Reenvisioning the comparison between Neural Collaborative Filtering and Matrix Factorization

Vito Walter Anelli*
vitowalter.anelli@poliba.it
Politecnico di Bari, Italy

Alejandro Bellogín
alejandro.bellogin@uam.es
Universidad Autónoma de Madrid, Spain

Tommaso Di Noia
tommaso.dinoia@poliba.it
Politecnico di Bari, Italy

Claudio Pomo*
claudio.pomo@poliba.it
Politecnico di Bari, Italy

ABSTRACT

Collaborative filtering models based on matrix factorization and learned similarities using Artificial Neural Networks (ANNs) have gained significant attention in recent years. This is, in part, because ANNs have demonstrated very good results in a wide variety of recommendation tasks. However, the introduction of ANNs within the recommendation ecosystem has been recently questioned, raising several comparisons in terms of efficiency and effectiveness. One aspect most of these comparisons have in common is their focus on accuracy, neglecting other evaluation dimensions important for the recommendation, such as novelty, diversity, or accounting for biases. In this work, we replicate experiments from three different papers that compare Neural Collaborative Filtering (NCF) and Matrix Factorization (MF), to extend the analysis to other evaluation dimensions. First, our contribution shows that the experiments under analysis are entirely reproducible, and we extend the study including other accuracy metrics and two statistical hypothesis tests. Second, we investigated the Diversity and Novelty of the recommendations, showing that MF provides a better accuracy also on the long tail, although NCF provides a better item coverage and more diversified recommendation lists. Lastly, we discuss the bias effect generated by the tested methods. They show a relatively small bias, but other recommendation baselines, with competitive accuracy performance, consistently show to be less affected by this issue. This is the first work, to the best of our knowledge, where several complementary evaluation dimensions have been explored for an array of state-of-the-art algorithms covering recent adaptations of ANNs and MF. Hence, we aim to show the potential these techniques may have on beyond-accuracy evaluation while analyzing the effect on reproducibility these complementary dimensions may spark. The code to reproduce the experiments is publicly available on GitHub at *https://tny.sh/Reenvisioning*.

*Authors are listed in alphabetical order. Corresponding authors: Vito Walter Anelli (vitowalter.anelli@poliba.it) and Claudio Pomo (claudio.pomo@poliba.it).

RecSys '21, September 27-October 1, 2021, Amsterdam, Netherlands

ACM ISBN 978-1-4503-8458-2/21/09…$15.00
https://doi.org/10.1145/3460231.3475944

CCS CONCEPTS

• **Information systems** → **Recommender systems**; *Collaborative filtering*; • **Computing methodologies** → *Learning from implicit feedback*; Neural networks; Factorization methods.

KEYWORDS

Item Recommendation, Matrix Factorization, Neural Collaborative Filtering

ACM Reference Format:
Vito Walter Anelli, Alejandro Bellogín, Tommaso Di Noia, and Claudio Pomo. 2021. Reenvisioning the comparison between Neural Collaborative Filtering and Matrix Factorization. In *Fifteenth ACM Conference on Recommender Systems (RecSys '21), September 27-October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3460231.3475944

1 INTRODUCTION

Artificial Neural Networks (ANNs) are ubiquitous in many research areas in recent years. Recommender Systems (RS) is a paradigmatic example where these techniques have been applied, in part because they open up possibilities in domains where classical techniques have difficulties in understanding the item content — i.e., video or image recommendation —, but also because they allow to extract complex patterns that, in principle, are not captured by more simple methods [38]. However, recent research challenges how useful these techniques are in the context of RS, and where their advantages really lie [9, 10, 16, 27].

More specifically, these studies have compared ANNs techniques against classical personalized algorithms — mostly matrix factorization or nearest neighbors —, emphasizing the lack of well-tuned baselines or incorrect, incomplete, or even unfair experimental conditions evidenced in the literature. Nonetheless, while these conclusions are useful to move forward on understanding when ANNs should be applied in recommendation, they neglect evaluation dimensions that are important in the RS community, such as diversity, novelty, coverage, and so on [12], since most of the authors have focused, so far, on the precision/accuracy of the recommended items produced by those methods.

In this context, we aim to bridge this gap and compare ANNs against classical RS under several evaluation dimensions. With this goal in mind, we focus on a recent paper [27] where the authors showed how proper hyperparameter selection could make simple operations like a dot product outperform similarity learning through ANNs. We have the following two main goals: first,

replicating the aforementioned paper, since it has a salient characteristic where the authors used in their tables results from other papers (claiming they used comparable evaluation settings and tuning, something that too often is not true as it is difficult to do properly [28]); once we are able to replicate these results, we **reproduce** them under different situations. In particular, we report beyond-accuracy evaluation metrics, to explore the extent these methods behave on complementary dimensions they have not been optimised for, or whose results have not been reported about.

Our main contributions are two-fold: on the one side, we corroborate the results reported recently in [27] where ANNs are outperformed by simple modifications on classical algorithms; moreover, we complement these observations with additional experimental dimensions, showing more accuracy metrics and their corresponding statistical analysis, together with novelty, diversity, and bias measurements, which allow us to provide a more complete overview of the performance of these algorithms when compared against ANNs and, hence, a better understanding of when and how these approaches might be useful.

2 BACKGROUND AND FORMULATION

In this section we formalize, first, the recommendation problem and later review Matrix Factorization and Neural Collaborative Filtering approaches. The notation used herein is summarised as follows. Matrices are denoted by uppercase letters A, vectors by lowercase bold letters $\mathbf{b}$, scalars by lowercase letters a. We denote the concatenation of the vectors $\mathbf{b}$ and $\mathbf{c}$ by $[\mathbf{b}, \mathbf{c}]$. Let there be a pool of users (U) to recommend to and a catalog of items (I) to recommend from. A recommendation algorithm returns a *score* for a given user-item pair that corresponds to the estimated degree of *satisfaction* for the user enjoying that item. In this work we focus on a specific kind of recommendation algorithms, where two d-dimensional embedding vectors, p and q, are combined into a single score. Conventionally, p represents the embedding of a user, q the embedding of an item, and $\phi(p, q)$ ($\phi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$) the similarity of the user to the item.

