On Target Item Sampling in Offline Recommender System Evaluation

Rocío Cañamares
rcanamares@gmail.com
Universidad Autónoma de Madrid

Pablo Castells
pablo.castells@uam.es
Universidad Autónoma de Madrid

ABSTRACT

Target selection is a basic yet often implicit decision in the configuration of offline recommendation experiments. In this paper we research the impact of target sampling on the outcome of comparative recommender system evaluation. Specifically, we undertake a detailed analysis considering the informativeness and consistency of experiments across the target size axis. We find that comparative evaluation using reduced target sets contradicts in many cases the corresponding outcome using large targets, and we provide a principled explanation for these disagreements. We further seek to determine which among the contradicting results may be more reliable. Through comparison to unbiased evaluation, we find that minimum target sets incur substantial distortion in pairwise system comparisons, while maximum sets may not be ideal either, and better options may lie in between the extremes. We further find means for informing the target size setting in the common case where unbiased evaluation is not possible, by an assessment of the discriminative power of evaluation, that remarkably aligns with the agreement with unbiased evaluation.

CCS CONCEPTS

• **Information systems → Recommender systems**; **Evaluation of retrieval results**.

KEYWORDS

offline evaluation, experimental design, target items, metrics, evaluation bias, discriminative power

ACM Reference Format:
Rocío Cañamares and Pablo Castells. 2020. On Target Item Sampling in Offline Recommender System Evaluation. In *14^{th} ACM Conference on Recommender Systems (RecSys '20), September 22–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3383313.3412259

1 INTRODUCTION

Online evaluation is generally considered the most direct and reliable means to inform decisions on algorithm selection and system updates in deployed recommendation technologies [21]. Offline experimentation remains however an essential instrument to filter out which system variants and change proposals are brought to more expensive online testing [15, 20]. Offline evaluation is also typically the primary means for the exploration of hyperparameter settings and largely stands, on the other hand, as the main instrument for empirical operation available to academic research [16]. Designing offline experiments that match online evaluation is still a challenge [32]. Biases pervade logged data, and confounders easily lurk into data manipulation and experimental procedures, compromising the reliability of evaluation results [7]. Even if it is generally not possible to fully eliminate those shortcomings, distortions can at least be sought to be mitigated as much as possible [10, 24, 37, 38], or at least made known to the experimenter [7]. Careful and aware experiment preparation, and understanding the potential effect of different settings in evaluation outcomes, are primary precautions in designing as informative offline experiments as possible [4, 8].

One typical operation in offline experiment design is sampling disjoint training and test subsets of the offline data [16, 20]. The many ways in which this can be carried out are well known (random, temporal, by user, by item, by rating, inherent in the dataset creation, etc. [16]); the consequences, pros and cons of such different options have been studied by many researchers in the field, and are widely documented and understood [3, 4, 12]. An offline evaluation setup involves however another major setting that has not been paid any comparable attention, and may even not always be explicit or conscious: sampling target user-item pairs for which the evaluated algorithms are requested to output scores for recommendation – that is, selecting the set of candidate items that the recommender systems should rank for each user in an experiment.

A primary aim of this paper is to raise attention for target sampling as a key configuration setting in offline recommender system evaluation. We find that this aspect has not been sufficiently analyzed, and our understanding of the consequence that different target sampling options may have on the outcome of comparative evaluation is, as far as we are aware, quite limited. Roughly speaking, the target setting range goes from including only the user-item pairs in the test set (the smallest sensible option) [2, 31], to all user-item pairs (the largest possible set) [4, 39]. Options in between are also often seen in evaluation reports, where an arbitrary number of non-relevant items are included in the target item set for each user [12, 25]. There is barely, to the best of our knowledge, any systematic study and clear understanding of whether different target sets might produce substantially different results –and if so, which results would be more reliable.

We address the question here, and show that different target subsets can indeed lead to different evaluation outcomes. We find that the difference is in fact systematic, and we give a principled explanation of observed disagreements. We identify a root cause of discrepancy in how different algorithm configurations handle unrated items, and the bias in different methods towards recom-

RecSys '20, September 22–26, 2020, Virtual Event, Brazil

ACM ISBN 978-1-4503-7583-2/20/09…$15.00
https://doi.org/10.1145/3383313.3412259

mending either popular items, or items with a high average rating value. Even though we can understand the observed variations, they still compromise the reliability of comparative results, as it becomes unclear which of the different outcomes we should trust when they disagree. Seeking insights on this issue, we analyze the effect of target sampling from different angles. We examine the gradual variation of evaluation results across the full range of possible target set sizes, seeking to identify the point where the disagreements arise. Considering the well-known biased nature of missing user preference observations [7, 28, 37], we compare the results of common evaluation using non-random unrated items to evaluation with ratings missing at random. We take this as a point of reference to assess the potential introduction of biases and dissociation from reliable evaluation that result at different target size ranges.

Seeking further criteria and practical guidance in target set size configuration, we analyze the loss of information and discriminative power that reducing –or enlarging– the target set may incur. In this direction, we find that standard statistical significance tests provide useful though limited insight for our purpose. We find that a relevant and more informative assessment of the effect of the target size can be obtained by analyzing the amount of ties between systems that a certain experiment setting can produce. Through theoretical and empirical analysis, we show that an insufficient target set can indeed weaken the reliability and informativeness of the comparison between systems. Yet, we also find that the largest possible target may not necessarily enable the best discrimination between systems either. As a final angle of analysis, we check for further anomalies that can result from reducing the target set, when the evaluated systems are not able to rank as many items as the evaluation cutoff may demand.

2 BACKGROUND AND RELATED WORK

For a long time algorithmic research in personalized recommendation was seen as a rating prediction problem, and as such, was evaluated by error metrics such as MAE and RMSE [16, 26]. Rating prediction error was naturally computed over the set of test ratings, as it is obviously not possible to measure the rating error on user-item pairs for which no rating is available. Classification-oriented evaluation such as ROC analysis (e.g. the AUC metric) was also often reported [19], and is similarly insensitive to the presence of unrated items in recommendations. Even when recommendation was evaluated as a ranking task, early work typically restricted the metric computation to the set of items for which the target user has a test rating [2, 31], possibly by inertia from the regression and classification perspective of the recommendation problem.

Including unrated items in the ranking was considered later on along with the use of information retrieval (IR) metrics for evaluation [1, 20], and became common practice as the view of recommendation as a ranking task started to become prevalent [12]. When unrated items are included in the evaluated rankings, the question arises which and how many items should be considered. If we turn to common practice in the IR field (which ranking-based evaluation of recommender systems implicitly or explicitly draws upon [4]), all documents –judged or unjudged– in the search space are generally included as targets to be ranked when evaluating IR systems –no document is left out. The option to exclude unjudged documents has nonetheless occasionally been considered and analyzed in the literature, referred to as *condensed rankings* [5, 34, 36, 42, 43].

Koren [25] proposed an intermediate option in between condensed and full rankings: including *some* randomly sampled unrated (or non-relevant) items –he was the first to suggest this as far as our knowledge goes. The number of non-relevant items was set to 1,000 as a practical or convenient choice, though no particular criterion was suggested for this setting. In the proposed evaluation protocol, Koren further suggested creating several rankings per user, each including only one of the relevant items in the test set for the user [12, 25]; upon this setting modified versions of precision and recall are defined and averaged over such rankings. But reducing the target set can be used just the same with common, more natural settings where a single ranking is created for each user [4]. Koren's idea caught on, and remains popular today: many authors are using target sets with different amounts of unrated items, anywhere from 50 to several hundreds [4, 11, 14, 17, 18, 22, 41, 45]. Many mention that a reduced target set makes experiments more economic, while some may just see this as a simple and convenient experimental setting. But there seems to be no guidance or understanding of what an appropriate target size would be.

