Measuring Anti-Relevance: a Study on When Recommendation Algorithms Produce Bad Suggestions

Pablo Sánchez
Universidad Autónoma de Madrid
pablo.sanchezp@uam.es

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

ABSTRACT

Typically, performance of recommender systems has been measured focusing on the amount of relevant items recommended to the users. However, this perspective provides an incomplete view of an algorithm's quality, since it neglects the amount of negative recommendations by equating the unknown and negatively interacted items when computing ranking-based evaluation metrics. In this paper, we propose an evaluation framework where anti-relevance is seamlessly introduced in several ranking-based metrics; in this way, we obtain a different perspective on how recommenders behave and the type of suggestions they make. Based on our results, we observe that non-personalized approaches tend to return less bad recommendations than personalized ones, however the amount of unknown recommendations is also larger, which explains why the latter tend to suggest more relevant items. Our metrics based on anti-relevance also show the potential to discriminate between algorithms whose performance is very similar in terms of relevance.

1 INTRODUCTION

In the last years, more and more attention is being paid to the evaluation of Recommender Systems (RS). The first evaluation metrics proposed were error-based metrics such as MAE or RMSE [10], used to measure the difference between the predicted rating and the real rating the user gave to an item. These metrics were applied in the Netflix prize in 2009, where the participants had to reduce the RMSE of the baseline recommender at least a 10%. However, some researchers warned that these classic metrics were not adapted for real world environments since they do not correlate well with user satisfaction [14, 16, 22]. As a consequence, metrics from Information Retrieval (IR) like precision, MAP, or NDCG were used to analyze the performance of top-N recommendations by treating them as lists of items in which the objective was to offer the user a subset of the collection that was as relevant as possible.

However, relevance is not the only dimension we can measure from a list of recommendations. Novelty and diversity [5, 23] or, more recently, item freshness [21] are other important aspects to take into account when suggesting items to users, since, in general, recommending the same types of items, very popular, or very old ones may reduce the interest of the user in the system – even when they are close to the user's interests.

Despite this variety of evaluation dimensions, all these metrics aim to somehow account for the number of relevant items a system provides to the user. Nonetheless, this perspective disregards an important source of information available in many recommender systems: not relevant items. Those items in the collection explicitly rated as such – i.e., with the lower values in a bounded scale or 'not liked' products –, provide a complementary view of the accuracy of recommendations: while it is clear that a recommender suggesting more relevant items is preferred to another showing less relevant items, when the number of relevant items is comparable, techniques producing too many bad recommendations should be avoided, as the user confidence in the system may be undermined [12].

With this idea in mind, we aim to analyze the failures the algorithms make when recommending. As far as we know, this is an issue not addressed in the field, at least from an analytical perspective and in terms of defining new metrics, as we present hereafter. One of the few examples we have found in the area is [7], where the authors measure how close different algorithms produce predictions with respect to the real ratings, and propose to use those findings to create a hybrid recommendation algorithm. In [9], the authors proposed a method to detect bad recommendations using a residual model to capture users' utility. In IR, where there is a long tradition on evaluation, however, we do find authors that introduce alternative definitions for evaluation metrics, either to focus on the most difficult queries [24] or directly on the not relevant documents (frustration metric) [17], and even others that question relevance as a criteria to evaluate the performance of systems [2].

In this paper, we present experiments performed on two real-world datasets, where results from classical metrics (only aware of the relevant items) as well as from new ones (aware of the not relevant items returned by the recommenders) are compared and discussed. We analyze the behavior of these new metrics (named as *anti-metrics* because of their complementary nature in terms of measuring the amount of irrelevance – or anti-relevance – included in a recommendation list) at different cutoffs, together with their discriminative power in some use cases where real algorithms seem to produce comparable performance. According to our results, the proposed anti-metrics could shed light on understanding which recommendation algorithms should be used, especially in those situations where the performance of several techniques is comparable in terms of classical evaluation metrics.

2 ANTI-METRICS: ADDING ANTI-RELEVANCE TO RS EVALUATION

Drawing from IR and recent RS papers, we adapt the Probabilistic Ranking Principle (PRP) for recommendation [3] and interpret it as

RecSys '18, October 2–7, 2018, Vancouver, BC, Canada

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

the starting point for the definition of an objective function that should be optimized by our algorithm [6]. The PRP states that *if a system's response to a query is a ranking of documents in order of decreasing probability of relevance, the overall effectiveness of the system to its users will be maximized* [19].

