On the Robustness and Discriminative Power of Information Retrieval Metrics for Top-N Recommendation

Daniel Valcarce
daniel.valcarce@udc.es
Information Retrieval Lab
University of A Coruña
A Coruña, Spain

Alejandro Bellogín
alejandro.bellogin@uam.es
Information Retrieval Group
Universidad Autóma de Madrid
Madrid, Spain

Javier Parapar
javierparapar@udc.es
Information Retrieval Lab
University of A Coruña
A Coruña, Spain

Pablo Castells
pablo.castells@uam.es
Information Retrieval Group
Universidad Autóma de Madrid
Madrid, Spain

ABSTRACT

The evaluation of Recommender Systems is still an open issue in the field. Despite its limitations, offline evaluation usually constitutes the first step in assessing recommendation methods due to its reduced costs and high reproducibility. Selecting the appropriate metric is a critical and ranking accuracy usually attracts the most attention nowadays. In this paper, we aim to shed light on the advantages of different ranking metrics which were previously used in Information Retrieval and are now used for assessing top-N recommenders. We propose methodologies for comparing the robustness and the discriminative power of different metrics. On the one hand, we study cut-offs and we find that deeper cut-offs offer greater robustness and discriminative power. On the other hand, we find that precision offers high robustness and Normalised Discounted Cumulative Gain provides the best discriminative power.

1 INTRODUCTION

Recommender Systems help users find their way in massive information spaces. In a world of information overload, these systems help users in finding relevant pieces of information [31]. The objective of a recommender system is to suggest items that may be of interest to the users. Although the idea is simple to explain, the goodness of a solution to the task is difficult to measure. What constitutes a good recommendation? How should recommender systems be evaluated?

Traditionally, recommender systems were defined as rating predictors. Their aim was to forecast the ratings that users would give to each item [18, 19]. Therefore, for this rating prediction task, the evaluation was based on error metrics such as RMSE (Root Mean Squared Error) or MAE (Mean Absolute Error) [18]. The rationale was that if a model is able to effectively predict ratings, we can use it for recommending those items with the highest predicted ratings. Nonetheless, it has been acknowledged that the assessment of recommendation methods based on error metrics does not lead to a good evaluation [18, 19, 27]. When deploying a recommender system in production, the common task is to provide a short list of good suggestions where the predicted rating values are not shown [19]. This task is usually referred to as top-N recommendation [14]. Therefore, the focus should be on providing a list with good items, not on accurately predicting any ratings. Moreover, rating prediction studies how well a system can predict the rating for any item while a top-N recommender focuses on the top relevant items for each user. For all these reasons, this paradigm shift has been brought to recommendation.

Although there is an increasing interest in assessing different recommendation properties (such as diversity and novelty [11]), accuracy remains the primary objective property of recommendation [18, 19]. For a given user, we can say that a particular recommendation is "correct" if that user likes the recommended item. To evaluate recommendations, we can conduct online or offline experiments. Online evaluations consist in asking real users about the quality of the recommendations they received. However, this is an expensive procedure. For this reason, as a first step, it is common to perform offline evaluations. This type of assessment procedure exploits a dataset of previously collected user-item interactions [18]. This dataset is usually divided into two splits: the training split is used as input to the recommendation algorithm and the test split is employed for measuring the performance of the recommender system using different evaluation metrics. An extra validation split for tuning the hyperparameters can also be used.

Offline evaluation is well established in the Information Retrieval (IR) field by the Cranfield paradigm and the TREC initiative [42]. IR and Recommender Systems (RS) are strongly related fields, where both seek to provide relevant pieces of information to the users [2]. The central difference is the representation of the information need: while an IR system typically uses an explicit query prompted by the user, a RS exploits the user's data as an implicit query. The Cranfield paradigm measures how a retrieval system meets the information needs of the users using ranking-oriented metrics. Some of these metrics have also been used for assessing RS in the top-N recommendation task. However, in contrast to IR, the evaluation of RS is a disputed issue. Since RS lack proper relevance judgements, researchers use a hold-out data from the collection to assess the quality of the recommendations. These judgements are incomplete and obtained in a very different way compared to the

RecSys '18, October 2–7, 2018, Vancouver, BC, Canada
This document is a draft, the DOI link points to the final version. https://doi.org/10.1145/3240323.3240347

IR relevance judgements. Since the assumptions of the Cranfield paradigm are substantially different from those of the recommender evaluation, **should IR metrics be applied to RS?** Although most of these metrics are already being used in RS evaluation, they have not been thoroughly studied in this field. A metric is robust when shows the same behaviour when less relevance judgements are available. Likewise, a metric is discriminative when changes in its values are statistically significant. For this reason, we study the robustness and discriminative power of several ranking-oriented IR metrics to top-N recommendation in order to answer that question.

