What recommenders recommend – An analysis of accuracy, popularity, and sales diversity effects

Dietmar Jannach, Lukas Lerche, Fatih Gedikli, Geoffray Bonnin

TU Dortmund, 44227 Dortmund, Germany
{firstname.lastname}@tu-dortmund.de

Abstract. In academic studies, the evaluation of recommender system (RS) algorithms is often limited to offline experimental designs based on historical data sets and metrics from the fields of Machine Learning or Information Retrieval. In real-world settings, however, other business-oriented metrics such as click-through-rates, customer retention or effects on the sales spectrum might be the true evaluation criteria for RS effectiveness. In this paper, we compare different RS algorithms with respect to their tendency of focusing on certain parts of the product spectrum. Our first analysis on different data sets shows that some algorithms – while able to generate highly accurate predictions – concentrate their top 10 recommendations on a very small fraction of the product catalog or have a strong bias to recommending only relatively popular items than others. We see our work as a further step toward multiple-metric offline evaluation and to help service providers make better-informed decisions when looking for a recommendation strategy that is in line with the overall goals of the recommendation service.

1 Introduction

A recent survey covering 330 papers published in the last five years showed that research in recommender systems (RS) is heavily dominated by offline experimental designs and comparative evaluations based on accuracy metrics [1]. Already some years ago, a too strong focus on accuracy as the only evaluation criterion was identified to be potentially problematic, e.g., in [2]. In recent years, aspects such as novelty, diversity, the popularity-bias of RS as well as potential trade-offs between different quality aspects obtained more attention in research, see, e.g., [3] or [4]. At the same time, laboratory studies and real-world online experiments indicated that higher predictive accuracy does not always correspond to the higher levels of user-perceived quality or to increased sales [5, 6]. In fact, content-based approaches showed to work surprisingly well in these studies and recent others such as [7]. With respect to precision and recall – the most popular accuracy metrics according to [1] – recent work also showed that popularity-based methods can represent a comparably strong baseline ([4], [8]). However, as reported in [6], recommending only popular items does not lead to the desired sales or persuasion effects. In addition, the recommendation of only popular items – or focusing in general on a small set of recommended items –

will naturally lead to an undesired reinforcement of already popular items, thus leading to limited sales diversity and catalog coverage, see e.g., [9] or [10].

In this work we analyze different recommendation algorithms with respect to a number of measures and in particular also with respect to their characteristics in terms of “aggregate” diversity and the concentration on certain items in the sense of [3] and [11]. Our results show that while some algorithms are on a par or comparable with respect to their accuracy, they recommend items from quite different areas of the product spectrum. Furthermore, some highly accurate algorithms tend to focus on a tiny fraction of the item catalog. A simulated experiment finally indicates that some algorithms may lead to an undesired popularity boost of already popular items, which can be in contrast to the potential goal of an RS to promote long-tail items. Overall, we see our work as a further step towards RS evaluation methods that are more focused on their potential utility in multiple dimensions like the utility for the customer or service provider as described in [12]. In order to further support the openness and reproducibility of RS research results in the sense of [13], we make the source code of the evaluation framework used in our experiments available as open source[1].

2 A multi-metric experimental evaluation

In this section, we will first describe our experimental setting which is a multiple-metric evaluation similar to [14]. We will shortly describe the various algorithms included in the measurements and characterize the particular data set for which we will report more details. Then, we present the results of our multiple-metric evaluation beginning with standard accuracy metrics such as the RMSE, Precision and Recall. In order to analyze the characteristics of individual algorithms in a more comprehensive and utility-oriented way, we then use further metrics which will be described in the corresponding sections. In particular, we are interested in the algorithms' capability of recommending long-tail items, their tendency of recommending only popular items as well as possibly resulting concentration effects.