Hence, to apply the PRP (or to estimate the PRP to evaluate a retrieval or recommender system) we must estimate the probability that a document or item i is relevant for a user (need or profile) u, i.e., $P(\text{Rel} = 1|u, i)$. This quantity is usually translated into RS evaluation as $P(r_{ui} \geq \tau_R|u, i)$, where τ_R is a relevance threshold, meaning that any item in the test set of user u that was rated above (or equal) to such threshold will be considered relevant.

In this paper, we study the *dual PRP problem*: estimating the probability of anti-relevance and ranking the documents according to the opposite of this probability, that is: $1 - P(\text{Rel} = 0|u, i)$. As before, this could be translated into RS as $1 - P(r_{ui} \leq \tau_{AR}|u, i)$ for some anti-relevance threshold τ_{AR}. It should be noted that these estimates can also be computed even when no ratings are available, as long as some measure of negative and positive interaction – e.g., products explicitly *liked* and *not liked* by the user – can be defined.

We argue that most evaluation metrics m are formulated as estimating the classical PRP:

$$m(R_u|\theta_{rel}) = C \sum_{i \in R_u} m(\theta_{rel}(r_{ui})|u, i) \tag{1}$$

since θ_{rel} encodes the dependency on relevance from the PRP $P(r_{ui} \geq \tau_R|u, i)$, where C is a normalization constant and R_u is the recommendation list computed for user u; that is, most evaluation metrics only accounts for the relevant items in the test set of a user. This formulation matches previous evaluation frameworks proposed in the area [23]. Now, to formulate the anti-metrics we follow the dual PRP problem as stated before:

$$\begin{aligned}\overline{m}(R_u|\theta_{arel}) &= C \sum_{i \in R_u} (1 - \overline{m}(\theta_{arel}(r_{ui})|u, i)) \propto \\ &\propto 1 - C' \sum_{i \in R_u} m(\theta_{arel}(r_{ui})|u, i) = 1 - m(R_u|\theta_{arel}) \end{aligned} \tag{2}$$

Thus, our anti-metrics formulation is equivalent to computing any relevance-based metric using an anti-relevance model (where an item is relevant if $r_{ui} \leq \tau_{AR}$) and returning its complement.

We should note that unknown items (those whose ratings are not in the test set) are still considered as not relevant by the classical metrics and by the anti-relevance model – i.e., they contribute with a 0 in the metric computation – however, since the anti-metric reports the complement value, the amount of unknown items is actually affecting the final result, and hence, it should be somehow considered. However, it is not easy to integrate this value in the final metric, since depending on the metric m the final result could be normalized or transformed in some way (e.g., NDCG); nonetheless, for simple, binary metrics such as precision or recall, this issue could be addressed by subtracting the number of unknown items in R_u (or $\text{unk}(R_u)$) as follows: $1 - (m(R_u|\theta_{arel}) + \text{unk}(R_u))$.

Finally, by optimizing the ranking obtained by the dual problem, relevant items are ignored (just as anti-relevant items were ignored when ranking by the original PRP). Hence, in order to balance the information measured in each case, we should combine the metrics based on relevance and anti-relevance. Thus, if we have a measure x computed by some metric m, and another measure $\overline{x}$ computed by its anti-metric $\overline{m}$, we can linearly combine those values (as in [17]) with the average $\mu(x) = 1/2(x + \overline{x})$, the harmonic mean $H(x) = 2\frac{x\overline{x}}{x+\overline{x}}$, or taking the likelihood ratio $LH(x) = \frac{x}{1-\overline{x}}$ inspired by the probabilistic interpretation of m and how this statistic is typically used to take decisions when comparing classifications based on two classes (in our case, Rel = 1 and Rel = 0).

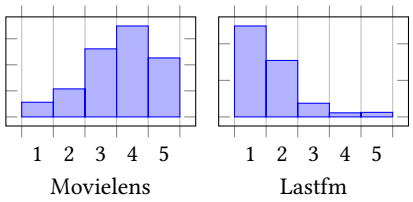

Figure 1: Rating distribution for Movielens and Lastfm

Table 1: Parameters tested in the recommenders. The best configurations were selected by maximizing NDCG@5.