Our experiments on three datasets show that considering more recommendations than the top ones (i.e., using deep metric cut-offs) improves the robustness and discriminative power of the evaluation. Additionally, we see that precision offers the best robustness figures whereas normalised Discounted Cumulative Gain has the highest discriminative power. Therefore, these metrics should be preferred when evaluating top-N recommenders.

2 RELATED WORK AND BACKGROUND

Evaluation plays a crucial role in IR and RS: the effectiveness of any retrieval or recommender system needs always to be measured empirically. IR has established the Cranfield paradigm as the standard evaluation methodology [42], but this is not the case in top-N recommendation where several different approaches coexist.

Cranfield paradigm. It is a well-founded evaluation methodology in IR based on the use of test collections which contain documents, topics and relevance judgements for each topic [42]. Assessors judge the documents to indicate which ones are relevant to a given topic. With these relevance judgements, we can evaluate the output of a retrieval system using ranking-oriented metrics. However, the Cranfield paradigm has three fundamental assumptions: i) the information need of the user—specified by a topic— can be approximated by topical similarity, ii) relevance is independent of the users which implies that a set of relevance judgements is valid for any user and iii) completeness of judgements, i.e., all the relevant documents for a topic are known. Although these assumptions are not generally true, they are reasonable and some deficiencies can be compensated [42]. Therefore, this paradigm is the standard systematic approach to the evaluation of retrieval systems. The evaluation for a topic consists in generating a list of documents sorted by decreasing relevance according to each retrieval model. Ranking-oriented metrics evaluate these rankings using the relevance judgements for that topic. The quality of a retrieval strategy is measured as the average metric score for all topics.

The main problem of this approach is that the volume of information in modern test collections is too large to have complete relevance judgements. For this reason, a process called pooling is conducted to select the documents whose relevance should be assessed by humans [35, 42]. Documents that do not appear in the pool are assumed to be non-relevant. The idea is that we should make relative evaluations (not absolute) with the test collections. To ensure this, relevance judgements should be unbiased. Pooling, if performed correctly, may be a good enough approximation [42]. However, large-scale datasets such as ClueWeb contain hundreds of millions of documents which are shallow pooled resulting in many potentially relevant documents unjudged.

The limitations and biases of the Cranfield paradigm have been extensively studied. There have been efforts to overcome the bias produced by pooling [7, 10]. Also, Buckley and Voorhees studied how the number of relevance judgements affects different precision-oriented metrics [9]. They defined the robustness of a metric with respect to incomplete judgements as how well the metric correlates with itself when the relevance judgements are incomplete. When using incomplete judgements, bpref correlated with itself with all judgements and with AP better than other standard IR metrics. They also found that bpref preserves the absolute scores and the relative ranking of systems better than MAP or precision. Yilmaz and Aslam later proposed three estimates of AP for the incomplete judgements scenario [44]. Their proposals showed a better correlation between themselves and AP than bpref. These correlations between system rankings were measured in terms of Kendall's correlation [23]. Among the three proposals, infAP was the metric that provided the best results [44]. To measure the robustness to incomplete judgements in these experiments, the metrics were calculated using random subsets of relevance judgements. Buckley and Voorhees used stratified random sampling [9] while Yilmaz and Aslam employed random sampling [44]. However, both samplings are identical in expectation [44]. Additionally, Lu et al. thoroughly studied the effect of the pooling depth in several IR metrics providing a list of advices for IR evaluation [24].

Besides robustness to incompleteness, discriminative power is another property of evaluation metrics that has been thoroughly studied in IR [8, 24, 32, 33] and measures the capability of a metric to discriminate among systems. We should note that the discriminative power not only depends on the metric but also on the test collection and the set of systems being compared. Buckley and Voorhees proposed a first attempt to study the discriminative power of a metric using a *fuzziness value* [8]. Later, Sakai introduced a more formal method based on the bootstrap test [32]. Given a significance level (e.g., $p = 0.05$), he computed the ratio of system pairs for which a statistical test finds a significant difference. In particular, Sakai employed the bootstrap test with Student's t statistic for this purpose [32]. To avoid fixing a particular significance level, Lu et al. proposed to report the median system-pair p-value as a measure of discriminative power [24]. Sakai also studied how incomplete judgements affect the discriminative power in IR [33].