In related research, Bellogín et al. [4] represented Koren's approach within a more general and comprehensive framework addressing the main design options in the application of IR methodology to the evaluation of recommender systems. Koren's approach is represented as a particular case in this framework, with the number of unrated items as a configuration parameter. The effect of this variable is briefly examined by Bellogín et al. along a certain range. The analysis focused on the effect on sparsity and metric values though; in particular, the potential impact on system comparisons was not analyzed, as will now be examined here.

Steck [39] did analyze the effect of target sampling when comparing systems, and found discrepancies in specific experiments comparing two particular configurations of a matrix factorization approach. More recently, Cañamares et al. [8] reported similar contradictions in a broader study of experiment design options. These works only consider the two extreme options though: condensed vs. full rankings; and no insights are provided as to which of either option should be more reliable when they do not agree. In the present paper we seek an explanation for the disagreements, we examine the spectrum in between the extremes, and we further wonder whether some criteria might suggest that a particular point in this range is preferable to others.

At the very time of this writing, Krichene and Rendle [27] formally proved under certain simplifications that target set reduction may affect comparative evaluation, and propose metric corrections to attenuate this effect, taking full rankings as the reference. This confirms some of our findings from a different angle, as we shall see. Beyond this, we find and explain that target size affects different recommendation algorithms differently; we seek to understand which size may result in more informative evaluation, and we find that full rankings are not necessarily the most preferable option.

Differences between condensed and full rankings have also been noted in search evaluation in the IR field [34, 36]. The InfAP [42, 43] and bpref [5] metrics, for instance, ignore –and are intended to better cope with– the unjudged documents. However, the qualitative discrepancies between these and full-ranking metrics are typically minor, and largely *system-neutral*, i.e. the differences do not seem to consistently impact any particular IR models over others. In

Table 1: Summary of notation.

U, I	Set of all users, set of all items.
$S \subset U \times I$	Set of "training ratings," i.e. user-item pairs for which a rating is available in the training subset.
$T \subset U \times I$	Set of "test ratings," i.e. user-item pairs for which a rating is available in the test subset.
$S_u \subset I$	Set of "training ratings" for $u \in U$, i.e. items for which a training rating by user u is available.
$S_i \subset U$	Set of "training ratings" for $i \in I$, i.e. users for which a training rating for i is available.
$T_u \subset I$	Set of "test ratings" for $u \in U$, i.e. items for which a test rating by user u is available.
$N_u \subset I \setminus (T_u \cup S_u)$	Set of unrated items that are selected as target for user $u \in U$.
$T_u \cup N_u$	Target set for user $u \in U$.

contrast, as we shall see, the discrepancy can be more important, systematic and biased when evaluating recommender systems.

3 FORMULATION AND PRELIMINARIES

We first describe and formalize in more detail the experiment design setting that is the focus of our analysis: the target set designation. We will use the following notation in the rest of the paper, that we summarize in Table 1. Given a set of users U and items I involved in an experiment, let $T \subset U \times I$ denote the subset of all user-item pairs for which a test rating is available (or sampled) for evaluation. We shall denote by $T_u = \{i \in I \mid (u, i) \in T\} \subset I$ the set of items with a test rating by $u \in U$. Likewise, $S \subset U \times I$ will denote the set of all user-item pairs with a training rating in the experiment, $S_u \in I$ represents the set of items rated by u in the training subset, and $S_i \subset U$ is the set of users with a training rating for $i \in I$. Finally, $N_u \subset I \setminus (T_u \cup S_u)$ shall denote the set of unrated items that are selected as target for a user u.[1]

When only a certain amount of non-relevant target items per user are sampled to evaluate recommender systems, the literature does generally not distinguish between unrated vs. explicitly non-relevant items [4, 12, 25], whereby they would seem exchangeable. Without loss of generality we will assume, for our own convenience, that the target set should always include all the items with a test rating, positive or negative. We thus assume the target set is $T_u \cup N_u$ and let the variation lie on the amount of added unrated items $|N_u|$. This is convenient as the rated non-relevant items are at least needed in the extreme setting where $N_u = \emptyset$ –a target set including only relevant items would otherwise produce the same metric value for all systems, and the experiment would not be informative at all.

We thus take the number of unrated target items $|N_u|$ as a parameter of the experiment, ranging in the interval $|N_u| \in \big[0, |I \setminus (T_u \cup S_u)|\big]$. The largest possible target set $N_u = I \setminus (T_u \cup S_u)$, to which we shall refer as *full targets* following [10], is typically the default in the recommender systems literature [4]. At the other end, with $N_u = \emptyset$ the smallest target set includes only test-rated items [3, 10, 20, 39] –the "condensed rankings" setting in IR evaluation [5, 8, 34, 36, 42, 43], to which we shall refer as *test targets* [10].

The most visible effect of test targets is a very high value in the evaluation metrics –this is generally an overestimation of the true metric value [4], as far as we may assume that an unrated item is less likely to be relevant (because users are typically biased to find and give feedback on items they like more often than items they do not [7, 28, 37]). The full targets option has the opposite distortion: it underestimates metric values by assuming unrated items are all non-relevant [20]. These distortions are not a problem per se, as the absolute value of metrics in offline evaluation is understood to be meaningless by itself [4, 20]. As long as the metric allows observing differences between systems, the value needs not be particularly important. If anything, we might anticipate that artificially low or high metric values might become too similar and cluttered, with reduced effect sizes and insufficient difference to tell systems apart –we will closely examine this possible pitfall in our study. As an extreme, test targets are not applicable to positive-only feedback, since all systems would be tied at the exact same metric value.

Intermediate examples between full and test targets have become frequent in the literature. Originally, Koren [25] arbitrarily took $|N_u| = 1,000$, but other sizes have been used as well: for instance, $|N_u| = 99$ [11, 41, 45], $|N_u| = 100$ [14, 18], $|N_u| = 50$ [22], $|N_u| = 999$ [17], or even varied size ranges [4]. The set N_u can be sampled in different ways. We shall consider here random uniform sampling over unrated items, as the common option; studying other sampling distributions is an interesting direction for future work. The target sampling policy is also orthogonal to the data splitting approach, and any option can be combined independently on each side: target sampling can be exercised in combination with a random rating split [20], temporal splits [26], leave-one-out [13], etc.

4 EXPERIMENTAL APPROACH

Our study combines analytical elaboration with empirical observations guiding, illustrating, confirming and providing further insights on the theoretical analysis.[2] In seeking to elucidate the effects of target sampling, we address the following research questions:

RQ1 – Can the target set size change the outcome of comparative offline evaluation?

RQ2 – Is the change systematic and can we find a principled explanation for it?

RQ3 – Which target set size is suitable for a most reliable comparative evaluation?

RQ4 – How does the target set size affect the discriminative power of an experiment?

RQ5 – Can reduced target sets cause problems with recommendation size that further distort evaluation?

We first describe the experimental setup upon which we will then run all our empirical analysis along the paper.

4.1 Data

Rather than finding which specific algorithms are best, our study aims to assess what experimental design is more reliable in a common and representative offline evaluation scenario, using common resources as are available to a wide research community. Hence as an exemplar of common data for offline recommender system evaluation we take MovieLens 1M [29], possibly the most popular public dataset in the recommender systems literature, containing 1, 000, 209 ratings from 6, 040 users for 3, 706 movies.