Matrix Factorization (MF) is a famous and classical example of model-based Collaborative Filtering methods [20]. The algorithm learns a latent representation of items and users, whose linear interactions aim to explain the observed feedback. There are several variations of MF proposed in the literature, and a comprehensive review would deserve a specific study that is out of the scope of this work. However, to provide the reader an intuition of how much the factorization strategy has been disruptive in recent years, we briefly review the works that are, in our humble opinion, the most representative or the ones that can show the myriad of possible applications of factorization models.

The first examples of factorization models were soon recognized as state-of-the-art models. Among these pioneering works, there could be found SVD [20], PureSVD [7], SVD++ [19], PMF [29, 30], NNMF [22], and SLIM [23]. Among the several methods on matrix factorization, Rendle's work has heavily influenced the evolution of the factorization models. In detail, BPR-MF [26] deserves particular attention because it boosted the MF research, and it is still considered as a state-of-the-art model. For completeness, Rendle also proposed Factorization Machines [25] that generalize the factorization approach. The biggest criticism of MF approaches, however, lies in their linearity. To address this concern, a recently popularized trend in the community of recommender systems is using deep neural architectures with deep neural networks that can model the non-linearity in data through nonlinear activation functions. In this respect, Neural Collaborative Filtering [14] and Neural Factorization Machines [13] have been recently proposed to overcome the inability of MF to capture non-linearities. Furthermore, Attentional Factorization Machines [36] use an attention network to learn the importance of feature interactions. Factorization models have been specialized for a variety of tasks such as Active-Learning [40], Context-aware [17], Cross-domain [11], Knowledge-aware [4, 5], and even explainable [39] recommendation.

In particular, Neural Collaborative Filtering [14] is one of the most representative recommendation approaches, which aims to estimate unknown user-item preference scores by exploiting deep neural networks [38]. Since Artificial Neural Networks (ANNs) can approximate any continuous function on a compact set as long as the ANN has enough hidden states [8], He et al. [14] propose to exploit ANNs to learn the affinity between p and q. Let $\Phi(\cdot)$ be the transformation function of the deep neural network defined as $\Phi : \mathbb{R}^{\dim(p)+\dim(q)} \rightarrow \mathbb{R}^d$, He et al. propose to concatenate the two embeddings and predict the score as follows:

$$\psi^{\text{MLP}}(\mathbf{p}, \mathbf{q}) := \Phi([\mathbf{p}, \mathbf{q}]). \tag{1}$$

Additionally, He et al. defines a *generalized* matrix factorization model, in which p and q are combined using element-wise multiplication ($\odot$):

$$\psi^{\text{GMF}}(\mathbf{p}, \mathbf{q}) := \mathbf{p} \odot \mathbf{q}, \tag{2}$$

Finally, He et al. propose a comprehensive model, named NeuMF, that combines the two previous approaches together:

$$\psi^{\text{NeuMF}}(\mathbf{p}, \mathbf{q}) := \psi^{\text{MLP}}(\mathbf{p}, \mathbf{q}) + \psi^{\text{GMF}}(\mathbf{p}', \mathbf{q}'), \tag{3}$$

where the prime symbol ($'$) suggests that those embeddings might have a different size and are, in fact, different from the former ones. Finally, a careful reader may have noticed that ψ has an output dimension of d. This is correct, since He et al. applies a final prediction layer on top of them:

$$\phi^{\text{NCF}} := \sigma(\mathbf{W} \cdot \psi(\mathbf{p}, \mathbf{q})) \tag{4}$$

where σ is an activation function, and W is an additional weight matrix that is learned along with the other model parameters.

More recently, Rendle et al. [27] define the embeddings as model parameters, and the affinity between p and q is modeled by means of a dot product:

$$\phi^{\text{dot}}(\mathbf{p}, \mathbf{q}) := b_g + b_p + b_q + \sum_{f=1}^{d} p_f q_f, \tag{5}$$

where g_b, b_p, and b_q denote the global, user, and item bias, respectively. In this way, [27] presents a direct comparison between ANNs and MF by changing the underlying operation between the embeddings, while keeping everything else comparable.

3 REPLICATION OF PRIOR EXPERIMENTS: SETTINGS AND RESULTS

This section focuses on describing how the replication of the experiments from papers Dacrema et al. [9], He et al. [14], Rendle et al. [27] has been set up. It starts by defining the evaluation protocol applied to compare Neural Collaborative Filtering (NCF) and Matrix Factorization (MF) against the baselines in their respective works.

3.1 Settings

Although this study involves the replication of the results from three different studies, this paper mainly aims to replicate the results from Rendle et al. [27]. In Rendle et al., the authors retrieve the already split datasets from the original NCF repository[1]. Specifically, He et al. [14] provide a split version of *MovieLens-1M* and *Pinterest*. To split these well-known datasets, the authors adopt a temporal leave-one-out policy, moving the last user interaction into the test set. Furthermore, they binarize *MovieLens-1M* to make the two datasets coherent with implicit feedback. Finally, they evaluate the methods on a shortlist of 101 candidate items for each user. This list comprises one relevant item (i.e., the transaction in the test set) and 100 negative items randomly sampled from not consumed items. In He et al. [14] and Rendle et al. [27], the authors evaluate the performance on top-10 recommendation lists computing Hit-Rate (HR) and Normalized Discounted Cumulative Gain (nDCG). The first estimates how many users have the withheld item in the top-10. The second measures the capability of the methods to rank the relevant item. In the following, the formulation of nDCG as presented in Krichene and Rendle [21] is adopted since it is the same one adopted in Rendle et al. [27]. Moreover, He et al. [14] and Rendle et al. [27] select the best models, for each recommendation system, according to HR@10. In this paper, the model selection follows the same strategy.

The present study involved the implementation of seven recommendation methods. MF implementation was designed accordingly to Rendle et al. [27] (also provided as a public repository[2]). Regarding NeuMF, the implementation refers to He et al. [14]. Finally, for the five remaining baselines, the implementation refers to Dacrema et al. [9] since it is the source for some of the results reported in Rendle et al. [27]. More specifically, the five implemented baselines are Slim [23], iALS [15], PureSVD [7], EASE^R [32], and RP$^3\beta$ [24]. According to the investigation provided by the authors [9], we replicate the baseline training exploiting the best hyperparameters found in the additional material[3].

To summarize, this paper replicates seven different recommendation algorithms from three different works: [27], [14], and [9]. The Elliot recommendation framework [3] is adopted as the benchmarking framework. Elliot provides an out-of-the-box recommendation pipeline. The tested models have been implemented as external models to grant complete adherence to the original implementations. All the implemented models and configuration files are publicly available[4] to provide a complete reproducibility environment with an ad-hoc version of Elliot.