RS evaluation. Although top-N recommendation is now the standard recommendation task in RS (in contrast to rating prediction), there are still several controversial issues regarding evaluation [14, 18, 19] such as the debate between offline and online experiments. Recent studies restricted to particular domains have shown discrepancies between CTR and offline metrics [1, 17]. Also, a more exhaustive study analysed seven different recommendation algorithms from a user-centric perspective using two accuracy metrics [12] finding a poor matching between the perceived quality and recall and fall-out metrics. In contrast, a posterior study in the e-tourism domain showed that recall and fallout are a good approximation of the quality perceived by the users [13]. Overall, online evaluation depends on several variables such as the domain, the demographics of the users or the display of recommendations. This complicates the unbiased evaluation of recommender algorithms. Additionally, reproducibility is difficult, if not impossible, to achieve when researchers do not have access to the original experimental environment. For all these reasons, offline experimentation has its place in the evaluation of recommender systems and usually constitutes the first step in assessing the performance of recommendation algorithms. Online assessments should be conducted in an industrial scenario, but offline evaluation is also valuable as a way of having an objective and preliminary comparison.

Recommender Systems test collections do not rely on pooling. Instead, they employ a fraction of the data from the dataset (such as ratings or clicks) for test purposes and the rest for building the recommendation model. However, this does not mean that RS evaluation is free from biases. In fact, Bellogín et al. showed that sparsity and popularity biases impact the evaluation of RS [4]. Also, metrics such as bpref and infAP which have been proposed in IR to address incompleteness of relevance judgements [9, 44] have rarely been used in recommendation [5, 26, 39]. To the best of our knowledge, there has not been a systematic review of metrics in RS regarding robustness to incompleteness and discriminative power.

3 IR METRICS FOR RECOMMENDATION

Cranfield paradigm can be adapted to RS evaluation in the following manner: the users play the role of queries (since they both are associated with an information need) and we need to evaluate item rankings instead of document rankings. Cranfield evaluation makes use of relevance judgements. In recommendation, we lack those judgements and we approximate them with hold-out data from the user. However, when evaluating RS, some Cranfield assumptions do not hold (see Table 1 for a summary). In particular, relevance is highly dependent on the users: the same item may not be relevant to two different people. Moreover, relevance judgements are far from complete. Since relevance is personal, we cannot build a set of relevance judgements using a group of experts. Furthermore, since we build the test dataset with hold-out data, the incompleteness of the relevance judgements is intrinsic to the recommendation task.

Additionally, whereas MAP has been considered a reference metric in IR (in spite of recent criticism [16]), RS lack consensus about which metric is the most reliable to measure the ranking quality of recommendations. In addition, when approximating relevance judgements with a hold-out test set, how much data is used for the training and test splits should be balanced. A larger training subset (at the expense of a reduced test subset) will allow better modelling, but it would provide worse evaluation reliability and vice versa. Finally, the long tail distribution of items in RS systems impacts the recommendation process. In contrast, IR evaluation does not have to deal with such a great imbalance in the popularity of documents.

Regarding user relevance in RS, all the items rated by the target user u in the test set with a value below a certain relevance threshold τ are considered non-relevant items and form the set $\mathcal{N}_u$. Likewise, $\mathcal{R}_u$ represents the set of relevant items for user u, i.e., those items rated by u in the test set with a score greater than or equal to the threshold τ. In a dataset with ratings ranging from 1 to 5, it is common to set τ to 4. Those items that the target user did not rate are considered unjudged (their relevance is unknown). Most IR ranking metrics ignore unjudged elements and treat them as non-relevant, but some metrics explicitly consider them separately.

Finally, in the top-N recommendation task, we seek to generate a ranking of the N most relevant items for a given user [14]. Therefore, these systems deal with a set of users $\mathcal{U}$ and a set of items $\mathcal{I}$. We represent the ranking of length n for user u by the list L_u^n. We refer to the access to the k-th position of that list by $L_u^n[k]$. We denote the rating from a user u to an item i by $r(u,i)$.

Table 1: Comparison between Information Retrieval and Recommender Systems evaluation assumptions.

Information Retrieval	Recommender Systems
Topical similarity can approximate the user's information need.	User's information need may be estimated in several different ways.
Relevance is independent on users.	Relevance is dependent on users.
Relevance judgements are almost complete (pooling depth).	Relevance judgements are far from complete.

Next, we present common IR metrics particularised to RS evaluation. All the following metrics range from 0 to 1 where the higher the value, the better. Also, these metrics are computed on a per-user basis (denoted here with the subscript u). To obtain the final value, we average the metric over all the users. If a RS cannot provide recommendations for a particular user, we assign a value of zero to all metrics for that user to penalise low user coverage (i.e., not being able to provide recommendations to some users). These metrics evaluate the quality of a recommendation ranking. It is common to truncate the ranking at position n (this is the cut-off and is represented by @n at the end of the metric name). Since some metrics have multiple versions with slight differences, in this work we follow `trec_eval`[1] implementation of the metrics which is the standard evaluation tool of the TREC initiative.