As mentioned earlier, one of the angles for assessing evaluation reliability in our study is to contrast the comparative outcomes with

[1] We assume the evaluated algorithms are not requested to recommend items that have a training rating, hence the exclusion of S_u from any target set.

[2] The source code of experiments is available at https://github.com/ir-uam/recsys2020.

unbiased evaluation. We will use for this purpose the Yahoo! R3 dataset [28], containing *missing at random* (MAR) test data sampled uniformly at random over user-item pairs, thus enabling unbiased metric estimates. The data consists of ratings for music entered by users in the Yahoo! LaunchCast streaming service [28]. The dataset involves 5, 400 users and 1, 000 music tracks; it includes 129, 179 *missing not at random* (MNAR) training ratings freely entered by users, and 54, 000 MAR test ratings (10 per user) for items assigned uniformly at random in a research survey.

Since the data splitting procedure is not the focus of our study, we simply use random 5-fold cross-validation, on both MovieLens 1M and the MNAR "training" subset of the Yahoo! R3 release. For simplicity, the few users that end up not having training data are removed from the experiment, as it is not possible to produce any kind of personalized recommendation for them. With Yahoo! R3 we can compute regular biased metric values using the test subset of the rating split of the MNAR training data, and unbiased metric values using the MAR test ratings provided in the dataset instead.

4.2 Algorithms

The goal of our study not being to find the best algorithms, but to examine potential pitfalls in evaluation, we select a few common, simple and representative algorithms and configurations. These include three collaborative filtering algorithms: implicit matrix factorization (iMF) [23], and user-based nearest-neighbors (kNN) in the normalized and non-normalized variants [6], with cosine similarity. In addition, we include three non-personalized recommendations: ranking by popularity (rating count), by the average rating, and random recommendation. When a collaborative filtering algorithm falls short of coverage (as discussed later in Section 6.4), we fill the missing rank positions with random items.

The hyparameters of kNN and iMF are set by grid search. The grid points for k in kNN are 10 to 100 by steps of 10; 100 to 1,000 by steps of 100; and 1,000 to $|U|$ by steps of 1,000. For iMF, the grid is $k \in \{5, 10, 50\}$, $\alpha \in \{1, 10, 100\}$, $\lambda \in \{0.1, 1, 10\}$. The hyperparameters are tuned in two configurations, summarized in Table 2: one is optimal for P@10 with full targets, and the other with test targets. We shall refer to these variants by appending "(full)" or "(test)" to the algorithm name, e.g. "iMF (full)" designates the implicit matrix factorization with hyperparameters tuned for full targets. Note that for non-normalized kNN the two optimal configurations are the same, whereby a single version "kNN (full/test)" of this algorithm will appear in the results that we will report. Normalized kNN is configured to require at least three neighbor ratings to rank a target item. The average rating takes binarized rating values, mapping to 1 the rating values above or equal to 4, and lower ratings to 0 –the average thus represents the ratio of positive ratings of each item. For more effective recommendation the average is smoothed, as is common, by Dirichlet smoothing (also called "Bayesian average") with $\mu = 1$ [7, 44]. Considering the different hyperparameter settings we will be comparing a total of eight systems in most of our experiments. Only at a particular point in Section 6.2 we will use a larger set by adding in further algorithm configurations to the pool.

4.3 Metrics

We evaluate recommendations by ranking-oriented metrics: precision, recall and nDCG. As is usual, we use test ratings as relevance judgments where a rating value of 4 or higher is taken as indicative of positive relevance, and considering unrated items as non-relevant. For simplicity, we take binarized ratings in nDCG –we observe quite the same outcomes with graded relevance. We use a typical cutoff of 10 as our primary metric depth. Rather than reporting statistical significance of comparisons for each set of results, we will examine the global variations of p-values across experimental configurations from a comprehensive perspective in a dedicated Section 6.2, where we analyze the effect of experimental settings on the discriminative power of evaluation based on one-tailed paired t-tests.

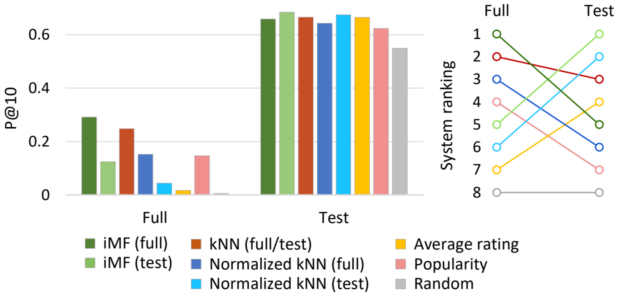

Figure 1: Full targets vs. test targets in MovieLens 1M. The bar charts show P@10 for each algorithm in the two target configurations, and the line graph on the right compares the system rankings by P@10. Each pair of crossing lines represent a disagreement between full vs. test targets in the comparative evaluation outcome. "Full" and "test" in parenthesis next to algorithm names in the legend indicate hyperparameter settings optimized for full and test targets respectively. Non-normalized kNN has only one variant in the figure since the same hyperparameter configuration is optimal for full and test targets, as mentioned in Section 4.2.

5 RESULT CONSISTENCY

We start our analysis by considering the two extremes of the target size range: test targets vs. full targets. Running a common example experiment, we observe contradictions between the two settings. We then elaborate a principled explanation for the observed disagreements. After this, we use comparison to unbiased evaluation as a reference to discern which setting is more reliable when different outcomes disagree.

5.1 Full vs. Test Targets

As a first step in our analysis we run a basic evaluation of our set of systems in full targets and test targets in MovieLens 1M, as a representative example of a typical experiment. Figure 1 shows the results for P@10 (equivalent outcomes are obtained with recall and nDCG). A first difference between the two settings becomes immediately apparent: test targets produce much higher metric values than full targets, illustrating our discussion in Section 3. Since no unrated items are included in the test targets setting, the relevance density is much higher than with full targets, hence the higher precision overall: delivering relevant recommendations is easier in this setting. This may not be important as it is long understood that the absolute values of offline metrics need not have a meaningful interpretation beyond a comparative purpose [4, 20].

The higher metric values with test targets come along however with a much reduced effect size in the difference between systems. Algorithms are left with less room for differentiation: even random recommendation would seem to do rather well, achieving P@10

Table 2: Hyperparameter settings of algorithms, optimized for the full and test targets setting in each dataset.

	MovieLens 1M		Yahoo! R3	
	Full Targets	Test Targets	Full Targets	Test Targets
Non-normalized user-based kNN	$k = 100$	$k = 100$	$k = 200$	$k = 200$
Normalized user-based kNN	$k = 10$	$k = 1000$	$k = 20$	$k = 500$
Implicit MF	$k = 50, \alpha = 1, \lambda = 1$	$k = 10, \alpha = 100, \lambda = 1$	$k = 10, \alpha = 1, \lambda = 1$	$k = 10, \alpha = 10, \lambda = 1$

above 0.5. One would wonder whether the reduced differences resulting from test targets might affect the statistical significance of system comparisons. We address this question in Section 6.2.

Even more importantly, we see some qualitative contradictions between the two experiment settings in the comparison between systems. One of such disagreements is not surprising: the variants optimized for full targets are suboptimal in test targets and vice-versa. Albeit predictable, this mismatch is important, as one is left in doubt as to which hyperparameter configuration should be chosen when bringing the outcome of offline evaluation to the decision that the experiment is intended to inform, e.g. in a production environment. But further discrepancies are observed: with full ratings, popularity-based recommendations are deemed more effective than ranking by the average rating, and non-normalized kNN is better than the normalized variant, whereas the opposite is the case with test targets. To more clearly highlight the discrepancies, the line graph on the right of Figure 1 displays the ranking order of the algorithms (by decreasing P@10) in the full vs. test targets settings. We can see several crossings between the lines, that reflect the inversions in comparisons. For instance, the best algorithm with full targets (iMF full) is in the fifth position with test targets.