Rec	Parameters
UB	Sim = {Vector Cosine, Set Jaccard}, $k = \{5, 10, 20, \cdots, 100\}$
IB	Sim = {Vector Cosine, Set Jaccard}, $k = \{5, 10, 20, \cdots, 100\}$
HKV	Iter = 20, Factors = {10, 50, 100}, $\lambda = \{0.1, 1, 10\}$, $\alpha = \{0.1, 1, 10, 100\}$
BPRMF	Factors = {10, 50, 100}, BiasReg = {0, 0.5, 1}, LearnRate = 0.05, Iter = 50, RegU = RegI = {0.0025, 0.001, 0.005, 0.01, 0.1, 0.0005}, RegJ = RegU/10

3 EXPERIMENTS AND RESULTS

We have performed experiments with two datasets: Movielens [11] (1M ratings by 6K users on 3.7K items) and Lastfm [4] (93K interactions between 1.9K users and 17.6K items). We have transformed the Lastfm dataset into explicit ratings (in a 1-5 scale) applying the following formula: $r_{ui} = round\left(4\frac{s_{ui}}{S_u}\right) + 1$, where s_{ui} is the number of listenings (or scrobblings) from a specific user u for an artist i, and S_u is the maximum number of listenings from that user.

The rating distribution is shown in Figure 1, where we observe how each dataset is highly skewed to different rating values: whereas the average rating in Movielens is 3.58, in Lastfm is 1.67.

In the reported experiments, we follow the TrainItems methodology [20], where every item in the training set is considered as a candidate item to be recommended by the algorithm, except those already rated by the user in the training set. The relevance threshold τ_R is 4 and the anti-relevance threshold τ_{AR} is 2, meaning that those items in the test set of a user rating with a value ≤ 2 are considered as anti-relevant by the anti-metrics, and those with a value ≥ 4 as relevant by the classical metrics. For both datasets, we have performed a 5-fold cross validation.

In the experiments, we compare different state-of-the-art algorithms. We use two neighborhood-based recommenders (UB, a user-based technique and IB, an item-based one), HKV (the matrix factorization technique from [13]), and a BPR optimization method with an MF technique (BPRMF) [18]. We also show the results of two unpersonalized approaches, a random and a popularity recommender (denoted as "Rnd" and "Pop", respectively). To complete the pool of recommenders, we include a skyline that optimizes for the number of relevant items with a rating $\geq \tau_R$ among those items in the test set of a user ("Skyline") and another that will recommend the anti-relevant items of that user ("$\overline{\text{Skyline}}$"). These two skylines

Table 2: Movielens results @10. The first, second and third best values are marked in bold, with ‡ and † respectively. Note that for $\overline{\text{Rel}}$, Brd, and Unk lower values are preferred.

Rec	NDCG	MAP	$\overline{\text{NDCG}}$	$\overline{\text{MAP}}$	%Rel	%$\overline{\text{Rel}}$	%Brd	%Unk
Rnd	0.004	0.002	‡0.998	‡0.999	0.6	‡0.2	†0.3	99.0
Pop	0.143	0.093	†0.983	†0.990	14.7	†0.9	2.7	81.7
IB	0.253	0.181	0.974	0.985	24.5	1.5	4.9	69.3
UB	†0.298	†0.211	0.973	0.984	†27.0	1.4	4.5	67.2
HKV	‡0.321	‡0.230	0.977	0.987	‡29.6	1.2	4.4	†65.0
BPRMF	0.246	0.178	0.962	0.978	24.9	2.2	5.5	67.5
Skyline	**1.000**	**1.000**	**1.000**	**1.000**	**79.2**	**0.0**	**0.0**	**0.0**
$\overline{\text{Skyline}}$	0.000	0.000	0.000	0.000	0.0	44.6	‡0.0	‡0.0

Table 3: Lastfm results (notation as in Table 2).