Precision (P). Precision measures how well a method puts relevant items in the first n recommendations regardless the rank:

$$P_u@n = \left|L_u^n \cap \mathcal{R}_u\right| / \, n \tag{1}$$

Recall. Recall measures the proportion of relevant items that are included in the recommendation list with respect to the total number of relevant items for a given user:

$$Recall_u@n = \left|L_u^n \cap \mathcal{R}_u\right| / \left|\mathcal{R}_u\right| \tag{2}$$

Average Precision (AP). Average Precision averages precision at the positions where a relevant item is found. $rel(L_u^n[k])$ indicates if the item at the k-th position of the ranking of size n for user u is relevant. When AP is averaged over the set of topics (users in our recommendation scenario) receives the name of mean AP (MAP).

$$AP_u@n = \frac{1}{|\mathcal{R}_u|} \sum_{k=1}^{n} rel\left(L_u^n[k]\right)\; P_u@k \tag{3}$$

Normalised Discounted Cumulative Gain (nDCG). This metric uses graded relevance (the values of the ratings) as well as positional information of the recommended items [22]. Let $D(i)$ be a discounting function, $G(u,n,k)$ be the gain we obtain by recommending item $L_u^n[k]$ to user u and let $G^*(u,n,k)$ be the gain associated to the k-th element in the ideal ranking of size n for the user u (where items are ranked in decreasing order of gain):

$$nDCG_u@n = \frac{\sum_{k=1}^{n} G(u,n,k)\, D(k)}{\sum_{k=1}^{n} G^*(u,n,k)\, D(k)} \tag{4}$$

A common discount function is $D(k) = \log_2^{-1}(k+1)$. Although there exist multiple options for defining the gain function, our

[1] https://github.com/usnistgov/trec_eval

preliminary experiments showed no meaningful differences among them. Therefore, we decided to use simply $G(u, n, k) = r(u, L_u^n[k])$ as the gain function hereinafter.

Reciprocal Rank (RR). It is computed as the inverse of the position of the first relevant element in the ranking. As AP, when averaged over a set of topics, this metric is called Mean RR (MRR).

$$RR_u = 1 \,/\, \min_k \left\{ rel\left(L_u^n[k]\right) > 0 \right\} \tag{5}$$

Bpref. This metric was designed to be highly correlated with AP but more robust to incomplete relevance judgements [9]. Bpref is inversely related to the number of judged non-relevant items that are located above each relevant item in the ranking list:

$$bpref_u@n = \frac{1}{|\mathcal{R}_u|} \sum_{k=1}^{n} rel\left(L_u^n[k]\right) \left(1 - \frac{\min(|L_u^k \cap \mathcal{N}_u|, |\mathcal{R}_u|)}{\min(|\mathcal{N}_u|, |\mathcal{R}_u|)}\right) \tag{6}$$

Inferred Average Precision (InfAP). InfAP yields the same score MAP provides when the relevance judgements are complete; however, it is also a statistical estimate of MAP when using incomplete judgements [44]. InfAP has shown a better correlation with AP than bpref under this scenario. This metric is given by:

$$infAP_u@n = \frac{1}{|\mathcal{R}_u|} \sum_{k=1}^{n} rel\left(L_u^n[k]\right) E[P@k] \tag{7}$$

where the expected precision at position k is defined as:

$$E[P@k] = \frac{1}{k} + \frac{k-1}{k} \frac{|L_u^{k-1} \cap \mathcal{R}_u| + \varepsilon}{|L_u^{k-1} \cap \mathcal{R}_u| + |L_u^{k-1} \cap \mathcal{N}_u| + 2\varepsilon} \tag{8}$$

and ε is a small constant (we set ε to 0.00001 in our experiments).

4 METHODOLOGIES TO EVALUATE IR METRICS IN TOP-N RECOMMENDATION

In this section, we propose methodologies to analyse the robustness to incompleteness and the discriminative power of the aforementioned metrics in the evaluation of top-N recommenders based on previous studies of these properties in IR. We start with the analysis on robustness to incompleteness because relevance judgements are very scarce in the recommendation scenarios and, thus, it is difficult to make a reliable assessment of recommenders. Moreover, when preferring one recommendation model over another, we need to have statistically sound guarantees—the discriminative power of a metric measure this desirable property.

Robustness to Incompleteness. When evaluating recommenders systems, incompleteness is pervasive. The ratings in the test set form the relevance judgements which are incomplete (in fact, this is an intrinsic property of the recommendation task). A desirable metric for recommendation should be robust to incompleteness in the test set. We can simulate incompleteness in an IR scenario using unbiased random sampling techniques [9, 44]. We follow a similar approach to induce incompleteness when evaluating recommender systems. However, previous work on RS evaluation has found two types of incompleteness in the recommendation task [4]. When we use IR metrics to assess recommender systems, two well differentiated biases arise: the sparsity bias and the popularity bias. For this reason, next, we analyse the robustness to the sparsity bias and the robustness to the popularity bias independently.