5.2 Analytical Explanation

Our observations are in line with former results from Steck [39], and extend them. Moreover, we find explanations for the observed results and contradictions. We can do so upon a matrix factorization scheme based on Steck's formulation [39], as follows. Let us consider a common matrix factorization approach that takes an item offset vector $b \in \mathbb{R}^{|I|}$ and two matrices $p \in \mathbb{R}^{|U| \times k}$, $q \in \mathbb{R}^{|I| \times k}$, and recommends items i to users u by ranking them based on the score $b_i + p_u q_i^t$, where the vector and matrices are obtained by minimizing the following cost function:

$$L(b,p,q) = \sum_{(u,i) \in U \times I} w_{u,i}((b_i + p_u q_i^t - r(u,i))^2 + \lambda(|p_u|^2 + |q_i|^2 + b_i^2)) \quad (1)$$

$$\text{with: } w_{u,i} = \begin{cases} w_1 > 0 \text{ if } (u,i) \in S \\ w_0 \geq 0 \text{ otherwise} \end{cases}$$

where $r(u,i)$ is imputed a value r_0 if $(u,i) \notin S$, i.e. when a training rating value is not available.[3]

Steck [39] found out that $w_0 = 0$ worked better for test targets, whereas $w_0 > 0$ (with a low r_0) worked better for full targets. Setting $w_0 = 0$ means training only on rated user-item pairs, whereas $w_0 > 0$ trains on rated and unrated pairs. Steck's observations thus have a coherent explanation: if the training and test data are sampled from a similar distribution, we should get better results when training on user-item samples drawn from an as similar distribution to the test data as possible. In the case at hand, this means: ignoring or including unrated items in training when they are ignored or included in evaluation, respectively, should produce the best results.

Our observations in Figure 1 can be explained by an extension of this rationale. The iMF algorithm that we use in our experiment always takes $r_0 = 0$, $w_0 = 1$, and $w_1 = 1 + \alpha$ (see [23] for detail). High α values have a similar effect as $w_0 \sim 0$: the algorithm is mainly trained on rated user-item pairs, and we can expect this to be most appropriate when evaluating with test targets (ignoring unrated items). For the same reason, small values of α make $w_0 \sim w_1$ and should work better in full targets (involving all unrated items). The optimal hyperparameters for iMF shown in Table 2 for the respective setting, confirmed in Figure 1, support this explanation.

Furthermore, the average rating and popularity can be seen as particular cases of matrix factorization with $k = 0$ and $\lambda = 0$. With this setting, taking partial derivatives in Equation 1 above and solving for b_i, we get an easy exact solution minimizing L:

$$\frac{\partial L}{\partial b_i}(b,p,q) = 0 \Leftrightarrow b_i + p_u q_i^t = b_i = \frac{w_1 \sum_{u \in S_i} r(u,i) + w_0 r_0(|U| - |S_i|)}{w_1 |S_i| + w_0(|U| - |S_i|)}$$

where $S_i \subset U$ denotes the set of users rating the item i in the training set. Now with $w_0 = 0$, the ranking function becomes $b_i = \sum_{u \in S_i} r(u,i)/|S_i|$, the average rating of i. Following our line of thought, this should work best when evaluated with test targets. On the other hand, taking $w_0 = w_1$ and $r_0 = 0$, the recommendation becomes $b_i = \sum_{u \in S_i} r(u,i)/|U| \propto \sum_{u \in S_i} r(u,i)$, i.e. the popularity of item i; and this should yield better results when evaluated with full targets. The average rating can thus be seen as a non-personalized recommendation that "trains" on rated user-item pairs, while popularity "trains" on rated and unrated pairs. Again, this explains exactly the results observed in Figure 1.

Finally, in prior work [7] we proved that normalized user-based kNN is biased by construction towards the average item rating, and the non-normalized variant is biased to popularity. This provides a consistent explanation for the observed behavior of kNN in Figure 1: the normalized variant works better in test targets (as the average rating does), while the non-normalized variant is more effective in full targets (as is popularity).

We have found explanations for why the results with full vs. test targets disagree, but we still do not have any indication of which of the two sets of comparisons might be the "correct" one –we address this next.

5.3 Evaluation Bias

Our analysis so far leaves an open question: given the observed contradictions in outcomes resulting from different target sizes, which –if any– is more reliable? To address this question different reliability criteria can be considered, and each opens a direction for further study. As a first criterion, we focus on unbiased system comparisons as an objective reference. Offline evaluation is known

[3] Note that $b_i \equiv r_0$ in Steck's formulation [39]. We define them here as two distinct parameters to simplify our analytic elaboration.

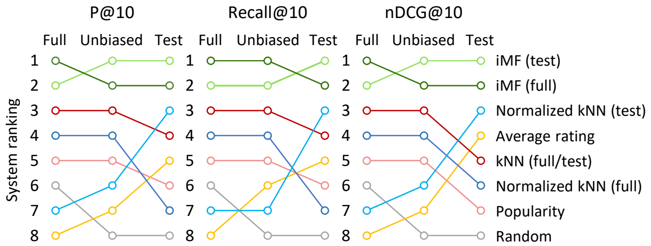

Figure 2: System ranking of observed full and test targets vs. unbiased precision, recall and nDCG in Yahoo! R3.

to be immersed in bias resulting from ratings missing not at random [7, 24, 28, 37]. The missing ratings are precisely in the focus of our study: the N_u set. Hence we consider the bias carried by this set as a relevant angle to check for in the observed contradictions. Specifically, we examine how closely the results with different amounts of unrated targets may match unbiased evaluation. We use for this purpose the Yahoo! R3 data, supporting biased and unbiased evaluation, as mentioned earlier in Section 4.1. Biased evaluation is run by using the MNAR training subset only, with 5-fold cross-validation akin to MovieLens. For unbiased evaluation, the same 5-fold MNAR training subsets are supplied as input to the evaluated algorithms, but the (single) MAR test data is used for metric computation.

Figure 2 shows the comparison of biased (full and test targets) and unbiased evaluation, now also showing recall and nDCG. We can observe that neither option, full or test targets, really seems to highly agree with unbiased system comparisons overall. The amount of inversions might seem slightly higher with the test targets setting, but the difference is far from sufficient to be anything close to conclusive. We may then wonder whether some point in between the two extremes might do better at matching unbiased evaluation. We seek answers to this question in the next section.

6 THE SWEET SPOT IN TARGET SIZE

We now turn to our next question: given the mismatch with unbiased evaluation observed with both test and full targets, can we find a better setting at some intermediate target size? To address the question, we start by observing the gradual evolution of evaluation along the target size interval, in terms of the metric values and the system comparison ranking. After that, we analyze the agreement with comparisons across the same range, using MAR data. We then introduce statistical power analysis as an additional means to assess desirable properties of experiments. Finally, we examine the potential experiment degradation resulting from coverage losses when target sets become insufficient.