2.1 Algorithms and data sets

Table 1 gives an overview of the algorithms which were evaluated in our study. We chose both popular baselines as well as different types of algorithms from recent research including a learning-to-rank method and a content-based technique, thus covering a broad range of RS approaches. For each data set and algorithm we empirically determined algorithm parameter values that led to high accuracy values. We did however not systematically optimize these values. The algorithms were tested on different data sets, see Section 3, but we will focus on the popular MovieLens data set here. We are aware that the data sets are quite small when compared to the Netflix Prize data set. However, we believe

[1] `http://ls13-www.cs.uni-dortmund.de/homepage/recommender101`

that most observed phenomena reported later on are more related to density and distribution characteristics of the data set than the plain number of available ratings. Furthermore, also in many real-world application scenarios and domains, we are often confronted with a limited number of ratings per user or items.

Table 1. Overview of the compared algorithms.

Non-personalized baselines	
PopRank	Popularity-based ranking.
ItemAvgP	Prediction and ranking based on item average rating.
Simple weighting schemes	
WeightedAvg	Predicts the weighted combination of the active user's average and the target item's average rating. Weight factors for users and items are determined through error minimization.[2]
Rf-Rec	A similar weighting scheme that makes predictions based on rating frequencies [16].
Standard CF algorithms	
Weighted SlopeOne	Recommendation based on rating differences [17].
User-kNN, Item-kNN	Nearest neighbor methods (nb. of neighbors k = 100, similarity threshold = 0, min. nb. of co-rated items = 3).
Funk-SVD	A typical matrix factorization (MF) method (50 factors, 100 initialization rounds) [18].
Koren-MF	Koren's factorized neighborhood model (Item-based approach, 50 factors, 100 initialization rounds, optimization parameters γ and λ were varied across data sets) [15].
Alternative item ranking approaches	
BPR	Bayesian Personalized Ranking [19], a method that learns to rank items based on implicit feedback. Default settings from the MyMediaLite implementation were used. (`http://www.ismll.uni-hildesheim.de/mymedialite/`)
CB-Filtering	A content-based ranking method based on TF-IDF vectors. Items are ranked based on the cosine similarity with the user profile (average vector of all liked items).

Next, we will report the observations made using a data set which is based on a subset of the MovieLens10M data set. As we are interested also in the behavior of computationally-intensive neighborhood-based methods, we randomly sampled about 5,000 active users (at least ten ratings given) and about 1,000 popular items (at least 10 ratings received), ending up with about 400,000 ratings. For these 1,000 movies, we harvested content-descriptions from the IMDb Web site and call this data set MovieLens400k. We have repeated the measurements for a number of other data sets, leading mostly to results which are generally in line with the observations for MovieLens400k. Details will be given later in Section 3.

[2] This method is in some respect similar to the baseline predictor in [15].

2.2 Accuracy results

The accuracy results obtained through a 4-fold cross-validation procedure with 75/25 splits are shown in Table 2. The first three columns – showing the root-mean-square error, as well as precision and recall within the top-10 recommendations – are not particularly surprising and generally in line with findings reported, e.g., for LensKit, in [13]. The differences among CF algorithms in terms of the RMSE are small and larger wins of the MF methods might only be visible for larger data sets, where automatically optimized parameter settings are required. With respect to precision, which was measured by counting only the elements for which a rating was available in the test set (denoted with *TS* in the table), all algorithms managed to place the few items rated with 5 stars at the top and thus outperformed the "most popular" baseline. Additional measurements of the NDCG and Area Under Curve metrics followed the trend of precision and recall. As another side observation, we noted that the unpersonalized ItemAvgP strategy on many data sets including this one or Yahoo!Movies performed very well on the NDCG and ROCAUC and was often nearly on a par with neighborhood-based methods or SlopeOne. We do not report further detailed results here due to space limitations.

Table 2. Accuracy metrics for the MovieLens400k data set.

