enhanced learning: a quantitative survey. *IEEE Trans. Learn. Technol.* **8**(4), 326–344 (2015)
21. Farzan, R., Brusilovsky, P.: Encouraging user participation in a course recommender system: an impact on user behavior. *Comput. Hum. Behav.* **27**(1), 276–284 (2011)
22. Felfernig, A., Boratto, L., Stettinger, M., Tkalčič, M.: Group Recommender Systems: An Introduction. SECE. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-75067-5>
23. Fenu, G., Nitti, M.: Strategies to carry and forward packets in VANET. In: Cherifi, H., Zain, J.M., El-Qawasmeh, E. (eds.) DICTAP 2011. CCIS, vol. 166, pp. 662–674. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21984-9_54
24. Griffiths, T.: Gibbs sampling in the generative model of latent Dirichlet allocation (2002)
25. Gunawardana, A., Shani, G.: Evaluating recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 265–308. Springer, Boston (2015). https://doi.org/10.1007/978-1-4899-7637-6_8

26. Guo, F., Dunson, D.B.: Uncovering systematic bias in ratings across categories: a Bayesian approach. In: Proceedings of the 9th ACM Conference on Recommender Systems, pp. 317–320. ACM (2015)
27. Guo, G., Zhang, J., Sun, Z., Yorke-Smith, N.: LibRec: a Java library for recommender systems. In: UMAP Workshops, vol. 4 (2015)
28. Hajian, S., Bonchi, F., Castillo, C.: Algorithmic bias: from discrimination discovery to fairness-aware data mining. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2125–2126. ACM (2016)
29. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Eighth IEEE International Conference on Data Mining, ICDM 2008, pp. 263–272. IEEE (2008)
30. Jannach, D., Kamehkhosh, I., Bonnin, G.: Biases in automated music playlist generation: a comparison of next-track recommending techniques. In: Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, pp. 281–285. ACM (2016)
31. Jannach, D., Lerche, L., Kamehkhosh, I., Jugovac, M.: What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Model. User-Adap. Inter.* **25**(5), 427–491 (2015)
32. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst. (TOIS)* **20**(4), 422–446 (2002)
33. Jing, X., Tang, J.: Guess you like: course recommendation in MOOCs. In: Proceedings of the International Conference on Web Intelligence, pp. 783–789. ACM (2017)
34. Klačnja-Milićević, A., Vesin, B., Ivanović, M., Budimac, Z., Jain, L.C.: Recommender systems in E-learning environments. *E-learning Systems. ISRL*, vol. 112, pp. 51–75. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-41163-7_6
35. Kopeinik, S., Kowald, D., Lex, E.: Which algorithms suit which learning environments? A comparative study of recommender systems in TEL. In: Verbert, K., Sharples, M., Klobučar, T. (eds.) *EC-TEL 2016. LNCS*, vol. 9891, pp. 124–138. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45153-4_10
36. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434. ACM (2008)
37. Manouselis, N., Vuorikari, R., Van Assche, F.: Collaborative recommendation of E-learning resources: an experimental investigation. *J. Comput. Assist. Learn.* **26**(4), 227–242 (2010)
38. MarketsandMarkets: Education and learning analytics market report (2018). <https://www.marketsandmarkets.com/Market-Reports/learning-analytics-market-219923528.html>
39. Nagatani, K., Sato, M.: Accurate and diverse recommendation based on users' tendencies toward temporal item popularity (2017)
40. Olteanu, A., Castillo, C., Diaz, F., Kiciman, E.: Social data: biases, methodological pitfalls, and ethical boundaries (2016)
41. Pampin, H.J.C., Jerbi, H., O'Mahony, M.P.: Evaluating the relative performance of collaborative filtering recommender systems. *J. Univ. Comput. Sci.* **21**(13), 1849–1868 (2015)
42. Rendle, S., Freudenthaler, C.: Improving pairwise learning for item recommendation from implicit feedback. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 273–282. ACM (2014)

43. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461. AUAI Press (2009)
44. Ricci, F., Rokach, L., Shapira, B.: Recommender systems: introduction and challenges. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 1–34. Springer, Boston (2015). https://doi.org/10.1007/978-1-4899-7637-6_1
45. Selwyn, N.: Data entry: towards the critical study of digital data and education. *Learn. Media Technol.* **40**(1), 64–82 (2015)
46. Siemens, G., Long, P.: Penetrating the fog: analytics in learning and education. *EDUCAUSE Rev.* **46**(5), 30 (2011)
47. Verbert, K., Drachsler, H., Manouselis, N., Wolpers, M., Vuorikari, R., Duval, E.: Dataset-driven research for improving recommender systems for learning. In: Proceedings of the 1st International Conference on Learning Analytics and Knowledge, pp. 44–53. ACM (2011)
48. Wasilewski, J., Hurley, N.: Are you reaching your audience? Exploring item exposure over consumer segments in recommender systems. In: Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization, pp. 213–217. ACM (2018)
49. Xing, W., Chen, X., Stein, J., Marcinkowski, M.: Temporal predication of dropouts in MOOCs: reaching the low hanging fruit through stacking generalization. *Comput. Hum. Behav.* **58**, 119–129 (2016)
50. Zhou, T., Kuscsik, Z., Liu, J.G., Medo, M., Wakeling, J.R., Zhang, Y.C.: Solving the apparent diversity-accuracy dilemma of recommender systems. *Proc. Nat. Acad. Sci.* **107**(10), 4511–4515 (2010)