6.1 Varying the Target Size

We begin by extending the observations in the previous section to a more detailed examination of where in the continuous range from the smallest to the largest possible target set the contradictions between comparisons arise. Figure 3 extends Figure 1 with the gradual evolution of metric values along the target size axis. To the evaluation with MovieLens 1M (left) we add Yahoo! R3 as a second dataset (right) in "biased mode", i.e. using 5-fold cross-validation of the MNAR training data subset alone (we will use the MAR test later). We show two additional metrics, Recall@10 and nDCG@10, for further perspective. For each metric and dataset, the figure displays evaluation outcomes in two ways: the metric values (left) and the ranking of systems by decreasing metric value (right). As in Figure 1, the latter graph helps notice inversions in system comparisons, now at the precise point where they occur. To better appreciate the changes around small target sizes the x axis points are roughly logarithmic, manually adjusted to integer values.

We can see that a similar amount of inversions occurs for all metrics. With MovieLens, most inversions arise when the target set becomes rather small (around $|N_u| \lesssim 100$), though some occur quite early (e.g. iMF-test vs. popularity), while inversions are more spread out in the Yahoo! dataset. The clutter of metric values in the test targets setting is more pronounced in the Yahoo! dataset. This is because the higher sparsity of the Yahoo! data causes a higher amount of ties between systems compared to MovieLens 1M, as we will discuss in Section 6.3. This is intensified in Yahoo! R3 by a heavy coverage loss affecting all systems when target sets are small, an effect we explain in Section 6.4.

We now extend our comparison with unbiased evaluation to a range of target sizes, using the MAR test data of Yahoo! R3 for this purpose, as in Section 5.3. We take the Kendall's τ correlation as a measure of the number of inversions between regular and unbiased evaluation, and study it along the target size spectrum. We show this as a curve (the red line) in Figure 4, left column. The correlation is measured between the system rankings obtained with each target size across the x axis (biased evaluation with MNAR data), and the system ranking by the unbiased metric estimate (using the MAR test data) –the higher the curve runs, the better the experiment represented in the x axis matches unbiased evaluation. The graphs suggest that the full and test targets extremes are not ideal, and the best approximation to unbiased evaluation (the peak of the red curve) is reached somewhere in between. We also see that the ideal point differs from one metric to another: $|N_u| = 100$ seems best in P@10 and Recall@10, while $|N_u| = 20$ is better for nDCG@10.

We might stop here and conclude that we can find the sweet spot in target size by contrast to unbiased comparisons. However, MAR test data is usually not available in common offline experiments for recommender system evaluation, and is a strong and very restrictive requirement. For instance, we cannot determine a desirable target size for MovieLens 1M based on this line of analysis alone, since all the ratings are MNAR in that dataset. Besides, if MAR data is available, we may just do unbiased evaluation, and target size may become a non-problem. We therefore continue our analysis seeking further insights that can help in deriving practical criteria for the target size in the absence of MAR data.

6.2 Discriminative Power

One dimension on which the informativeness of evaluation is often assessed is discriminative power in telling two systems apart, i.e. in determining with any certainty which of two systems is better. Power alone is not a guarantee that the outcome of evaluation faithfully represents what we intend to assess (e.g. user satisfaction), but it is an indication that comparative outcomes are not just produced by chance. Statistical significance tests are commonly used for this purpose [9, 35]. As a summary measure of the statistical power of

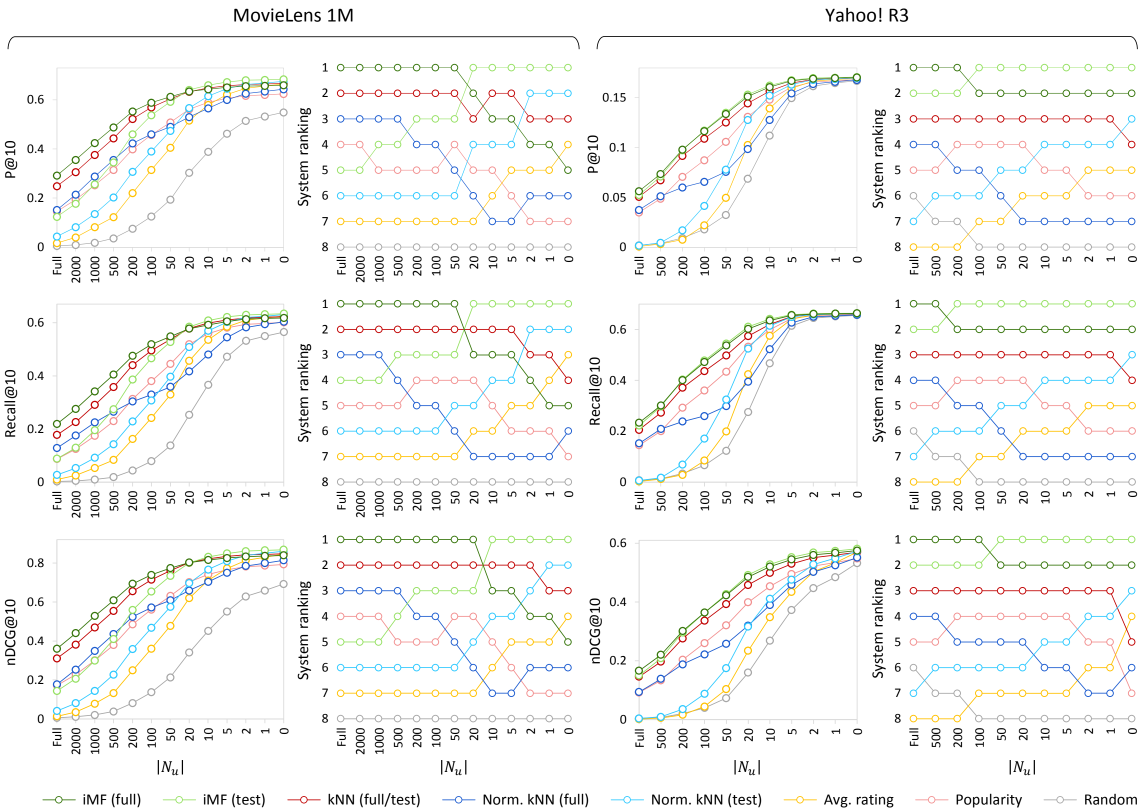

Figure 3: Precision, recall and nDCG at cutoff 10 vs. the amount of unrated targets in MovieLens 1M (left) and Yahoo! R3 (right).

an experiment as a whole (rather than for each pairwise comparison), we can take the sum of p-values for all system pairs in an experiment, as in e.g. [33, 40]. Figure 4 shows this sum for each target size as a (black) curve for precision, recall and nDCG in Yahoo! R3 (left) and MovieLens 1M (right), where p-values are computed with one-tailed paired t-tests for all pairwise metric differences. The higher the black curve runs, the weaker is the statistical power of the experiment represented in the x axis.

For a smoother curve, we add further systems to the pool: specifically, all the configurations of kNN and iMF explored in hyperparameter tuning (see Section 4.2), for a total of 57 systems, amounting to 1,596 pairwise comparisons (and as many p-values). Otherwise, with only eight systems we would only get 28 p-values, too few and noisy to perceive any clear trend in a heavily jagged curve. Even though the set of systems behind this one curve is different from the rest of our measurements, we can hope to get a perspective on the effect of target size on discriminative power that is still relatable to the rest of our analysis. Alternatively, we can accept the limitations of statistical tests as a tool in addressing our research questions; we elaborate on that critical perspective in the next section, providing in fact a better alternative to the p-value analysis.