Rec	NDCG	MAP	$\overline{\text{NDCG}}$	$\overline{\text{MAP}}$	%Rel	%$\overline{\text{Rel}}$	%Brd	%Unk
Rnd	0.000	0.000	‡1.000	‡1.000	0.0	‡0.0	†0.0	100.0
Pop	0.097	0.077	†0.942	†0.972	1.8	†5.7	0.5	93.1
IB	0.248	0.204	0.857	0.918	4.5	13.0	1.6	83.7
UB	†0.294	†0.248	0.855	0.917	†5.0	13.4	1.7	†83.0
HKV	‡0.316	‡0.272	0.875	0.931	‡5.2	11.9	1.7	84.4
BPRMF	0.240	0.195	0.875	0.932	4.4	11.8	1.5	85.1
Skyline	**0.984**	**0.980**	**1.000**	**1.000**	**12.8**	**0.0**	**0.0**	**0.0**
$\overline{\text{Skyline}}$	0.000	0.000	0.093	0.128	0.0	76.2	‡0.0	‡0.0

are included to explore the maximum and minimum values we can achieve in every metric. The parameters of the recommenders are shown in Table 1. The proposed evaluation metrics were implemented on top of the RankSys library [5], from which we also used most of the recommenders, except for BPRMF, where we used MyMediaLite [8]. All the necessary scripts and source code to replicate the results can be found in the following Bitbucket repository: PabloSanchezP/AntiRelevanceMetrics.

3.1 Performance comparison: metrics vs anti-metrics

In Tables 2 and 3, we show the results obtained by the evaluated recommenders for the different datasets at a cutoff of 10. The first two columns represent two classical ranking metrics (NDCG and MAP) followed by their corresponding anti-metrics as defined in Section 2. The last four columns show the ratio of relevant and anti-relevant items ("Rel" and "$\overline{\text{Rel}}$"), borderline items whose ratings are in the (τ_{AR}, τ_R) interval ("Brd", in this case, items with a rating r such that $2 < r < 4$ in the test set), and the ratio of items whose rating is unknown ("Unk") – these ratios are normalized according to the maximum number of recommended items, in this case, 10.

The first thing we notice in these results is that the best and worst values are achieved by different recommenders (when ignoring the skyline techniques). In particular, Rnd obtains the best result with anti-metrics and the worst result with classical metrics. This is because, as evidenced by the amount of unknown items returned, such recommender tends to not return any relevant item, but it also does not include almost any anti-relevant item in its recommendations, since most of its recommendations are unknown to the users. However, when we use a personalized recommender, the chances of producing a bad recommendation to the user increase, as the worst recommender in terms of anti-metrics is always a personalized technique (either BPRMF in Movielens, or UB in Lastfm).

We attribute this effect to the well-known issue of *missing items not at random* [15] and the datasets being very sparse. In fact, an estimation of the amount of unknown items is directly the rating sparsity (i.e., $1 - |R|/(|U| \cdot |I|)$)[1]. Hence, considering that the recommended items are either relevant, anti-relevant, borderline, or unknown for each user, a random recommender is highly biased towards recommending unknown items, whereas personalized techniques return less unknown items but they have to successfully discriminate between relevant and anti-relevant items – an easier task in datasets such as Movielens, where there are around 3.5 times more relevant items than anti-relevant ones; however, in Lastfm this relationship is reversed, and, hence, the recommenders tend to return more anti-relevant items (see Table 3).

Secondly, we observe UB and HKV perform the best in terms of classical metrics, consistent with literature results. In general, personalized recommenders achieve higher values of classical metrics than other baselines, whereas Pop and Rnd obtain higher values of anti-metrics, mostly due to their tendency to return more unknown items, as discussed before. Results from the anti-metrics are very close to each other and relatively high; the only dataset where $\overline{\text{NDCG}}$ is below 0.9 is Lastfm, since there exist more anti-relevant ratings because of the implicit-to-explicit transformation used (see Figure 1). Additionally, it should be noted that the Skylines do not achieve a perfect score in some cases because of the TrainItems methodology, which prevents using items that only appear in the test set, even though they might be relevant or anti-relevant.

Finally, let us focus on a particular example of how the recommenders behave under these metrics: IB and BPRMF in Lastfm; in Section 3.3 we shall analyze in more detail other recommenders. We observe that, in terms of NDCG and MAP, IB slightly outperforms BPRMF – one possible reason is that the latter recommender does not consider the actual ratings to create its model. However, the amount of relevant items (Rel) returned by each technique is very similar, and, interestingly, the amount of anti-relevant items ($\overline{\text{Rel}}$) is much smaller for BPRMF, at the expense of more unknown items being recommended. This comparison would lead to the following tradeoff: whereas more unknown recommendations probably translate into more novel or diverse recommendations, the user satisfaction is also more uncertain, on the other hand, IB is producing more items explicitly rated by the user as negative recommendations, which, in general, should be avoided.