3.2 Results

The first set of experiments aims to replicate Table 1 from Rendle et al. [27]. In that table, the authors compare Neural Matrix Factorization (NeuMF) and MF with a shortlist of baselines: Popularity, SLIM, and iALS. In detail, the authors report from Dacrema et al. [9] the results for Popularity, SLIM, NeuMF, and iALS. Instead, MF is trained using the publicly available implementation they provide. Overall, this table questions the prominence of NeuMF and shows the high performance achieved by MF.

Hence, Table 1 replicates and extends the results provided in Table 1 from Rendle et al. [27]. In this study, all the recommendation algorithms have been retrained according to the best hyperparameters provided in Dacrema et al. [9]. Specifically, Table 1 reports HR and nDCG values for *MovieLens-1M* and *Pinterest* datasets, respectively. The careful reader may have noticed that, for each dataset, both replicated and original results are reported. Original results columns are marked with references to the source papers. Dacrema et al. [9] also consider other recommendation algorithms. Interested in a more comprehensive comparison, we have selected EASE^R, RP$^3\beta$, and PureSVD for further replication. Finally, since Dacrema et al. [9] do not consider Matrix Factorization, no confusion arises regarding the origin of the results. Interestingly, Table 1 further confirms the findings of the original experiments showing that MF consistently overcomes the other baselines. It is worth mentioning how well the new experiments approximate the original ones.

Nonetheless, Rendle et al. [27] clearly state, in Table 1, that MF results are reported from (their) Figure 2. That figure compares MF, Learned Similarity (MLP), NeuMF, and pretrained NeuMF, considering different embedding sizes. However, the results reported from Rendle et al. (except for MF) are from He et al. [14]. Therefore, to conduct a thorough replication, we herein replicate some pivotal experiments reported in that figure. In detail, we have decided to replicate six MF experiments (three embedding sizes for each dataset) and eight NeuMF experiments (four embedding sizes for each dataset). For MF, we considered 32, 128, and 192 as embedding sizes. For NeuMF, we considered 24, 48, 96, and 192 as embedding sizes (according to Appendix 3 from Rendle et al. [27]).

Thus, Figure 1 reports the original values from He et al. [14] regarding Learned Similarity (MLP) and pretrained NeuMF and reports our replicated experiments' results for MF and NeuMF. It is noteworthy mentioning that our MF experiments overlap with Rendle et al. showing that the MF curves dominate the others. However, the NeuMF curve shows different behavior from He et al. It is even more interesting to notice that the NeuMF experiment with 48 as the embedding size is also reported by Dacrema et al. [9], and the results are very close to ours.

Reviewing *Pinterest* results, MF confirms to be the best model in terms of HR and nDCG. Even in this context, NeuMF never overcomes the MF models. Actually, NeuMF reaches the best performance in terms of HR and nDCG for the NeuMF model with

[1] https://github.com/hexiangnan/neural_collaborative_filtering
[2] https://github.com/google-research/google-research/tree/master/dot_vs_learned_similarity
[3] https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation/blob/master/DL_Evaluation_TOIS_Additional_material.pdf
[4] https://github.com/sisinflab/Reenvisioning-the-comparison-between-Neural-Collaborative-Filtering-and-Matrix-Factorization

Table 1: Comparison of NeuMF and MF with various baselines with cutoff @10. The table replicates (and compare with) the results from Dacrema et al. [9], Rendle et al. [27]. The best results are highlighted in bold, the second best results is underlined. The columns with the Δ symbol indicate the absolute variation (for each metric) between the values of the experiments reproduced and those reported in the articles by Dacrema et al. [9] and Rendle et al. [27].

Method	*MovieLens-1M*		*MovieLens-1M* [9, 27]		Δ *MovieLens-1M*		*Pinterest*		*Pinterest* [9, 27]		Δ *Pinterest*	
	nDCG	HR	nDCG	HR	nDCG	HR	nDCG	HR	nDCG	HR	nDCG	HR
MostPop	0.2542	0.4535	0.2543	0.4535	$-1 \cdot 10^{-04}$	0	0.1410	0.2743	0.1409	0.2740	$1 \cdot 10^{-04}$	$3 \cdot 10^{-04}$
SLIM	0.4480	0.7164	0.4468	0.7162	$1.2 \cdot 10^{-03}$	$2 \cdot 10^{-04}$	0.5615	0.8696	0.5601	0.8679	$1.4 \cdot 10^{-03}$	$1.7 \cdot 10^{-03}$
iALS	0.4385	0.7123	0.4383	0.7111	$2 \cdot 10^{-04}$	$1.2 \cdot 10^{-03}$	0.5587	0.8766	0.5590	0.8762	$3 \cdot 10^{-04}$	$4 \cdot 10^{-04}$
NeuMF	0.4211	0.6952	0.4349	0.7093	$-1.38 \cdot 10^{-02}$	$-1.41 \cdot 10^{-02}$	0.5480	0.8704	0.5576	0.8777	$-9.6 \cdot 10^{-03}$	$-7.3 \cdot 10^{-03}$
MF	**0.4545**	**0.7310**	**0.4523**	**0.7294**	$2.2 \cdot 10^{-03}$	$1.6 \cdot 10^{-03}$	**0.5776**	**0.8898**	**0.5794**	**0.8895**	$1.8 \cdot 10^{-04}$	$3 \cdot 10^{-04}$
$EASE^R$	0.4494	0.7192	0.4494	0.7192	0	0	0.5605	0.8684	0.5604	0.8684	0	0
$RP^3\beta$	0.4011	0.6758	0.4011	0.6758	0	0	0.5685	0.8796	0.5685	0.8796	0	0
PureSVD	0.4299	0.6926	0.4303	0.6937	$-4 \cdot 10^{-04}$	$-1.1 \cdot 10^{-03}$	0.5233	0.8261	0.5241	0.8268	$-8 \cdot 10^{-04}$	$-7 \cdot 10^{-04}$

a number of factors equal to 16, according to the findings provided by Dacrema et al. [9]. Nonetheless, increasing that number of factors, we witness a performance decrease: both HR and nDCG decrease as the number of factors increases. Furthermore, Table 1 reports the overall results for the methods involved in the investigation. These outcomes confirm the evidence shown by Rendle et al. [27]: also other MF-based methods, like SLIM and iALS, outperform NeuMF. Beyond MF, also $EASE^R$ provides a very notable performance. PureSVD behaves similarly to NeuMF. Finally, $RP^3\beta$ does not appear competitive as the other models in the investigation: its performance is consistently worse than the others. All these findings further confirm the results provided in Dacrema et al. [9]. Another finding (from Dacrema et al. [9]) the careful reader can rediscover in our experiments is the $RP^3\beta$ performance on the *Pinterest* dataset: although MF again demonstrates its higher accuracy, $RP^3\beta$ demonstrates competitive performance overcoming all the remaining baselines. Overall, the general take-home message of Dacrema et al. [9] experiments is confirmed: *NeuMF is often not better than relatively simple and well-known techniques.*