Sparsity bias The sparsity bias arises in RS evaluation when we lack known relevance for all the user-items pairs [4]. In recommendation, users' profiles are incomplete by definition: we build the test set as a hold-out subset of the users' profile. Moreover, in a scenario without incompleteness, we will be unable to recommend anything because nothing unknown would be available to suggest. Note that the sparsity bias causes the absolute values of the metrics to lose meaning, but the relative values can still be valid for comparative purposes [4].

We propose to measure the robustness of different metrics to the sparsity bias by evaluating those metrics using random samples of the test set. We create test sets by removing relevance judgements randomly. For each test set size, we can take different random samples. Given a set of recommenders, we evaluate them according to a particular metric and compute the ranking of systems. Then, we measure Kendall's correlation of this ranking with respect to the ranking obtained by evaluating those systems using the original test set. Finally, by averaging the rank correlation for each sample of the same size, we obtain a final estimate of the robustness of a metric for each test size. A highly robust metric will yield higher average correlation values.

Popularity bias In contrast to IR, missing relevance judgements are not uniformly distributed (this has been referred to as *missing not at random* [25, 36]). The distribution of ratings in a recommendation scenario follows a heavy skewed long-tail distribution. Bellogín et al. studied this popularity bias and found that it strongly affects the reliability of several IR metrics [4].

Since previous works on recommender systems remove popular items to deal with the popularity bias [4, 14], we propose to build progressively smaller test sets removing ratings from the most popular items to measure the robustness of a metric to this bias. Then, we can study the change in the correlation between systems rankings of different subsets of the test set and the original test set. The higher the value of the correlation, the higher the robustness of such metric to the popularity bias.

Discriminative Power. When we compare two recommendation techniques, we expect that the variation in the values of a metric to indicate a statistically significant difference. Otherwise, if the difference is not significant, we would not be able to conclude anything with that metric. We propose to measure the discriminative power of several IR metrics on different datasets. We follow a procedure similar to the method presented by Sakai [32], but we change the statistical test. Instead of using Bootstrap with the Student's t statistic, we can employ the permutation test (also known as Fisher's randomisation test) with the difference in means as test statistic [15]. The permutation test provides a better estimation of the p-value [34]. Since computing the exact p-value requires the computation of 2^n permutations (where n is the number of test users), we can approximate the result of this test using Monte Carlo sampling. With 100,000 samples, we can compute a two-sided p-value of 0.05 with an estimated error of ±0.001 and a p-value of 0.01 with an error of ±0.00045 [15, 34].

For each metric, we may plot the p-values of the statistical test between all possible system pairs sorted by decreasing value as in [32]. We call each of those curves the p-value curve of a metric. Since a highly discriminative metric will yield low p-values, we would then prefer metrics with p-value curves close to the origin. Furthermore, we also want to compute a value that summarises the discriminative power of a metric. For this purpose, we use the sum of the p-values between all system pairs as an approximation of the area under the p-value curve. We call this value DP (discriminative

Table 2: Datasets statistics

Dataset	Users	Items	Ratings	Density	Gini
MovieLens 1M	6,040	3,706	1,000,209	4.468%	0.634
LibraryThing	7,279	37,232	749,401	0.277%	0.581
BeerAdvocate	33,388	66,055	1,571,808	0.071%	0.865

power). The lower the value of DP, the higher the discriminative power of the metric. Note that DP has a use for comparing metrics when using the same set of systems on the same dataset.

5 EXPERIMENTAL SETTINGS

In this section, we describe the experimental settings. We present the employed datasets and the training-test splitting strategy. Then, we explain the details of the followed evaluation methodology. Last, we provide a brief description of the RS used in the experiments.

Datasets. We used three collections with explicit feedback in form of 1-5 ratings: MovieLens 1M[2], LibraryThing and BeerAdvocate[3]. Table 2 indicates the number of users, items and ratings, as well as the density (percentage of user-item pairs that have a rating) and the Gini coefficient for measuring the long tail (i.e., the inequality in the distribution of ratings across items) of the datasets.

We created the training and test splits of the datasets by taking 80% of the ratings of each user for the training set and the remaining data is used as test set. We avoid further biases by having the same proportion of training and test data for each user.

Evaluation methodology. Several protocols for offline evaluation in Recommender Systems have been proposed [3, 19]. We decided to follow the AllItems approach which has been regarded as a fair evaluation methodology and is similar to how systems are evaluated in IR (where no hold-out test set is available) [3]: for each user u, we rank all items in the dataset that have not been rated by u in the training set. This methodology consists in ranking all items in the test set except those already rated by the target user in the training set. In this way, an ideal recommender system will be able to achieve a perfect score in all the studied metrics. Note that this evaluation procedure is highly correlated to other variants [3].