Interestingly, we see that the significance curve (black) has roughly opposite monotonicity to the correlation with unbiased evaluation (red) in Yahoo! R3 analyzed in the previous section (we naturally do not have a "red curve" for MovieLens as the dataset does not provide MAR data for unbiased evaluation). That is, experiments seem to match unbiased comparisons better when statistical power is stronger. Statistical tests may thus hint a plausible explanation for the correlation to unbiased evaluation, and might provide a useful criterion in predicting and settling for a suitable target set size. As we just mentioned though, this analysis has limitations. On top of the need for a large system pool, the power curve seems somewhat unstable (particularly so in MovieLens), and it is unclear whether the ups and downs reflect any true sudden difference, or some kind of statistical noise. In particular, the power analysis does not clearly tell whether the "sweet spot" identified earlier (peak of the red curve in matching unbiased evaluation) is preferable to full targets or not. We therefore seek further clarification.

6.3 Tie Analysis

Naive intuition might suggest that taking all items in |N_u| (i.e. the full targets setting) should result in the most informative observations, following a general principle of using as much information as is available. A simple thought experiment proves this not to be the case though: consider a scenario where we have an unlimited supply of items without test ratings. As we take $|N_u| \to \infty$, relevant items would be so rare amidst heaps of unrated items in the full targets setting that placing them in the top n would become an increasingly daunting task for any recommender system, and eventually all algorithms would converge towards a global tie with metric values converging towards zero –and the experiment would converge towards providing zero information. Before reaching this global tie, non-zero differences between algorithms would become increasingly noisy, certainly mismatching unbiased evaluation.

On the other end, as the target set becomes smaller, the number of different possible size-n subsets of the target set (i.e. the different

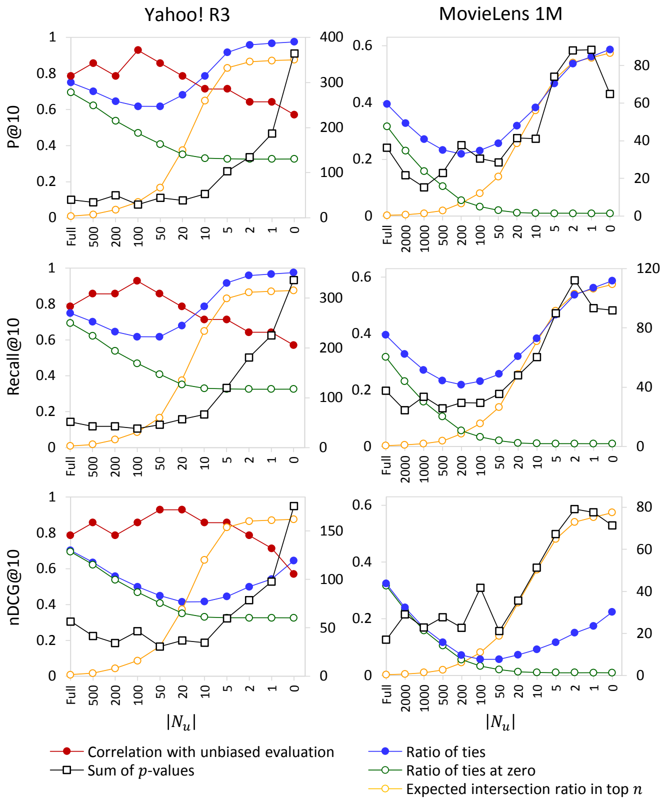

Figure 4: Seeking the "sweet spot" in target size on Yahoo! R3 (left) and MovieLens 1M (right) for precision (top), recall (middle) and nDCG (bottom) at cutoff 10. Note that the metrics in the figure are not axis titles but "graph row" titles. For each target set size (x axis) and metric, the graphs display: the ratio of ties between the different algorithms (blue curve); the ratio of ties where the metric value is zero (green curve); the expected intersection ratio in the top $n = 10$ between two random rankings (yellow curve); and the sum of p-values for all system comparisons (black curve). The latter curve ranges along the secondary right y axis in the graphs, while all other curves range in the left y axis. Additionally, for Yahoo! R3, the graphs show the Kendall's τ correlation between the system ranking for each target size and the ranking by unbiased evaluation (red curve). Note that the number of ties (blue and green) for precision and recall are naturally the same (for each dataset). The yellow curve is the same for all graphs of each dataset, since the expected intersection is metric-independent.

top n recommendations) is drastically reduced. In fact, if $|N_u \cup T_u| \leq n$, the top n recommended items would be just the same for any algorithm, the only possible difference being the item order –but not the selection. For some metrics like precision or recall, this means a plain tie. Hence, different algorithms have an increasingly hard time in producing different recommendations and metric values from each other as we approach test targets –differences become increasingly few and, again, noisy. This may not be the case to the same degree for rank-sensitive metrics such as nDCG, which are able to discriminate between different rankings of even the same set of top n items, if one ranking places relevant items higher above n than the other.

The rationale in this line of reasoning is that extreme target settings may produce many ties, these may distort the evaluation, and may well explain the mismatch with unbiased system comparisons. We therefore check the number of ties along the target size axis. The blue curve in Figure 4 shows the ratio of ties (i.e. pairs of tied systems) on a per-user basis for each metric. As we might expect, the number of ties decreases monotonically as we get away from the extremes, until reaching a minimum somewhere in between. To further confirm our intuition, we show in the figure the amount of ties at a metric value of zero (green curve), and the expected intersection ratio at top n between two random rankings (yellow curve), as a measure of the (lack of) room for distinction between systems, which can be expressed in closed form as $\min(1, n/|N_u \cup T_u|)$. As we can see, these two curves are plausible components of the overall trend in ties: as a lower bound, ties at zero (green) correlate with the lift in total ties (blue) towards full targets. Likewise, the raise in total ties towards test targets follows the raise in top n intersection (yellow). It is also interesting to note that nDCG is more tolerant than precision and recall to small targets and high intersection: since nDCG is sensitive to item position, we see that the expected ranking intersection (yellow) has a less tight ascendancy on the number of ties (blue) for nDCG than it has for the other two rank-insensitive metrics. Fewer ties are thus produced and may explain why the correlation with unbiased evaluation stands up to smaller target sets for nDCG than for precision and recall.

Quite remarkably, we find that the total number of ties (blue) and the correlation with unbiased evaluation (red) now have almost perfectly opposite monotonicity, i.e. the experiment matches unbiased comparisons best when ties are the fewest. In contrast to the p-value analysis described earlier, requiring a large pool of systems to perceive an effect, eight systems are enough for a meaningful observation of tie evolution as we can see in Figure 4. The tie analysis can thus provide a more precise explanation and prediction for the correlation to unbiased evaluation, and a better criterion in settling for a suitable target size. In MovieLens 1M, this analysis would suggest taking around 50 unrated items for nDCG and 200 for precision and recall as the setting minimizing the number of ties, based on observation of Figure 4 right (blue curve). Interestingly, this suggests that target sizes reported in this range in the literature [11, 14, 18, 22, 41, 45] may have been in fact a good choice.

We would not infer from our observations that discriminative power is itself a direct cause of a correspondence with unbiased evaluation. We see a more plausible effect of better discrimination as an indirect enabler. In spite of the bias in MNAR data, it has been found that natural user behavior may lead to a fair degree of agreement between biased and unbiased evaluation, mainly due to a strong relevance bias in the observation sampling [7, 10]. In our case, we hypothesize that discriminative power reflects an efficient use of information in an experiment, where a reduced exposition to noise lets the correspondence to unbiased evaluation emerge more clearly. We should not discard a degree of chance either in how tightly the optimal points match each other in our experiment.