3.2 Sensitivity to the ranking cutoff

In this section, we analyze the sensitivity of the proposed metrics to different cutoffs. Figure 2 shows the results obtained by some metrics at different cutoffs for Lastfm, similar results were found for the other datasets but are not shown for the sake of space. Here, we include a well-known metric (and its anti-metric) from IR [1]: Success@n (that returns 1 if the recommender returns at least one relevant item up to n) and $\overline{\text{Success}}$@n (that returns 0 if there is at least one anti-relevant item in the first n positions).

In general, we observe that for higher cutoffs the values of the anti-metrics decrease, and, at the same time, classical metrics increase – except for Rnd recommender, whose values are all polarized either into 1 ($\overline{\text{NDCG}}$, $\overline{\text{Success}}$, Unk) or 0 (NDCG, $\overline{\text{Rel}}$, Success). The metrics increase/decrease sharply for the UB, while they remain

[1] Actually, the Unk column is computed in a per-user fashion, hence, the formulation would be more similar to $|U|^{-1} \sum_{u \in U} (1 - |(u, \cdot, r) : r \neq \emptyset|/|I|)$, where $\emptyset$ denotes that a rating is unknown, and U and I are the users and items in the system.

Table 4: Two case studies using the Lastfm dataset with results @5, including the average (μ), harmonic mean (H), and likelihood (LH) aggregated values. We represent in bold the best values in each case study.

Rec	Params	NDCG	MAP	$\overline{\text{NDCG}}$	$\overline{\text{MAP}}$	%Rel	$\overline{\text{\%Rel}}$	%Brd	%Unk	μ(NDCG)	μ(MAP)	H(NDCG)	H(MAP)	LH(NDCG)	LH(MAP)
IB	(VC, 90)	**0.218**	**0.193**	0.856	0.884	0.025	0.161	0.021	**0.794**	0.537	0.538	0.347	**0.317**	1.515	1.662
HKV	(50, 0.1, 100)	0.215	0.188	**0.917**	**0.941**	0.024	**0.097**	**0.016**	0.862	**0.566**	**0.565**	**0.348**	0.314	**2.585**	**3.215**
BPRMF	(100, 1, 0.005)	0.213	0.183	0.870	0.902	**0.025**	0.149	0.021	0.805	0.542	0.542	0.342	0.304	1.640	1.860
UB	(SJ, 40)	**0.266**	**0.235**	0.846	0.877	**0.030**	0.176	0.025	**0.769**	0.556	0.556	**0.405**	**0.371**	1.732	1.907
HKV	(100, 0.1, 1)	0.263	0.232	**0.859**	**0.892**	0.030	**0.160**	**0.024**	0.786	**0.561**	**0.562**	0.403	0.368	**1.873**	**2.147**

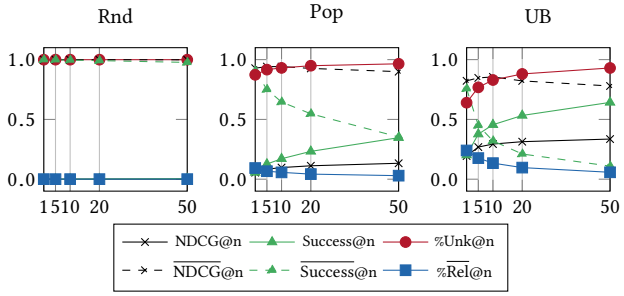

Figure 2: Results for Lastfm at different cutoffs.

more steady for the Pop recommender, evidencing that the personalized recommenders are able to return more relevant items sooner. For the Success metric, in particular, this means that more users receive at least one relevant recommendation when increasing the ranking length, while anti-relevant items still appear in such rankings, according to the $\overline{\text{Success}}$ metric; this effect is more pronounced, again, for the UB recommender than for Pop.

On the other hand, the amount of anti-relevant items is higher at the beginning for UB than Pop, consistent with the results previously presented in Table 3. Symmetrically, the amount of unknown items for UB is smaller than Pop for lower cutoffs, and reach a similar value for both recommenders at the largest cutoff (50).