For assessing the robustness to incompleteness, on the one hand, we use the methodology for studying the sparsity bias (presented in Sec. 4) sweeping from samples with 100% of the ratings of the original test set to samples with 5% of the ratings in steps of 5% to simulate the sparsity bias. We compute the average of 50 samples of each test set size which provides a good estimate in our experiments. On the other hand, when using the methodology for analysing the popularity bias (see Sec. 4), we start from using the ratings of 100% of items to using only the ratings of the 80% least popular items in steps of 1% to simulate the popularity bias.

Additionally, many IR metrics are based on binary relevance: each item is either relevant or non-relevant for a given user. In this work, since we focused on explicit feedback datasets, we have to specify how to transform the ratings (a form of incomplete graded relevance) to binary relevance. For this purpose, we set the relevance threshold τ to 4, i.e., we consider non-relevant every item rated below τ. Those items that are not rated by the target user in the test set are neither relevant nor non-relevant—they are equivalent to the unjudged documents in the Cranfield paradigm.

Recommender Systems. When examining metrics, we need systems to compare. Previous works in IR studying different metrics employed the runs submitted to TREC [9, 24, 44]. Since we do not have an equivalent in RS, we implemented 21 recommendations techniques and used their outputs to study the properties of several IR metrics[4]. Note that we have chosen multiple types of algorithms to have a representative set of recommendation techniques.

- **Random, Popularity**: basic baselines.
- **CHI2**, **KLD**, **RSV**, **Rocchio's Weights** [38]: neighbourhood-based techniques that stem from Rocchio's feedback model.
- **RM1**, **RM2** [29]: neighbourhood-based techniques that use relevance-based language models.
- **LM-WSR-UB**, **LM-WSR-IB** [40]: user-based and item-based approaches that with language models for neighbourhoods.
- **NNCosNgbr-UB**, **NNCosNgbr-IB** [14]: user-based and item-based versions of a neighbourhood technique.
- **SLIM** [28]: sparse linear methods.
- **HT** [45]: graph-based technique
- **SVD, PureSVD, BPRMF, WRMF** [14, 21, 30, 37]: matrix factorization techniques.
- **LDA** [6]: Latent Dirichlet Allocation.
- **PLSA** [20]: Probabilistic Latent Semantic Analysis.
- **UIR-Item** [43]: probabilistic user-item relevance model.

6 CHOOSING AMONG CUT-OFFS

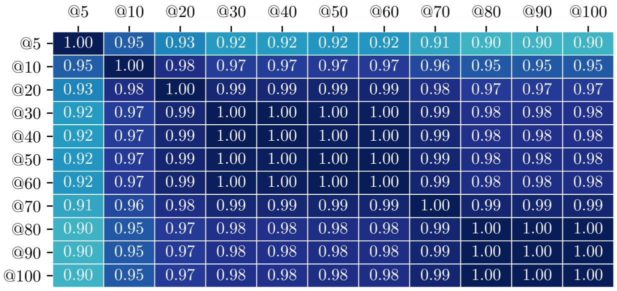
When applying a ranking metric, we have to select the cut-off. Recommenders usually show only a few suggested items because users seldom consider more than the top ones. For this reason, recommender systems literature usually employ shallow cut-offs such as 5 or 10 [18]. However, the selection of the exact value of the cut-off in some research papers is somewhat arbitrary. Although RS typically present few recommendations to their users, deeper cut-offs may provide a more reliable assessment of the recommenders offline evaluation. Therefore, next, we analyse which cut-offs are preferable regarding robustness and discriminative power.

Correlation Among Cut-offs. We study Kendall's correlation between systems when using the same metric with different cut-offs. We find high correlations between rankings when studying cut-offs from 5 to 100. Overall, the correlation between cut-offs above 20 is very high. Those correlations are almost always higher than 0.9 on the LibraryThing and BeerAdvocate datasets. Note that previous work has considered that two rankings with a correlation above 0.9 are almost equivalent [41]. On the MovieLens dataset, most of the correlations are above 0.85 and the lowest found correlation was between $P@5$ and $P@100$ and $Recall@5$ and $Recall@100$ with a value of 0.76. For the sake of space, we choose a representative example: Fig. 1 shows the correlation between different cut-offs of nDCG on MovieLens 1M. The largest discrepancy is between the cut-off at 5 and the rest of cut-offs. However, all the correlations are at least 0.9 which represents a very strong correlation. Therefore, we can conclude from this experiment that the choice of the cut-off

[2] https://grouplens.org/datasets/movielens

[3] http://snap.stanford.edu/data/web-BeerAdvocate.html

[4] We provide the source code of the experiments and the complete output of the recommender systems at http://www.dc.fi.udc.es/~dvalcarce/metrics.html.