Algorithm	RMSE	Pre@10(TS)	Rec@10(TS)	Pre@10(All)	Rec@10(All)
Funk-SVD	**0.847**	**0.416**	**0.788**	0.056	0.095
SlopeOne	0.855	0.412	0.782	0.029	0.046
User-kNN	0.856	0.413	0.783	0.035	0.064
Koren-MF	0.861	0.408	0.777	0.028	0.052
Rf-Rec	0.862	0.408	0.777	0.039	0.072
Item-kNN	0.864	0.407	0.776	0.030	0.058
WeightedAverage	0.893	0.407	0.776	0.030	0.058
ItemAvgP	0.925	0.407	0.776	0.027	0.056
BPR	-	0.361	0.716	**0.109**	**0.249**
PopRank	-	0.354	0.709	0.051	0.124
CB-Filtering	-	0.346	0.700	0.021	0.036

However, when we used a different scheme to measure precision and recall by including all items in the test set (denoted with "All") the results are different. The outcome of this measuring scheme for precision and recall depends on the overall number of items in the test set, possibly leading to very small numbers for large catalogs. However, as our data set contains about 1,000 items, the measurement method is in some sense similar to the procedure used in [8] where 1,000 items with unknown ratings were placed in the test set. Now, as also reported in [8], the popularity-based baseline is hard to beat even for MF approaches. In our setting, PopRank for example had a much better recall than Funk-SVD and comparable precision, even though slight improvements for Funk-SVD might be possible by further tweaking algorithm parameters. The comparably simple Rf-

Rec scheme is also ranked higher in the comparison when a different method for measuring is chosen. Overall, however, the best-ranked method on these measures is the "learning-to-rank" approach BPR which outperforms the other techniques by far.

Therefore, as reported in the literature, the ranking of algorithms based on offline experimentation might not only follow different trends when using the RMSE and precision/recall as a metric but can also depend on the particular way a metric is determined. Furthermore, the question which algorithm is the most appropriate, depends on the recommendation scenario (e.g., "find all good items" etc. [20]) so that in one application scenario an algorithm with higher recall might be more favorable than a highly precise one.

As we will see next, the good performance of Rf-Rec and also BPR in the first measurement might be found in their tendency of focusing on popular items.

2.3 Popularity-bias, coverage, and aggregate diversity

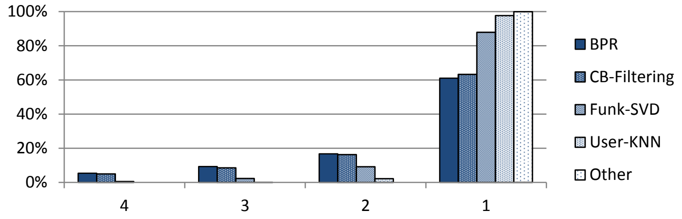
The number of ratings per movie in the MovieLens400k data set as most RS research data sets resembles a typical "Long Tail" distribution. Beside the provision of accurate recommendations, the goal of an RS provider could be to sell more niche items from this long tail. We were therefore interested whether or not the RS algorithms shown in Table 1 behave differently with respect to their ability to recommend products from the whole product catalog. In particular, we measured how many items of the catalog actually ever appear in top-10 recommendation lists. Note that we use the term "(aggregate) diversity" to denote this special form of catalog coverage as done in [3] or [11]. In other works, the diversity of items in recommendation lists with respect to their content features was also identified to be an important factor that can influence the perceived value of a recommender system, see e.g., in [21]. Measuring the level of intra-list diversity was however not in the focus of our current study.

Catalog coverage and aggregate diversity. As a first step of our analysis, we used the evaluation approach described in [11]: We grouped items in bins of 100 elements and sorted them in increasing order based on their actual frequency of appearing in top-10 recommendation lists. Figure 1 shows the first four bins containing the most frequently recommended items by a representative subset of the analyzed algorithms for illustration purposes[3].

We can observe that for many of the strategies, only a tiny fraction of the available items ever appears in top-10 lists. In particular, the Rf-Rec scheme and Koren's neighborhood MF scheme (merged in "Other" in Figure 1) focused on only about 40 different items[4]. On the other end the user-based kNN recommender had a range of 270 items, from which about 100 items (see bin 1

[3] We show 4 out of 10 bins for the 963 items of the MovieLens400k data set. "Other" characterizes the seven remaining algorithms from Table 1, which all concentrate their recommendations on less than 90 items.