3.3 Case study: benchmarking similar methods

We now explore how we can use the anti-metrics to decide among algorithms with similar performance. Table 4 shows results for two subsets of different configurations of recommenders whose performance is very close, in this case, we compare IB, HKV, and BPRMF (the three recommenders with almost the same value for Rel and very close values of NDCG and MAP) and UB and HKV (again, with the same value for Rel).

In the first case, we observe that IB and BPRMF recommend the same ratio of relevant items, however BPRMF includes less anti-relevant items, as evidenced by the higher values of the anti-metrics $\overline{\text{NDCG}}$ and $\overline{\text{MAP}}$. The third recommender included in this case, HKV, outperforms the other algorithms in terms of anti-metrics, while achieving a pretty decent performance with classical metrics. This technique presents the lowest value of $\overline{\text{Rel}}$ and, hence, it might be argued that it could be preferred over the other algorithms, considering the three obtain very similar performance when accounting for the amount of relevant items recommended.

To shed some light into these results, Table 4 also includes aggregated values of a metric and its anti-metric based on the average, harmonic mean, and likelihood (see Section 2). We observe that most of these statistics agree on selecting HKV as the best method.

A similar situation is found for the second case with UB and HKV. Here, relevance-based metrics prefer UB, even though Rel is exactly the same for both algorithms. Interestingly, $\overline{\text{Rel}}$ is clearly lower for HKV, which, like in the previous example, could favor this method over the other. In fact, based on the aggregated values, we observe that the statistics based on average and likelihood agree on selecting HKV as the best performing method, however the harmonic mean prefers UB, although for a small margin.

4 CONCLUSIONS

Recommender Systems evaluation is possibly the most critical step during the whole recommendation process as it allows us to determine if the proposed algorithm works as expected or not. We argue that most of current research has been focused on maximizing the number of relevant items retrieved by the recommenders, neglecting the negative suggestions received by the users; hence, current evaluation methodologies are solving only part of the problem, and, as a consequence, almost no progress on understanding the failures of the recommendation algorithms has been made.

In this paper, we have presented a framework where anti-relevance is incorporated into traditional evaluation metrics, which allows us to analyze the behavior with respect to the negative (or anti-relevant) items produced by the recommenders. We have found that bad algorithms in terms of relevance-based metrics – such as the random recommender – do not necessarily return anti-relevant items, since unknown items to the user also play an important role and, typically, account for a large margin of the recommendations presented to the user; while personalized methods tend to return more anti-relevant items than not-personalized approaches. Based on our results, we conclude that the anti-metrics are valuable to discriminate between algorithms that perform at a very similar level in terms of relevance-based metrics. There might be, however, some inconsistencies when deciding which algorithm is actually the best one, but using the aggregation statistics proposed (especially, the likelihood) should provide a more obvious indication about which recommender should be selected in order to find a balance between maximizing successful recommendations and minimizing anti-relevant ones.

In the future, we aim to extend the presented analysis to more families of algorithms and specially difficult recommendation tasks such as cold-start or cross-domain, to understand the behavior of the recommendation techniques in those scenarios. More importantly, we would like to analyze how to extend our framework to situations where no explicit ratings are available, but other form of anti-relevance can be inferred, either directly or through the user interaction with the system.

ACKNOWLEDGMENTS

This work was supported by the project TIN2016-80630-P (MINECO).

REFERENCES

[1] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. 2011. *Modern Information Retrieval - the concepts and technology behind search, Second edition.* Pearson Education Ltd., Harlow, England. http://www.mir2ed.org/

[2] Nicholas J Belkin, Michael Cole, and Ralf Bierig. 2008. Is relevance the right criterion for evaluating interactive information retrieval. In *In Proceedings of the SIGIR 2008 Workshop on Beyond Binary Relevance: Preferences, Diversity, and SetLevel judgments.*

[3] Rocío Cañamares and Pablo Castells. 2018. From the PRP to the Low Prior Discovery Recall Principle for Recommender Systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 1081–1084. https://doi.org/10.1145/3209978.3210076

[4] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011). 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, 387–388. https://doi.org/10.1145/2043932.2044016