	@5	@10	@20	@30	@40	@50	@60	@70	@80	@90	@100
@5	1.00	0.95	0.93	0.92	0.92	0.92	0.92	0.91	0.90	0.90	0.90
@10	0.95	1.00	0.98	0.97	0.97	0.97	0.97	0.96	0.95	0.95	0.95
@20	0.93	0.98	1.00	0.99	0.99	0.99	0.99	0.98	0.97	0.97	0.97
@30	0.92	0.97	0.99	1.00	1.00	1.00	1.00	0.99	0.98	0.98	0.98
@40	0.92	0.97	0.99	1.00	1.00	1.00	1.00	0.99	0.98	0.98	0.98
@50	0.92	0.97	0.99	1.00	1.00	1.00	1.00	0.99	0.98	0.98	0.98
@60	0.92	0.97	0.99	1.00	1.00	1.00	1.00	0.99	0.98	0.98	0.98
@70	0.91	0.96	0.98	0.99	0.99	0.99	0.99	1.00	0.99	0.99	0.99
@80	0.90	0.95	0.97	0.98	0.98	0.98	0.98	0.99	1.00	1.00	1.00
@90	0.90	0.95	0.97	0.98	0.98	0.98	0.98	0.99	1.00	1.00	1.00
@100	0.90	0.95	0.97	0.98	0.98	0.98	0.98	0.99	1.00	1.00	1.00

Figure 1: Correlation between different cut-offs of nDCG metric on the MovieLens 1M dataset.

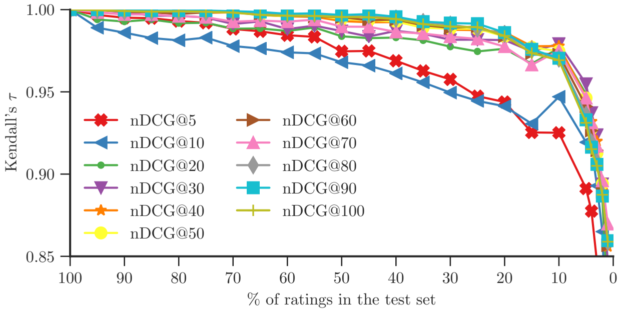


Figure 2: Kendall's correlation of different cut-offs of nDCG with respect to themselves using the full test set when increasing the sparsity bias on the MovieLens 1M dataset.

does not affect the ranking of the systems severely; however, it may affect the robustness or the discriminative power.

Robustness Among Cut-offs. We test the robustness to sparsity and popularity of different cut-offs from 5 to 100 of each metric following the procedure explained in Sec. 4. The results confirm that larger cut-offs yield better figures of robustness when increasing the sparsity and the popularity bias of the test set. As a representative example, Fig. 2 plots the robustness to the sparsity bias of different cut-offs of nDCG on MovieLens 1M. Likewise, Fig. 3 plots the robustness to the popularity bias of different cut-offs of nDCG. In both figures, we can see that robustness increases as we use deeper cut-offs. This phenomenon also occurs in the other studied metrics on the three datasets with slight variations. We omit them due to lack of space.

Discriminative Power Among Cut-offs. We study the discriminative power (DP) of each metric using cut-offs from 5 to 100. Using the procedure described in Sec. 4, we plot the *p*-values of the paired statistical tests sorted by decreasing value on the MovieLens 1M (see Fig. 4). We observe that deeper cut-offs (above 50) consistently provide better figures of DP than shallower cut-offs. Different metrics on the three datasets present similar results.

Implications. In light of these results, we can conclude that the studied metrics with **deeper cut-offs are more robust to the sparsity and popularity biases and have better discriminative power**. Additionally, since the ranking of systems produced by a metric when varying the cut-off from 5 to 100 does not change notably, we should prefer deeper cut-offs. Therefore, if there is no strong reason to choose a shallow cut-off such as 5 or 10, calculating the metric over a larger ranking (let say $n = 100$ recommendations) should be preferred in offline experiments. Note that such deep cut-off provides better properties even though we may (and generally will) lack n relevance judgements for each user.

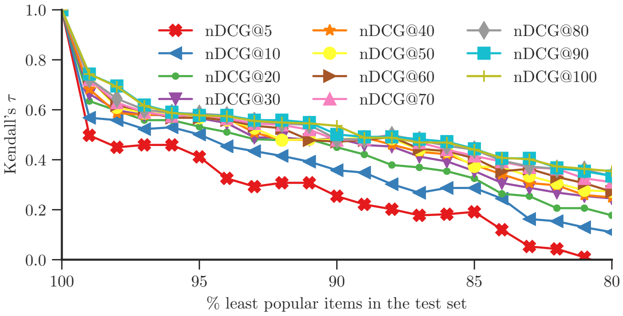


Figure 3: Kendall's correlation of different cut-offs of nDCG with respect to themselves using the full test set when increasing the popularity bias on the MovieLens 1M dataset.