[4] In this measurement, all items unseen by a user were part of the test set. Various parameter variations of Koren's method did not lead to different algorithm behavior. One explanation could be a strong effect of the item-bias factor of the learned model.



Gini index	
Funk-SVD	0.71
BPR	0.74
CB-Filtering	0.74
User-KNN	0.95
Other	0.98

Fig. 1. Distribution of actual recommendations (= being in the top-10 of a recommendation list) for the first 400 most recommended items (grouped into 4 bins with 100 items each), as well as the Gini index for the MovieLens400k data set.

in Figure 1) are recommended with a chance of 97.68%. Both Funk-SVD (860 items) and BPR (868) as well as content-based filtering (893) nearly cover the whole item space. The distribution of recommendations nevertheless still has the form of a long-tail, which is however far wider. The top 100 products for Funk-SVD, BPR, and CB-Filtering still accounted for 87.89%, 61.0% and 63.20% of the recommendations.

Figure 1 also shows the corresponding Gini index values for the concentration of the recommendations, see also [11]. Higher values of this index – whose values can be between 0 and 1 – indicate a stronger concentration on a small set of items. The results show that many RS algorithms have a very strong tendency to concentrate on a small product spectrum as indicated in Figure 1. Again, when ranking the algorithms according to the potential business goal of good catalog coverage and long-tail recommendations, a different order is advisable than when only considering accuracy.

Effects of algorithm parameters: diversity and accuracy. Given the strong difference between the two MF methods with respect to the Gini index, we hypothesized that the algorithm parameters will not only influence the accuracy but also the concentration index. Figure 2 shows the effect of varying the number of initial training rounds for the Funk-SVD method.

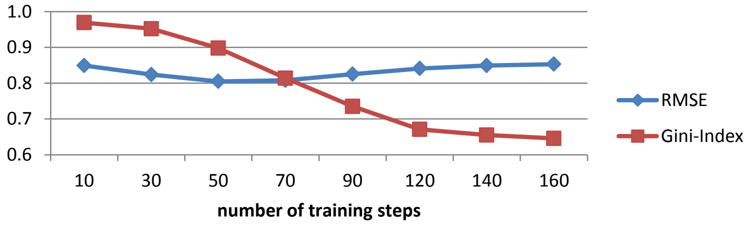


Fig. 2. RMSE and Gini index depending on number of training steps for Funk-SVD.

We can see that increasing the number of rounds has a positive impact on the RMSE and reaches the best values already after about 50 rounds (see also [13] who observed a flattening at about 100 rounds) and then remains stable or slightly increases again, which can be caused by an overfitting tendency. Since the number of training rounds strongly influences the computation time of the method, one might be tempted to set the value to about 50 rounds or even to much smaller values as comparably good RMSE values can be achieved relatively quickly. When considering the development of the Gini index, however, it becomes evident that a much higher number of iterations is required to avoid the possibly undesired algorithm behavior of concentrating on a very small fraction of the catalog. As shown in Figure 2 the curve begins to flatten out after about 120 iterations[5]. Therefore a tradeoff-decision between accuracy, diversity and efficiency might be required.

The Koren-MF algorithm is likewise very dependent on its parametrization. For this particular data set we found a similar dependency for RMSE and Gini index with the step-size parameter γ. Other algorithms parameters however did not influence the two metrics considerably.

Popularity bias of algorithms. After having analyzed how many different items are actually being recommended, another question is whether some algorithms have a tendency to focus on popular items. To assess this algorithm property, we again created top-10 recommendations for each user using different algorithms and measured (I) the popularity of items based on the average item rating, (II) the average popularity of the recommended items in terms of the number of available ratings per item, (III) the distribution of recommendations when the items are organized in bins based on their popularity, again measured using the number of ratings.