Finally, Table 2 compares, for the sake of completeness, our experiments on NeuMF without pretraining with the same configuration from He et al. [14]. The results in columns marked with the reference are from Table 2 in He et al. [14] and correspond to the results for the NeuMF model without pretraining. As shown before qualitatively, the replicated results obtained through the benchmark framework overlap the original ones. However, considering 64 factors on *Pinterest*, an appreciable difference can be observed that regards the nDCG value. This is probably due to the non-deterministic initialization of the model that leads to slightly different results. The effect seems to be more evident in the models with a greater embedding size, suggesting that the model accumulates the initial uncertainties. Remarkably, the deviation in the results exhibits a different trend (from the original model). Even though this could be a signal of lack of robustness of the model, further investigation is needed to shed light on this behavior.

Table 2: Performance of NeuMF without pre-training. The table compares replicated experiments (on the left) with prior experiments He et al. [14]. Differently from He et al. [14], the results on *Pinterest* show a performance decrease with 64 factors. All metrics are with cutoff @10.

Factors	*MovieLens-1M*		*MovieLens-1M* [14]		*Pinterest*		*Pinterest* [14]	
	nDCG	HR	nDCG	HR	nDCG	HR	nDCG	HR
8 [14] - 24 [27]	0.409	0.688	0.410	0.688	0.547	0.868	0.546	0.869
16 [14] - 48 [27]	0.416	0.691	0.420	0.696	0.548	0.870	0.547	0.871
32 [14] - 96 [27]	0.418	0.699	0.425	0.701	0.541	0.869	0.549	0.870
64 [14] - 192 [27]	0.421	0.695	0.426	0.705	0.536	0.861	0.551	0.872

4 COMPARING ANNS AND MF ON NEW CONTEXTS

From now on, our investigation extends the previously described replicated experiments. These new experiments share the same setup of the previous section and exploit the same benchmark framework. The purpose is to provide a broader view of the experiments considering other evaluation dimensions. First, we extend the list of metrics used to measure the accuracy of generated recommendation lists. Second, we investigate beyond-accuracy evaluation dimensions, covering the novelty and diversity of the recommendations and the bias induced by the recommendation algorithms. All the considered metrics have been implemented in Elliot[5] [3] and are publicly available[6]. The specific nDCG formulation used in this paper is named *nDCGRendle2020* to avoid confusion with the alternative implementation.

4.1 An extended Accuracy evaluation

The existence of a high correlation between the accuracy metrics has been recently shown [33]. Nevertheless, an evaluation

[5] https://github.com/sisinflab/elliot
[6] https://github.com/sisinflab/Reenvisioning-the-comparison-between-Neural-Collaborative-Filtering-and-Matrix-Factorization

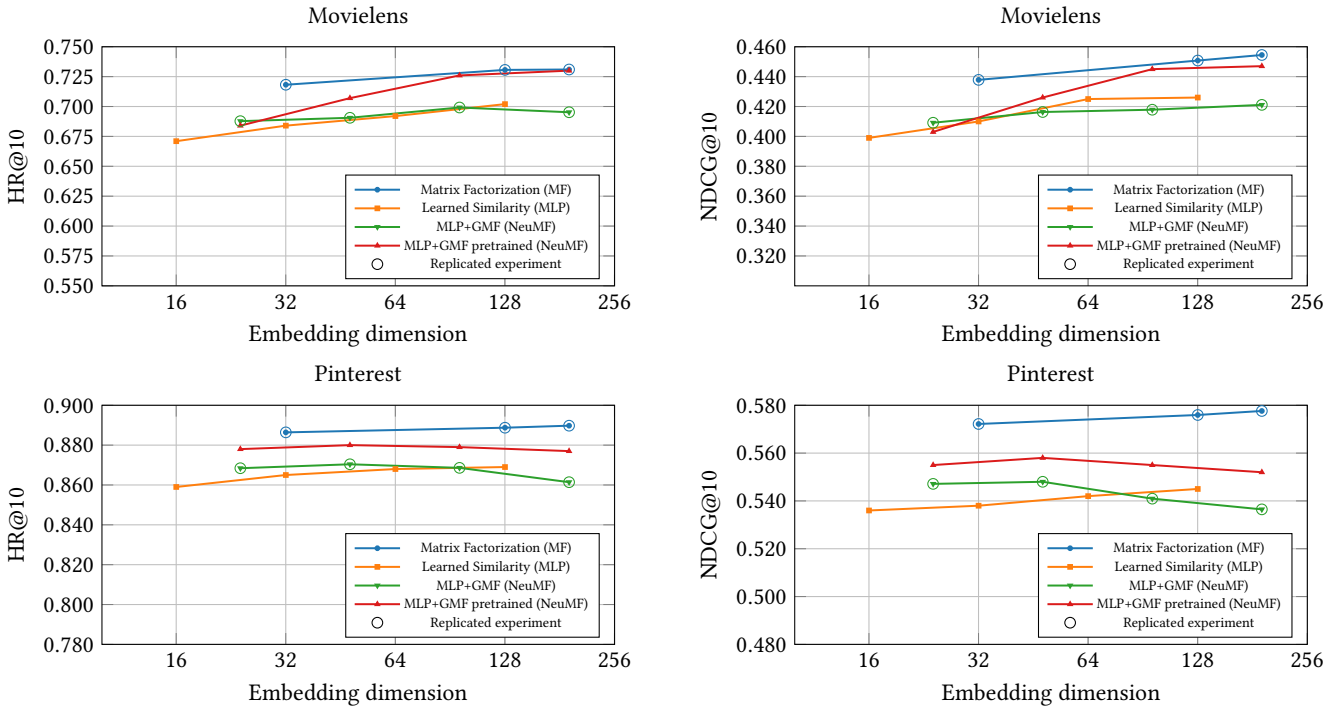

Figure 1: Comparison of learned similarities (MLP, NeuMF) with a dot product: The results for MLP and pretrained NeuMF are from He et al. [14], Rendle et al. [27]. MF substantially outperforms MF, NeuMF, and pretrained NeuMF. Nonetheless, on *MovieLens-1M*, when considering large embeddings, pretrained NeuMF is competitive.

that examines only HR and nDCG could be quite limited. Therefore, we further extend the previous analysis considering other five metrics: F1-measure (F1) [12], Limited Area Under the Curve (LAUC) [31], Mean Average Precision (MAP) [12], Mean Average Recall (MAR) [12], and Mean Reciprocal Rank (MRR) [35].