Our analysis may also point to a limitation of statistical tests in the way ties are handled. Ties would add evidence in support of the null hypothesis that two systems are equally effective and

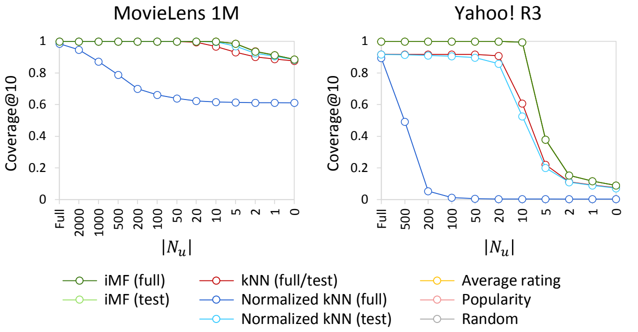

Figure 5: Recommendation coverage vs. target size in MovieLens 1M and Yahoo! R3. Note that iMF (test), average rating, popularity and random have the same coverage as iMF (full), whereby their curve is hidden underneath the latter.

indistinguishable. However, ties do not result in higher p-values –they generally tend to be ignored in statistical tests, and there is not a clear principled approach to properly take them into account in the computation of p-values [30]. This is sometimes dismissed in the implicit hope that ties are rare in recommender systems evaluation, which is far from true. Tie ratios around 75% in Yahoo! R3 and 40% in MovieLens in our evaluation (which is an example of typical offline experiments) seem too high to be ignored and just rely on p-values alone. While statistical tests might not find any objection with full targets, the comparison to unbiased evaluation would hint in the same direction as the tie analysis: that full targets may not be the best option.

6.4 Coverage Loss

An additional pitfall with small target sets, noted by Cañamares et al. [8], is the potential loss of coverage for certain algorithms. For instance, with user-based k nearest neighbors, if a user has few relevant test ratings it may happen that none of her k neighbors has training ratings for any of the (now few) target items. If so, the target set can simply not be ranked, and the system cannot deliver a recommendation at all for this user. Even when a few items can be ranked, they can be fewer than the desired metric cutoff requires, which is also problematic.

Coverage shortfall can be handled in different ways when computing metrics [8]: it can be forgiven by reducing the metric depth to the recommendation size when it falls short, and ignoring the users who are delivered empty recommendations; or it can be punished by counting as non-relevant the empty positions. Either option can have strong effects in evaluation that very easily go unnoticed: an algorithm can be deemed to be remarkably accurate (or the opposite), when in truth it is only suffering a massive loss of coverage caused by the experimental design. An intermediate option fills in the lost coverage with a fallback recommendation algorithm. In our experiments we use random recommendation for this purpose, making the performance degradation noticeable.

We illustrate in Figure 5 the extent of coverage loss in our experiments, showing coverage@10 across the target size range for each algorithm and dataset, with:

$$\text{coverage@}n = \frac{1}{|U|n} \sum_{u \in U} \min(n, |R_u|)$$

as defined in [8], where R_u denotes the set of items that an algorithm is able to rank. We can see in the figure how reducing the target size results in a growing coverage loss. Normalized kNN (full) is the worst hit algorithm because of its small neighborhood ($k = 10$), plus the requirement of at least 3 neighbor ratings to compute the score of an item. With fewer targets items, it becomes increasingly hard to find candidate items rated by three or more (out of ten) neighbor users. This adds to the low performance in test targets of a kNN configuration that is optimized for full targets.

The coverage loss is particularly dramatic in Yahoo! R3. This is because the training subset for MNAR evaluation is about half as dense as MovieLens 1M, whereby collaborative filtering has added difficulty in finding the needed data overlaps for scoring items. Moreover, the test target size in Yahoo! is less than five items per user on average, whereas MovieLens has over 30 test ratings per user. Most users in Yahoo! therefore have fewer test items than the metric depth: $|T_u| < 10$. In the test targets setting, this means that the size of recommendations falls short for all algorithms (even non-personalized recommendations), hence a drastic loss of coverage. Yet, since this specific effect similarly impacts all algorithms, it should not affect comparisons between systems, aside the loss of discriminating power discussed earlier.

As a conclusion, coverage loss would also advise against small target sets, and test targets in particular. In fact, anything smaller than full targets may incur some coverage loss for some algorithms that are particularly sensitive to this (as is our normalized kNN-full variant), even for target sizes that we are finding particularly adequate from other perspectives. Specific treatment and examination would be needed to properly evaluate such algorithms, possibly taking into account what coverage may specifically imply in the actual conditions and requirements for which the algorithm is envisioned to be deployed (e.g. in a production setting).

7 CONCLUSIONS

As far as we know this is, along with [27], the first work after [39] to specifically research the discrepancies in offline evaluation resulting from different target size configurations; the first to seek a principled explanation of the disagreements, to examine this in a continuum perspective, and to analyze the consequence in evaluation reliability, which we do from different angles. We find and explain problems when restricting recommendations to the items with test data, which would generally advise against this option unless a specific reason would prescribe it. At the same time, we find that removing *some* (or even a significant) amount of unrated items from recommendations can make evaluation most informative –a conclusion supported by both analytical elaboration and empirical confirmation.

Comparison to unbiased evaluation is a revealing perspective in this regard. We find a connection between the deviation from unbiased results and the discriminative power of experiments. To this specific respect, tie analysis in comparative evaluation seems more informative than traditional statistical significance tests. We would hence contend, as a collateral conclusion, for systematically reporting tie ratios along with traditional p-values for a better assessment of the discriminative power of comparative experiments –as was occasionally analyzed long ago [5].

Many directions for future work can be envisioned. Contrasting our analysis to actual validation in user studies, with direct, richer feedback from users and more extensive control on evaluation biases, is one of them. Extending our study to further offline datasets is also natural step for a more complete perspective. Different splitting procedures and other distributions in target item sampling (e.g. by popularity) can be explored as well. The explanation why different algorithms are affected differently by the target set size can sought to be extended to further algorithmic families. The connections to evaluation biases [7, 10, 24, 37, 38] can be researched in further depth as well, and would extend our understanding of the role of the target set in offline evaluation.

ACKNOWLEDGMENTS

This work was partially supported by the Spanish Government (grant ref. PID2019-108965GB-I00).

REFERENCES

[1] G. Adomavicius and A. Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (June 2005), 734–749.

[2] C. Basu, H. Hirsh, and W. W. Cohen. 1998. Recommendation as classification: using social and content-based information in recommendation. In *Proc. of the 15th National Conf. on Artificial Intelligence (AAAI 1998)*. AAAI Press, Menlo Park, CA, USA, 714–720.

[3] A. Bellogín, P. Castells, and I. Cantador. 2011. Precision-Oriented Evaluation of Recommender Systems: An Algorithmic Comparison. In *Proc. of the 5th ACM Conf. on Recommender Systems (RecSys 2011)*. ACM, New York, NY, USA, 333–336.

[4] A. Bellogín, P. Castells, and I. Cantador. 2017. Statistical biases in information retrieval metrics for recommender systems. *Inf. Ret.* 20, 6 (July 2017), 606–634.

[5] C. Buckley and E. M. Voorhees. 2004. Retrieval Evaluation with Incomplete Information. In *Proc. of the 27th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR 2004)*. ACM, New York, NY, USA, 25–32.

[6] R. Cañamares and P. Castells. 2017. A probabilistic reformulation of memory-based collaborative filtering: Implications on popularity biases. In *Proc. of the 40th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR 2017)*. ACM, New York, NY, USA, 215–224.

[7] R. Cañamares and P. Castells. 2018. Should I follow the crowd? A probabilistic analysis of the effectiveness of popularity in recommender systems. In *Proc. of the 41st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR 2018)*. ACM, New York, NY, USA, 415–424.

[8] R. Cañamares, P. Castells, and A. Moffat. 2020. Offline Evaluation Options for Recommender Systems. *Inf. Ret.* 23, 4 (Aug. 2020), 387–411.