[5] Pablo Castells, Neil J. Hurley, and Saul Vargas. 2015. Novelty and Diversity in Recommender Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, 881–918. https://doi.org/10.1007/978-1-4899-7637-6_26

[6] Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong (Eds.). ACM, 659–666. https://doi.org/10.1145/1390334.1390446

[7] Michael D. Ekstrand and John Riedl. 2012. When recommenders fail: predicting recommender failure for algorithm selection and combination. In *Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012*, Padraig Cunningham, Neil J. Hurley, Ido Guy, and Sarabjot Singh Anand (Eds.). ACM, 233–236. https://doi.org/10.1145/2365952.2366002

[8] Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. MyMediaLite: a free recommender system library. 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, 305–308. https://doi.org/10.1145/2043932.2043989

[9] Huan Gui, Haishan Liu, Xiangrui Meng, Anmol Bhasin, and Jiawei Han. 2016. Downside management in recommender systems. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016, San Francisco, CA, USA, August 18-21, 2016*, Ravi Kumar, James Caverlee, and Hanghang Tong (Eds.). IEEE Computer Society, 394–401. https://doi.org/10.1109/ASONAM.2016.7752264

[10] Asela Gunawardana and Guy Shani. 2015. Evaluating Recommender Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, 265–308. https://doi.org/10.1007/978-1-4899-7637-6_8

[11] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *TiiS* 5, 4 (2016), 19:1–19:19. https://doi.org/10.1145/2827872

[12] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22, 1 (2004), 5–53. https://doi.org/10.1145/963770.963772

[13] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy.* IEEE Computer Society, 263–272. https://doi.org/10.1109/ICDM.2008.22

[14] Joseph A. Konstan and John Riedl. 2012. Recommender systems: from algorithms to user experience. *User Model. User-Adapt. Interact.* 22, 1-2 (2012), 101–123. https://doi.org/10.1007/s11257-011-9112-x

[15] Benjamin M. Marlin, Richard S. Zemel, Sam T. Roweis, and Malcolm Slaney. 2007. Collaborative Filtering and the Missing at Random Assumption. In *UAI 2007, Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, Vancouver, BC, Canada, July 19-22, 2007*, Ronald Parr and Linda C. van der Gaag (Eds.). AUAI Press, 267–275. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1699&proceeding_id=23

[16] Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006*, Gary M. Olson and Robin Jeffries (Eds.). ACM, 1097–1101. https://doi.org/10.1145/1125451.1125659

[17] Sung-Hyon Myaeng and Robert R. Korfhage. 1990. Integration of user profiles: models and experiments in information retrieval. *Inf. Process. Manage.* 26, 6 (1990), 719–738. https://doi.org/10.1016/0306-4573(90)90048-7

[18] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, Jeff A. Bilmes and Andrew Y. Ng (Eds.). AUAI Press, 452–461. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1630&proceeding_id=25

[19] S. E. Robertson. 1997. Readings in Information Retrieval. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Chapter The Probability Ranking Principle in IR, 281–286. http://dl.acm.org/citation.cfm?id=275537.275701

[20] Alan Said and Alejandro Bellogín. 2014. Comparative recommender system evaluation: benchmarking recommendation frameworks. In *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*, Alfred Kobsa, Michelle X. Zhou, Martin Ester, and Yehuda Koren (Eds.). ACM, 129–136. https://doi.org/10.1145/2645710.2645746

[21] Pablo Sánchez and Alejandro Bellogín. 2018. Time-Aware Novelty Metrics for Recommender Systems. In *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings (Lecture Notes in Computer Science)*, Gabriella Pasi, Benjamin Piwowarski, Leif Azzopardi, and Allan Hanbury (Eds.), Vol. 10772. Springer, 357–370. https://doi.org/10.1007/978-3-319-76941-7_27

[22] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, Qiang Yang, Irwin King, Qing Li, Pearl Pu, and George Karypis (Eds.). ACM, 213–220. https://doi.org/10.1145/2507157.2507160

[23] 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://doi.org/10.1145/2043932.2043955

[24] Ellen M. Voorhees. 2004. Measuring ineffectiveness. In *SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004*, Mark Sanderson, Kalervo Järvelin, James Allan, and Peter Bruza (Eds.). ACM, 562–563. https://doi.org/10.1145/1008992.1009121