Table 3 reports the results for the extended accuracy evaluation. Observing the big picture, MF is still one of the most competitive models, consistently being the best model regarding all the considered metrics on *Pinterest*. However, the situation is quite different on *MovieLens-1M*, since EASE^R shows the best performance in terms of MAP, MAR, and MRR. Another interesting confirmation is the RP$^3\beta$ performance on *Pinterest*. For all the considered metrics, it shows to be the second-best model. However, if we observe the outcomes on *MovieLens-1M*, the situation is much more confusing. Previous experiments showed that MF was the most accurate method, followed by EASE^R. Table 3 shows a quite different scenario, with EASE^R being the best model regarding MAP, MAR, and MRR, and the second best concerning the remaining metrics. Conversely, MF still shows competitive results, but regarding MAP and MRR, it is not in the first two places. Overall, MF, Slim, and iALS outperform NeuMF on these two datasets, hence confirming the most important finding of the previous experiment.

4.1.1 Statistical hypothesis tests. To complete the study regarding the accuracy evaluation, we investigated whether the differences between the accuracy results of the various methods are statistically significant. Figure 2 shows eight heatmaps of statistical significance calculated with the Student's paired t-test. Statistically significant differences (with a p-value lower than 0.05) are drawn in green. In contrast, p-values greater than or equal to the 0.05 threshold value are colored by shades of red.

Figure 2 confirms that MF significantly overcomes the other methods regarding nDCG and HR on both *MovieLens-1M* and *Pinterest*. Besides MF, and considering the same metrics, the differences between EASE^R, iALS, and Slim are not always statistically significant. Moreover, when analyzing MAP and MRR, it is noteworthy that the difference between EASE^R, Slim, and MF are not significant. For what concerns NeuMF, the situation is different. Indeed, the differences with EASE^R, PureSVD, and Slim are not always significant. Finally, RP$^3\beta$ deserves a concluding remark since all the differences with the other models are statistically significant, thus confirming its positive performance on *Pinterest* and the below-the-average one on *MovieLens-1M*.

4.2 Novelty and Diversity

Once it is established how accurate the various methods are, our study expands beyond the accuracy evaluation. This section focuses on the ability of the recommendation algorithms to propose unknown items (Novelty), on overall item coverage, and on the ability to suggest highly diversified recommendation lists. For what concerns Novelty, we measure Expected Free Discovery (EFD) [34] and Expected Popularity Complement (EPC) [34], which measure the ability of a recommendation system to recommend items from

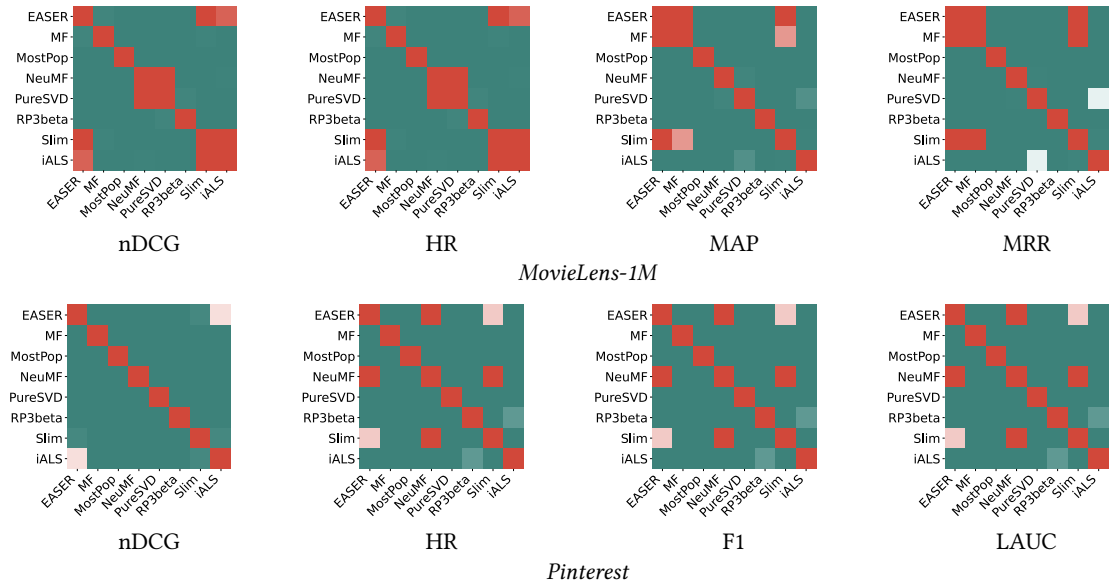

Figure 2: Statistical hypothesis tests using Student's paired t-test with a threshold value (light red) of p=0.05. Algorithm pairs which results are statistically significant are in green, the results that are not statistically significant are in red.

Table 3: Comparison of NeuMF and MF with various baselines on an extended set of accuracy metrics with cutoff @10. The best results are highlighted in bold, the second-best result is underlined.