Regarding measure (I) we could observe that the average item rating was around 4 for most algorithms. Exceptions were BPR (average = 3.4), CB-Filtering (3.2) and PopRank (3.6), which also recommended movies that were not liked by everyone (but have been seen and rated by many people). On measure (II), Pop-Rank naturally is the "winner" and only recommends blockbusters to everyone (about 1.600 ratings per item). However, BPR is second on this list (940 ratings) while SlopeOne (380) and CB-Filtering (330) form the other end of the spectrum and recommend also long-tail items.

Figure 3 visualizes measure (III)[6], the distribution of recommended items when they are organized in 9 equally sized bins of increasing popularity (based on the number of ratings). We are aware that the figure has to be interpreted carefully as a higher value for some bins can be caused by a very small set of items which are recommended to everyone by some algorithm. Still, we see some general tendencies for the different algorithms, in particular that BPR very often picks items from the bin that contains the most popular items and that popularity of an item correlates strongly with the chance of being recommended. Funk-SVD and CB-Filtering have no strong popularity bias and User-kNN seems to tend to the extremes of the scale; SlopeOne, as expected due to its design, also recommends unpopular items.

[5] We also varied the number of latent features for Funk-SVD but could not observe strongly varying results.

[6] We omit PopRank whose recommendations are all in the "most popular"-bin.

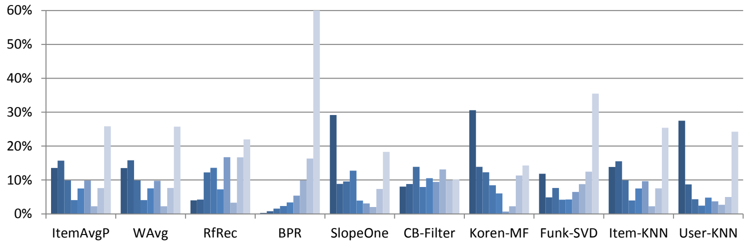


Fig. 3. Distribution of recommendations (= being in the top-10 of a recommendation list) for all items sorted by popularity (number of ratings) and grouped into 9 bins.

Injecting a stronger popularity bias. Our measurements so far as well as observations from the literature suggest that recommending popular items is a very simple strategy to achieve high precision and recall values, at least when all items in the test set are considered. Thus, when using such a measure, it might be quite easy to improve an algorithm's accuracy simply by introducing an artificial popularity bias. While such more biased recommendations might be of little value in practice (see also [8] or the real-world study presented in [6]), researchers might draw wrong conclusions about an algorithm's true value when only relying on precision/recall metrics.

To illustrate the effects of introducing an artificial popularity bias, we conducted an experiment in which we used the popular Funk-SVD algorithm and filtered its recommendations in a way that we only retained items, which were rated by at least k users. For this measurement, we used the publicly available standard MovieLens100k data set.

Table 3. Effects of an artificial popularity bias for precision and recall strategies *All* (all items in the test set) and *TS* (only items with known ratings in the test set).

Algorithm	Pre@10(All)	Rec@10(All)	Pre@10(TS)	Rec@10(TS)
PopRank	0.053	0.098	0.356	0.640
FunkSVD	0.057	0.065	0.415	0.705
FunkSVD, k=100	0.098	0.117	0.416	0.568
FunkSVD, k=200	0.114	0.138	0.384	0.319
FunkSVD, k=300	0.103	0.117	0.314	0.121

Table 3 shows that focusing on popular items can actually increase precision and recall values when a certain measurement method is used. The strategy chosen in the experiment is very simple and leads to poorer results when the threshold value is set too high. Other, more elaborate schemes could however help to even further improve the numbers. When compared with the common measurement method *Precision TS*, we can in contrast see that adding a stronger popularity bias leads to poorer results. Given this observation, the usage of the *Precision/Recall TS* measurement methods might be more appropriate for application domains where a too strong focus on popular items can be risky.