[9] B. A. Carterette. 2012. Multiple Testing in Statistical Analysis of Systems-Based Information Retrieval Experiments. *ACM Transactions on Information Systems* 30, 1 (March 2012).

[10] P. Castells and R. Cañamares. 2018. Characterization of fair experiments for recommender system evaluation: A formal analysis. In *Proc. of the Workshop on Offline Evaluation for Recommender Systems (REVEAL 2018) at the 12th ACM Conf. on Recommender Systems (RecSys 2018)*.

[11] W. Cheng, Y. Shen, Y. Zhu, and L. Huang. 2018. DELF: A dual-embedding based deep latent factor model for recommendation. In *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI 2018)*. Morgan Kaufmann, Burlington, MA, USA, 3329–3335.

[12] P. Cremonesi, Y. Koren, and R. Turrin. 2010. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proc. of the 4th ACM Conf. on Recommender Systems (RecSys 2010)*. ACM, New York, NY, USA, 39–46.

[13] M. Deshpande and G. Karypis. 2004. Item-Based Top-N Recommendation Algorithms. *ACM Transactions on Information Systems* 22, 1 (Jan. 2004), 143–177.

[14] T. Ebesu, B. Shen, and Y. Fang. 2018. Collaborative memory network for recommendation systems. In *Proc. of the 41st Annual Int. ACM SIGIR Conf. on Research and Development in Inf. Ret. (SIGIR 2018)*. ACM, New York, NY, USA, 515–524.

[15] A. Gruson, P. Chandar, C. Charbuillet, J. McInerney, S. Hansen, D. Tardieu, and B. Carterette. 2019. Offline Evaluation to Make Decisions About PlaylistRecommendation Algorithms. In *Proc. of the 12th ACM Int. Conf. on Web Search and Data Mining (WSDM 2019)*. ACM, New York, NY, USA, 420–428.

[16] A. Gunawardana and G. Shani. 2015. Evaluating Recommender Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, Boston, MA, USA, 265–308.

[17] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T.-S. Chua. 2018. Outer product-based neural collaborative filtering. In *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI 2018)*. Morgan Kaufmann, Burlington, MA, USA, 2227–2233.

[18] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. 2017. Neural collaborative filtering. In *Proc. of the 26th Int. Conf. on World Wide Web (WWW 2017)*. ACM, New York, NY, USA, 173–182.

[19] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. 1999. An Algorithmic Framework for Performing Collaborative Filtering. In *Proc. of the 22nd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR 1999)*. ACM, New York, NY, USA, 230–237.

[20] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems* 22, 1 (Jan. 2004), 5–53.

[21] K. Hofmann, L. Li, and F. Radlinski. 2016. Online Evaluation for Information Retrieval. *Found. Trends Inf. Retr.* 10, 1 (June 2016), 1–117.

[22] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proc. of the 24th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2018)*. ACM, New York, NY, USA, 1531–1540.

[23] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proc. of the 8th IEEE Int. Conf. on Data Mining (ICDM 2008)*. IEEE Computer Society, Washington, DC, USA, 15–19.

[24] D. Jannach, L. Lerche, I. Kamehkhosh, and M. Jugovac. 2015. What Recommenders Recommend: an Analysis of Recommendation Biases and Possible Countermeasures. *User Modeling and User-Adapted Interaction* 25, 5 (Dec. 2015), 427–491.

[25] Y. Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proc. of the 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2008)*. ACM, New York, NY, USA, 426–434.

[26] Y. Koren. 2009. Collaborative Filtering with Temporal Dynamics. In *Proc. of the 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2009)*. ACM, New York, NY, USA, 447–456.

[27] W. Krichene and S. Rendle. 2020. On Sampled Metrics for Item Recommendation. In *Proc. of the 26th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2020)*. ACM, New York, NY, USA, 1748–1757.

[28] B. M. Marlin and R. S. Zemel. 2009. Collaborative Prediction and Ranking with Non-random Missing Data. In *Proc. of the 3rd ACM Conf. on Recommender Systems (RecSys 2009)*. ACM, New York, NY, USA, 5–12.

[29] F. M. Maxwell and J. A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (Dec. 2015).

[30] M. McGee. 2018. Case for omitting tied observations in the two-sample t-test and the Wilcoxon-Mann-Whitney Test. *PLOS ONE* 13, 7 (July 2018), 1–19.

[31] R. J. Mooney and L. Roy. 1999. Content-Based Book Recommending Using Learning for Text Categorization. In *Proc. of the 5th ACM Conf. on Digital Libraries*. ACM, New York, NY, USA, 195–204.

[32] M. Rossetti, F. Stella, and M. Zanker. 2016. Contrasting Offline and Online Results When Evaluating Recommendation Algorithms. In *Proc. of the 10th ACM Conf. on Recommender Systems (RecSys 2016)*. ACM, New York, NY, USA, 31–34.

[33] T. Sakai. 2006. Evaluating evaluation metrics based on the bootstrap. In *Proc. of the 29th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR 2006)*. ACM, New York, NY, USA, 525–532.

[34] T. Sakai. 2007. Alternatives to Bpref. In *Proc. of the 30th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR 2007)*. ACM, New York, NY, USA, 71–78.

[35] T. Sakai. 2018. *Laboratory Experiments in Information Retrieval - Sample Sizes, Effect Sizes, and Statistical Power*. The Information Retrieval Series, Vol. 40. Springer.

[36] T. Sakai and N. Kando. 2008. On Information Retrieval Metrics Designed for Evaluation with Incomplete Relevance Assessments. *Inf. Ret.* 11, 5 (March 2008), 447–470.

[37] H. Steck. 2010. Training and Testing of Recommender Systems on Data Missing not at Random. In *Proc. of the 16th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2010)*. ACM, New York, NY, USA, 713–722.

[38] H. Steck. 2011. Item Popularity and Recommendation Accuracy. In *Proc. of the 5th ACM Conf. on Recommender Systems (RecSys 2011)*. ACM, New York, NY, USA, 125–132.

[39] H. Steck. 2013. Evaluation of recommendations: rating prediction and ranking. In *Proc. of the 7th ACM Conf. on Recommender Systems (RecSys 2013)*. ACM, New York, NY, USA, 213–220.

[40] D. Valcarce, A. Bellogín, J. Parapar, and P. Castells. 2020. Assessing ranking metrics in top-N recommendation. *Inf. Ret.* 23, 4 (June 2020), 411–448.

[41] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen. 2017. Deep matrix factorization models for recommender systems. In *Proc. of the 26th Int. Joint Conf. on Artificial Intelligence (IJCAI 2017)*. Morgan Kaufmann, Burlington, MA, USA, 3203–3209.

[42] E. Yilmaz and J. A. Aslam. 2006. Estimating average precision with incomplete and imperfect judgments. In *Proc. of the 15th ACM Int. Conf. on Information and Knowledge Management (CIKM 2006)*. ACM, New York, NY, USA, 102–111.

[43] E. Yilmaz and J. A. Aslam. 2008. Estimating Average Precision when Judgments are Incomplete. *Knowledge and Information Systems* 16, 2 (Aug. 2008), 173–211.

[44] C. Zhai and J. D. Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems* 22, 2 (April 2004), 179–14.

[45] Q. Zhang, L. Cao, C. Zhu, Z. Li, and J. Sun. 2018. CoupledCF: Learning explicit and implicit user-item couplings in recommendation for deep collaborative filtering. In *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI 2018)*. Morgan Kaufmann, Burlington, MA, USA, 3662–3668.