Method	*MovieLens-1M*					*Pinterest*				
	F1	LAUC	MAP	MAR	MRR	F1	LAUC	MAP	MAR	MRR
MostPop	0.0825	0.4531	0.0647	0.3072	0.1937	0.0499	0.2742	0.0341	0.1717	0.1009
SLIM	0.1303	0.7159	0.1204	0.5372	0.3648	0.1581	0.8694	0.1535	0.6757	0.4649
iALS	0.1295	0.7117	0.1172	0.5288	0.3537	0.1594	0.8764	0.1525	0.6786	0.4587
NeuMF	0.1264	0.6947	0.1120	0.5106	0.3363	0.1583	0.8702	0.1487	0.6653	0.4472
MF	**0.1316**	**0.7232**	0.1188	0.5383	0.3573	**0.1618**	**0.8896**	**0.1584**	**0.6958**	**0.4796**
$EASE^R$	0.1308	0.7187	**0.1210**	**0.54202**	**0.3655**	0.1579	0.8682	0.1532	0.6752	0.4639
$RP^3\beta$	0.1229	0.6753	0.1053	0.4853	0.3166	0.1599	0.8794	0.1554	0.6836	0.4710
PureSVD	0.1259	0.6921	0.1153	0.5178	0.3486	0.1502	0.8259	0.1422	0.6339	0.4286

the long tail. Concerning aggregate diversity metrics, we adopt Item Coverage [12] that measures the overall number of items the recommender suggests to the population. Finally, to measure how diversified the recommendation lists are, we exploit two popular distributional inequality metrics, the Gini Index (Gini) [12] and Shannon Entropy (SE) [12]. The Gini Index is defined as 1 - Gini Index from Gunawardana and Shani [12], so that a higher value corresponds to a greater degree of diversification.

Figure 3 shows twelve bar charts that compare MF and NeuMF with the other baselines regarding the six observed metrics on the two datasets. Let the analysis focus on Novelty. It is worth noticing that, even here, MF outperforms NeuMF and the other baselines since it generates recommendation lists with a larger number of items belonging to the long tail. Conversely, NeuMF shows poor performance, and only $RP^3\beta$ and Most Popular behave worse. In general, also other matrix factorization models such as iALS and SLIM are shown to be competitive against the other baselines under analysis. However, under the perspective of recommendation Diversity, the scenario dramatically changes. In fact, regarding Item Coverage, NeuMF is the best performing model on *MovieLens-1M*, and a very competitive one on *Pinterest* (the best one is $RP^3\beta$), suggesting a higher overall number of items present in the catalog. Conversely, MF (and the other MF-based models) are not able to win the comparison. For what regards recommendation list diversification, the Gini bar chart reveal a more clear ranking of the methods. Again, MF fails to be effective in terms of diversity, and, on *MovieLens-1M*, only $EASE^R$ and $RP^3\beta$ show lower results. NeuMF shines neither on *MovieLens-1M* dataset nor on *Pinterest* dataset. However, in both cases, it shows a greater propensity to generate personalized lists than MF. Interestingly, on both datasets, the iALS model is particularly competitive regarding the two distributional inequality metrics. Finally, even here, the reader may appreciate how different the $RP^3\beta$ performance is on the two datasets.

4.3 Analysis of Recommendation Biases

In the final part of the study, we focus on how the recommendation algorithms induce or amplify bias into the recommendation lists. Indeed, user-item interactions are often distributed unevenly over different groups of users and categories of items. This could be due to various reasons ranging from the naturally varying user preferences to the existence of a recommendation system in the preference collection system. Recommendation algorithms can inherit or even amplify this imbalanced distribution, leading to various kinds of bias. To examine the bias effect we consider five different metrics: Average Coverage of Long Tail items (ACLT) [2], Average

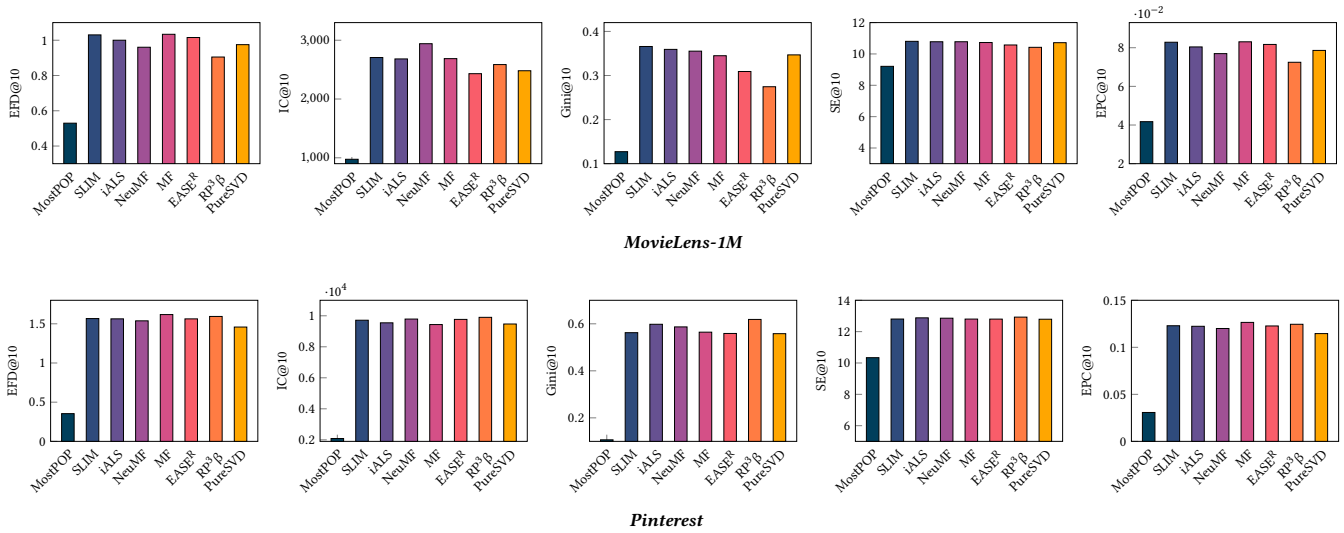

Figure 3: Novelty and Diversity comparison of NeuMF and MF with various baselines (higher is better).

Percentage of Long Tail Items (APLT) [1, 2], Average Recommendation Popularity (ARP) [2, 37], Ranking-based Statistical Parity (RSP) [41], and Ranking-based Equal-Opportunity (REO) [41].

Figure 4 shows ten bar charts that compare MF and NeuMF against the other baselines regarding these five bias measures on the two datasets. The most straightforward metric to analyze is ARP. This metric measures the average popularity of the recommended items in each list. Interestingly, MF and NeuMF behave similarly on *MovieLens-1M*, while EASE^R and RP$^3\beta$ are more prone to suggest popular items. In contrast, the other MF-based methods, iALS, Slim, and PureSVD, show the best performance. However, on *Pinterest*, the ranking is less clear since all the methods behave in a similar way. Let the analysis focus on ACLT and APLT. APLT measures the average percentage of long-tail items in the recommended lists, while ACLT measures how much exposure long-tail items get in the recommendations. These two metrics exhibit three interesting behaviors: (i) both iALS and NeuMF seem to be less prone to these kinds of biases, (ii) MF, EASE^R, and PureSVD show to be heavily affected by them, (iii) the difference of RP$^3\beta$ performance on the datasets influences the bias of the generated recommendations.