2.4 Popularity reinforcement

Providers of recommendation services on e-commerce platforms are typically interested in the long-term effects of the service on user satisfaction or sales. Unfortunately, measuring such long-term effects is difficult even when it is possible to conduct A/B tests. One of the few works in that direction are the ones by [22] and [23], who observed that the online recommender systems guided customers to a different part of the product spectrum.

The opposite effect, namely that recommenders can even lead to decreased sales diversity and increase the popularity of already popular items was discussed, e.g., by [9]. In order to assess possible effects of different algorithms on the popularity distribution in an offline experimental design, we ran the following simulation on a 200k-rating subset of MovieLens400k[7] to simulate the popularity-enforcing effect of each algorithm over time.

First, we generated a top-10 recommendation list for each user with the algorithm under investigation. To simplify the simulation, we assumed that users only rate items appearing in the recommendation list. We therefore randomly pick one of the recommended items and create an artificial rating for it. This simulated rating is randomly taken according to the overall rating distribution in the data set[8]. Once such an artificial rating was created for each user, all these new ratings are added to the database. This procedure was repeated 50 times to simulate the evolution of the rating database. At each iteration we measured (I) the *concentration of the ratings in the data set* using the Gini index (Figure 4) with bins of 30 products and (II) the number of different products recommended to all users in that run.

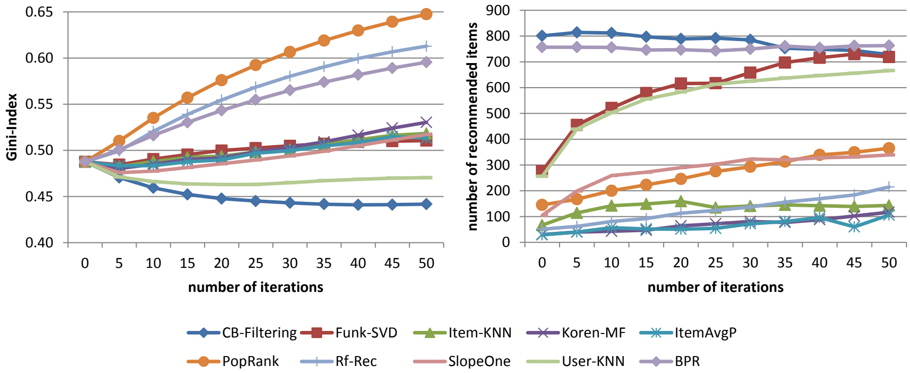


Fig. 4. Simulation results - Gini index.

[7] We reduced the data set size because of the long running-times of the neighborhood-based schemes. The data set characteristics are similar to the larger data set.

[8] Selecting the rating based on the distribution of ratings of an individual item or user would have also been possible.

Figure 4 shows that the effects on the rating distributions strongly vary depending on the chosen algorithm. One class of algorithms – including of course PopRank – leads to a stronger effect in the rating database and already popular items become even more popular and constantly appear in the recommendation lists of even more other users. Both Rf-Rec and BPR fall into this category. For another class of algorithms including the MF approaches and Item-kNN, the concentration index only slowly increases over time. Koren-MF also belongs to this category, which indicates that the recommendation of popular items is only boosted slowly over time. Finally, User-kNN and CB-Filtering initially lead to a stronger diversification of the ratings which then remains stable or increases again.

While the effects are clearly amplified through our specific simulation strategy, we believe that the obtained results indicate that there are significant differences between algorithms, which should be taken into account when looking for an appropriate recommendation strategy.

Looking at measure (II) at the right hand side of Figure 4, the number of distinct items recommended in one iteration per algorithm vary across data sets. Given that there are about 1,000 items in the catalog, CB-Filtering and BPR initially recommend nearly every item to at least one user. Over time, this number slightly decreases. Funk-SVD and User-kNN represent another category of algorithms which start with a comparably high number of recommended items and later on strongly diversify their recommendations[9]. All other algorithms initially recommended a small number of items and only slightly increase the recommendation spectrum over time.

Combining these results with the tendency of some algorithms to concentrate on a small item spectrum, we can observe that both Rf-Rec as well as Koren-MF only recommend very few items (see Figure 4). Koren-MF, however, seems to be able to promote less-popular items resulting in a comparably slow increase of the Gini index in Figure 4. BPR-based recommendations, finally, cover a broad range of items that appear at least once in some recommendation list but in the long run lead to a comparably strong concentration of ratings.

3 Measurements on additional data sets

In order to validate that the observations reported in this paper are not specific for the given characteristics of our MovieLens400k data set, we repeated the experiments on a number of other data sets. In particular, we used the publicly available data sets MovieLens100k, a subset of the BookCrossing ratings and a subset of the Yahoo!Movies rating data set for which we also crawled content information. Furthermore, we tested the algorithms on two further non-public data sets from a telecommunication provider [6] and a data set obtained from the hotel booking platform HRS.com [24]. Except MovieLens100k, all other data

[9] Note that this is not in contradiction with the observations for FunkSVD reported in Figure 4, where we could see an increase of the rating concentration.

sets have a considerably higher sparsity (0.002 to 0.011) than the relatively dense MovieLens400k data set (0.084). Regarding the available ratings MovieLens100k has at least 20 ratings per user and item, whereas for all other data sets we created rating subsets such that the minimum number of ratings was at least 10.

Overall, the general trends reported for the MovieLens400k data set can also be observed for the other data sets. As for RMSE and *Precision/Recall TS*, the matrix factorization (MF) approaches were in most cases only slightly better or on a par with classical kNN schemes, SlopeOne or simple weighting schemes and their advantages might only become visible for larger data sets. With respect to *Precision/Recall All*, the MF methods however outperformed the traditional schemes also for the given small data sets. Considering the parametrization of the algorithms for the data sets, we observed that in particular the results of the Koren-MF method vary strongly depending on the values of the algorithm parameters. These parameters must therefore be carefully tuned in practical settings. Finally, while the accuracy of kNN methods is most of the times comparably good, these techniques often suffer from a limited prediction coverage.

The general trend of a strong concentration on only a small set of items by RF-Rec and the Koren-MF method could also be observed for the other data sets. Similarly, the superior performance of BPR and also PopRank with respect to *Precision/Recall All* and their trend to reinforce the popularity of already popular items was visible across the different data sets.

4 Conclusion

The current practice of evaluating RS based mainly on accuracy metrics is facing various (known) limitations as well as potential methodological problems such as not reported baselines and inconsistently used metrics. Due to these issues, the results of offline analyzes may remain inconclusive or even misleading and the correspondence of such measurements with real-world performance metrics can be unclear. Real-world evaluations and, to some extent, lab studies represent probably the best methods to evaluate systems. Still, we believe that more practically-relevant insights can be achieved also in offline experimental studies, when algorithms are evaluated along several dimensions and on different data sets. In particular, we believe that the analysis of potential trade-offs (e.g., between diversity and accuracy) as done in a growing number of recent papers should be put even more into the focus of future research.

In this paper, we have analyzed known algorithms with respect to the diversity, item popularity and accuracy of the recommendations. Our observations indicate that – depending on their parametrization and the usage scenario – different algorithms lead to different effects and that the choice of the "best" approach to be deployed in a live system should be guided by the consideration of these effects. With the provision of the software used in the experiments, we finally hope to contribute to the reproducibility of RS research.