Finally, we focus our investigation on RSP and REO. RSP measures whether items in different groups have the same probabilities of being recommended. Poor RSP means one or more groups have lower recommendation probabilities than others. REO measures the bias that items in one or more groups have lower recommendation probabilities given the items enjoyed by users. Differently from RSP, REO-based bias does not depend on sensitive attributes.

In this study, even though additional information could be retrieved to form item groups, the purpose is to conduct the investigation based on the same information available to the original authors. Therefore we formed two distinct groups of items based on the popularity signal. One group comprises the 20% most popular items, while the other includes the remaining items. For this reason, in the following, we refer to them as PopRSP and PopREO.

On *MovieLens-1M* dataset, iALS and SLIM exhibit the best performance regarding both metrics. Even here, NeuMF demonstrates to be less prone than MF to this type of bias. MF does not show unsatisfactory results regarding both statistical parity and equal opportunity, but it never overcomes NeuMF. Finally, EASE^R, RP$^3\beta$, and PureSVD are affected by the bias and under-recommended items from minority groups, even though these items are present in the user history. In contrast, on *Pinterest*, RP$^3\beta$ shows leading performance, along with iALS and NeuMF. As detailed in Section 3.1, following He et al. [14] and Rendle et al. [27], all the recommenders were optimized for accuracy. It is left as future work an extended analysis where the effect of different optimization goals could have on the recommendation accuracy and beyond-accuracy dimensions, as in Kaminskas and Bridge [18].

5 CONCLUSION

Understanding how the different recommendation algorithms work under unique evaluation dimensions is critical to advance the field. In this work, we aimed to shed some light on this aspect, by contrasting recent models that are competitive against Neural Network approaches under complementary dimensions — not only accuracy, but novelty, diversity, coverage, and bias. In particular, we focus on the methods presented in He et al. [14], Rendle et al. [27], and complemented our experimental exploration with the extensive analysis done in Dacrema et al. [9]. We have been able to replicate most of the results reported in those papers, where NeuMF is outperformed by the MF variation presented in Rendle et al. [27]. Moreover, when reproducing these approaches in new contexts, such as other evaluation dimensions or more accuracy metrics, baselines like EASE^R and RP$^3\beta$ are confirmed as solid candidates to be included in any comparison in the future, as their performance in terms of accuracy, diversity, and novelty is sometimes better than those of neural network approaches. However, it is important to highlight that the trend obtained for NeuMF is slightly different

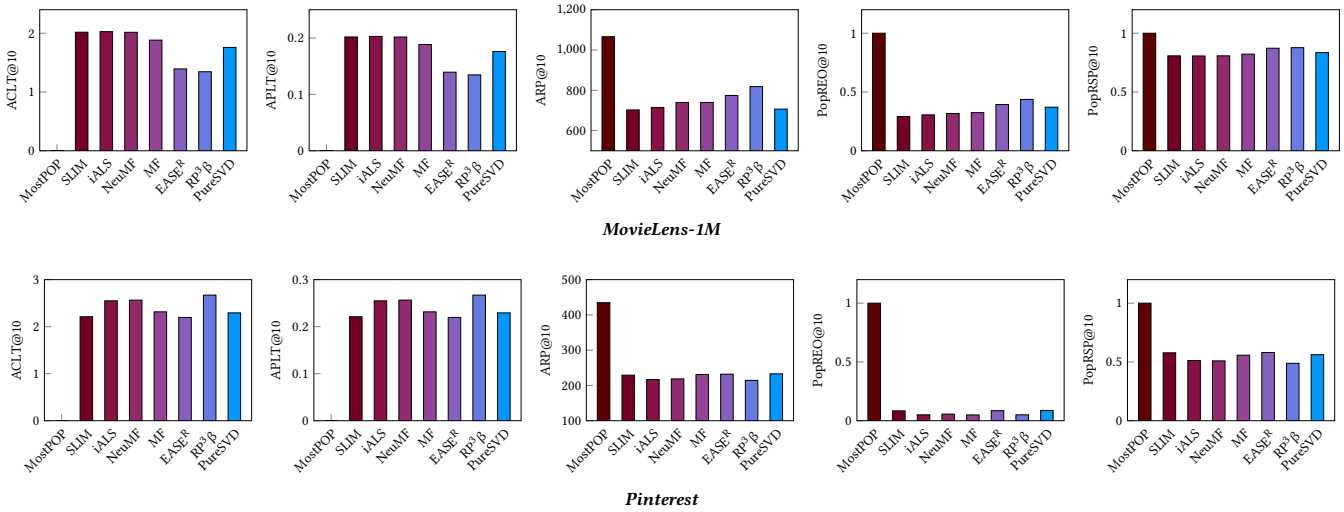

Figure 4: Analysis of Bias for NeuMF, MF and various baselines considering a cutoff @10. For ACLT, and APLT, higher is better, while for ARP, PopREO, and PopRSP, smaller is better.

than from the original paper and, in particular, our extended analysis on accuracy evidences that the difference between this method and other baselines are not always statistically significant.

Our experiments have summarized and re-evaluated results from 3 recent papers, but they can be complemented in several ways. For example, one direction that has been unexplored so far is the effect that the splitting methodology or the item selection strategy could have in all these methods. Recent research has evidenced that how items are selected may affect the evaluation results [6]; however, because we wanted to replicate the exact conditions of these papers, we did not change these experimental settings. It will be interesting to analyze this aspect and how it (may) change the ranking of the methods. Another potential venue to improve this comparison is on the selection of datasets. Again, as we wanted to replicate the original papers, we were limited to use *MovieLens-1M* and *Pinterest*, however, it is crucial to understand how these methods work in other domains and under a wide array of evaluation dimensions, such as those explored here.