References

1. Jannach, D., Zanker, M., Ge, M., Gröning, M.: Recommender systems in computer science and information systems - a landscape of research. In: Proceedings 13th International Conference on E-Commerce and Web Technologies (EC-WEB 2012), Vienna (2012) 76–87
2. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: Proceedings of the 2006 Conference on Human Factors in Computing Systems (CHI 2006). (2006) 1097–1101
3. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering **24**(5) (2012) 896–911
4. Steck, H.: Item popularity and recommendation accuracy. In: Proceedings of the 2011 ACM Conference on Recommender Systems, Chicago, Illinois, USA (2011) 125–132
5. Cremonesi, P., Garzotto, F., Negro, S., Papadopoulos, A., Turrin, R.: Looking for "good" recommendations: A comparative evaluation of recommender systems. In: Proceedings of the IFIP Conference on Human-Computer Interaction 2011, Lisbon, Portugal (2011) 152–168
6. Jannach, D., Hegelich, K.: A case study on the effectiveness of recommendations in the mobile internet. In: Proceedings of the 2009 ACM Conference on Recommender Systems, New York (2009) 41–50
7. Kirshenbaum, E., Forman, G., Dugan, M.: A live comparison of methods for personalized article recommendation at Forbes.com. In: Proceedings European Conference on Machine Learning and Knowledge Discovery in Databases - ECML PKDD 2012, Bristol, UK (2012) 51–66
8. Cremonesi, P., Koren, Y., Turrin, R.: algorithms on top-n recommendation tasks. In: Proceedings of the 2010 ACM Conference on Recommender Systems, Barcelona (2010) 39–46
9. Fleder, D., Hosanagar, K.: Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. Management Science **55**(5) (2009) 697–712
10. Prawesh, S., Padmanabhan, B.: The "top N" news recommender: count distortion and manipulation resistance. In: Proceedings of the 2011 ACM Conference on Recommender Systems, Chicago, USA 237–244
11. Zhang, M.: Enhancing the diversity of collaborative filtering recommender systems. PhD Thesis. Univ. College Dublin (2010)
12. Said, A., Tikk, D., Shi, Y.: Recommender Systems Evaluation: A 3D Benchmark. In: ACM RecSys 2012 Workshop on Recommendation Utility Evaluation: Beyond RMSE, Dublin, Ireland (2012) 21–23
13. Ekstrand, M.D., Ludwig, M., Konstan, J.A., Riedl, J.T.: Rethinking the recommender research ecosystem: reproducibility, openness, and LensKit. In: Proceedings of the 2011 ACM Conference on Recommender Systems, Chicago, Illinois, USA (2011) 133–140
14. Meyer, F., Fessant, F., Clérot, F., Gaussier, E.: Toward a new protocol to evaluate recommender systems. In: ACM RecSys 2012 Workshop on Recommendation Utility Evaluation: Beyond RMSE, Dublin, Ireland (2012) 9–14
15. 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, Las Vegas, USA (2008) 426–434

16. Gedikli, F., Bagdat, F., Ge, M., Jannach, D.: RF-REC: Fast and accurate computation of recommendations based on rating frequencies. In: 13th IEEE Conference on Commerce and Enterprise Computing, CEC 2011, Luxembourg (2011) 50–57
17. Lemire, D., Maclachlan, A.: Slope one predictors for online rating-based collaborative filtering. In: SIAM Conference on Data Mining, Newport Beach (2005) 471–480
18. http://sifter.org/~simon/journal/20061211.html (last accessed: 03/2013) (2006)
19. 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, Montreal, Canada (2009) 452–461
20. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems **22**(1) (2004) 5–53
21. Castagnos, S., Jones, N., Pu, P.: Eye-Tracking Product Recommenders' Usage. In: Proceedings of the 2010 ACM Conference on Recommender Systems, Barcelona, Spain (2010) 29–36
22. Dias, M.B., Locher, D., Li, M., El-Deredy, W., Lisboa, P.J.: The value of personalised recommender systems to e-business: A case study. In: Proceedings of the 2008 ACM Conference on Recommender Systems, Lausanne, Switzerland (2008) 291–294
23. Zanker, M., Bricman, M., Gordea, S., Jannach, D., Jessenitschnig, M.: Persuasive online-selling in quality & taste domains. In: Proceedings 7th International Conference on E-Commerce and Web Technologies (EC-Web 2006), Krakow, Poland (2006) 51–60
24. Jannach, D., Karakaya, Z., Gedikli, F.: Accuracy improvements for multi-criteria recommender systems. In: Proceedings of the 13th ACM Conference on Electronic Commerce. EC'12, Valencia, Spain (2012) 674–689