ACKNOWLEDGMENTS

The authors acknowledge partial support of the projects: PON ARS01_00876 BIO-D, Casa delle Tecnologie Emergenti della Città di Matera, PON ARS01_00821 FLET4.0, PIA Servizi Locali 2.0 H2020 Passapartout - Grant n. 101016956, PIA ERP4.0, and PID2019-108965 GB-I00, IPZS-PRJ4_IA_NORMATIVO.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning-to-Rank Recommendation. In *RecSys*. ACM, 42–46.
[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing Popularity Bias in Recommender Systems with Personalized Re-Ranking. In *FLAIRS Conference*. AAAI Press, 413–418.
[3] Vito Walter Anelli, Alejandro Bellogín, Antonio Ferrara, Daniele Malitesta, Felice Antonio Merra, Claudio Pomo, Francesco Maria Donini, and Tommaso Di Noia. 2021. Elliot: A Comprehensive and Rigorous Framework for Reproducible Recommender Systems Evaluation. In *SIGIR*. ACM, 2405–2414.
[4] Vito Walter Anelli, Tommaso Di Noia, Eugenio Di Sciascio, Azzurra Ragone, and Joseph Trotta. 2020. Semantic Interpretation of Top-N Recommendations. *IEEE Transactions on Knowledge and Data Engineering* (2020).
[5] Vito Walter Anelli, Tommaso Di Noia, Eugenio Di Sciascio, Azzurra Ragone, and Joseph Trotta. 2019. How to Make Latent Factors Interpretable by Feeding Factorization Machines with Knowledge Graphs. In *ISWC (1) (Lecture Notes in Computer Science, Vol. 11778)*. Springer, 38–56.
[6] Rocío Cañamares and Pablo Castells. 2020. On Target Item Sampling in Offline Recommender System Evaluation. In *RecSys*. ACM, 259–268.
[7] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys*. ACM, 39–46.
[8] George Cybenko. 1989. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.* 2, 4 (1989), 303–314.
[9] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2021. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ACM Trans. Inf. Syst.* 39, 2 (2021), 20:1–20:49.
[10] Maurizio Ferrari Dacrema, Federico Parroni, Paolo Cremonesi, and Dietmar Jannach. 2020. Critically Examining the Claimed Value of Convolutions over User-Item Embedding Maps for Recommender Systems. In *CIKM*. ACM, 355–363.
[11] Ignacio Fernández-Tobías, Iván Cantador, Paolo Tomeo, Vito Walter Anelli, and Tommaso Di Noia. 2019. Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization. *User Model. User Adapt. Interact.* 29, 2 (2019), 443–486.
[12] Asela Gunawardana and Guy Shani. 2015. Evaluating Recommender Systems. In *Recommender Systems Handbook*. Springer, 265–308.
[13] Xiangnan He and Tat-Seng Chua. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In *SIGIR*. ACM, 355–364.
[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. ACM, 173–182.
[15] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM*. IEEE Computer Society, 263–272.
[16] Dietmar Jannach, Gabriel de Souza Pereira Moreira, and Even Oldridge. 2020. Why Are Deep Learning Models Not Consistently Winning Recommender Systems Competitions Yet?: A Position Paper. In *RecSys Challenge*. ACM, 44–49.
[17] Yu-Chin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware Factorization Machines for CTR Prediction. In *RecSys*. ACM, 43–50.
[18] Marius Kaminskas and Derek Bridge. 2017. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Trans. Interact. Intell. Syst.* 7, 1 (2017), 2:1–2:42.
[19] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*. ACM, 426–434.
[20] Yehuda Koren and Robert M. Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook*. Springer, 77–118.
[21] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *KDD*. ACM, 1748–1757.

[22] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. 2014. An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems. *IEEE Trans. Ind. Informatics* 10, 2 (2014), 1273–1284.

[23] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *ICDM*. IEEE Computer Society, 497–506.

[24] Bibek Paudel, Fabian Christoffel, Chris Newell, and Abraham Bernstein. 2017. Updatable, Accurate, Diverse, and Scalable Recommendations for Interactive Applications. *ACM Trans. Interact. Intell. Syst.* 7, 1 (2017), 1:1–1:34.

[25] Steffen Rendle. 2010. Factorization Machines. In *ICDM*. IEEE Computer Society, 995–1000.

[26] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*. AUAI Press, 452–461.

[27] Steffen Rendle, Walid Krichene, Li Zhang, and John R. Anderson. 2020. Neural Collaborative Filtering vs. Matrix Factorization Revisited. In *RecSys*. ACM, 240–248.

[28] Alan Said and Alejandro Bellogín. 2014. Comparative recommender system evaluation: benchmarking recommendation frameworks. In *RecSys*. ACM, 129–136.

[29] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *NIPS*. Curran Associates, Inc., 1257–1264.

[30] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *ICML (ACM International Conference Proceeding Series, Vol. 307)*. ACM, 880–887.

[31] Gunnar Schröder, Maik Thiele, and Wolfgang Lehner. 2011. Setting Goals and Choosing Metrics for Recommender System Evaluations. In *UCERSTI2 workshop at the 5th ACM conference on recommender systems, Chicago, USA*, Vol. 23. 53.

[32] Harald Steck. 2019. Embarrassingly Shallow Autoencoders for Sparse Data. In *WWW*. ACM, 3251–3257.

[33] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. 2020. Assessing ranking metrics in top-N recommendation. *Inf. Retr. J.* 23, 4 (2020), 411–448.

[34] Saul Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*, Bamshad Mobasher, Robin D. Burke, Dietmar Jannach, and Gediminas Adomavicius (Eds.). ACM, 109–116. https://dl.acm.org/citation.cfm?id=2043955

[35] Ellen M. Voorhees. 1999. The TREC-8 Question Answering Track Report. In *TREC (NIST Special Publication, Vol. 500-246)*. National Institute of Standards and Technology (NIST).

[36] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. In *IJCAI*. ijcai.org, 3119–3125.

[37] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. 2012. Challenging the Long Tail Recommendation. *Proc. VLDB Endow.* 5, 9 (2012), 896–907.

[38] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 52, 1 (2019), 5:1–5:38.

[39] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR*. ACM, 83–92.

[40] Yu Zhu, Jinghao Lin, Shibi He, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2020. Addressing the Item Cold-Start Problem by Attribute-Driven Active Learning. *IEEE Trans. Knowl. Data Eng.* 32, 4 (2020), 631–644.

[41] Ziwei Zhu, Jianling Wang, and James Caverlee. 2020. Measuring and Mitigating Item Under-Recommendation Bias in Personalized Ranking Systems. In *SIGIR*. ACM, 449–458.