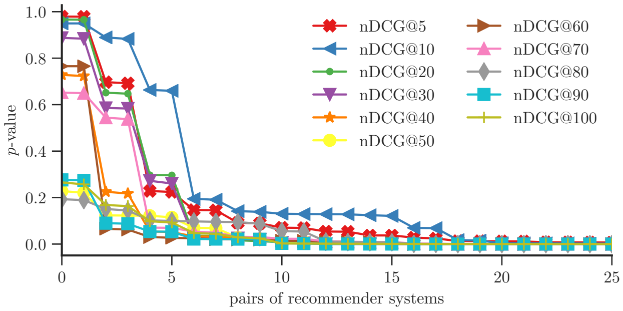


Figure 4: Analysis of the discriminative power of different cut-offs of nDCG on the MovieLens 1M dataset.

7 CHOOSING AMONG METRICS

In the previous section, we compared each metric against themselves using different cut-offs and we found that a cut-off of $n = 100$ is a good choice due to its robustness and discriminative properties. Now, we fix the cut-off to 100 and compare the previous metrics among each other to study which have more desirable properties—robustness and discriminative power.

Correlation Among Metrics. Herlocker et al. studied the correlation among some metrics (some of them barely used anymore) using only variants of one collaborative filtering algorithm on one dataset and recommends further investigation [19]. Therefore, we study the correlation among the system orderings according to different modern ranking metrics on three datasets. Fig. 5 shows Kendall's correlation among metrics on the MovieLens 1M, LibraryThing and BeerAdvocate datasets. On the LibraryThing collection, all correlations are above 0.9 threshold which indicates that the metrics produce almost identical rankings. On the other two datasets, we observed stronger differences with some correlations below 0.8.

We see that MRR differs noticeably, especially on the MovieLens 1M and the LibraryThing datasets. Bpref also shows a low correlation with the other metrics on the BeerAdvocate collection. Note that bpref is poorly correlated with MAP on this dataset which is a surprising result since bpref was designed to do so [9]. We suspect that this may be produced by the highly skewed long tail of this dataset. Instead, MAP is strongly correlated with nDCG on the three datasets. Nevertheless, the ranking produced by the rest of the metrics showed a fairly strong correlation among them.

Robustness Among Metrics. Fig. 6 depicts the results of the experiments of robustness to the sparsity bias. We can see that all the metrics are fairly robust to this bias since the correlation is above 0.9 even when removing half of the test set. Precision and nDCG showed very good figures of robustness to sparsity on the three datasets (precision especially on BeerAdvocate). In contrast, bpref, and to a lesser extent infAP and MRR, show poor robustness to sparsity. This result is interesting because it is different from what happens in IR. On the one hand, bpref and infAP are techniques proposed for dealing with incomplete judgements in IR [9, 44], but in top-N recommendation they are less robust than other metrics. We should note that bpref and infAP were designed for approximating average precision in scenarios with incomplete judgements while this metric is not such gold standard in recommendation. Still, it is surprising that MAP showed better robustness figures than bpref and infAP on LibraryThing and BeerAdvocate. On the other hand, utility-based metrics such as MRR were found to be more resilient to changes in pooling depth which is related to the sparsity bias in recommendation [24].

Fig. 7 shows the robustness to the popularity bias. On the BeerAdvocate dataset, the correlations quickly drop after removing a small percentage of the most popular items even reaching negative correlation values. This phenomenon is likely caused by the highly skewed long tail distribution of this dataset. Therefore, it is difficult to draw conclusions from this collection. Overall, precision is the best metric in terms of robustness to popularity whereas MRR is the worst one. The robustness to the popularity bias of the rest of the metrics depends heavily on the dataset.

We can claim that MRR is the least robust metric. This utility-based metric suffers heavily from sparsity and popularity biases. In contrast, precision is the most robust metric. More sophisticated metrics such as nDCG also present good figures of robustness; however, their additional complexity may be the reason why they are less robust than simple binary metrics such as precision.

Discriminative Power Among Metrics. Fig. 8 reports our findings in terms of discriminative power of the different studied metrics. We also present the values of DP (an approximation of the area under the p-value curve) in Table 3. Although the results vary across datasets, we can find some general trends. We see that bpref, and to a lesser extent infAP, presents low discriminative power across all datasets. In contrast, nDCG and precision (in this order) present the highest discriminative power on the test collections with great difference to the rest of the metrics. Finally, MAP, Recall and MRR show an erratic performance in terms of discriminative power depending on the dataset.

Table 3: Values of DP (lower is better) of P, Recall, MAP, nDCG, MRR, bpref and infAP (using a cut-off of 100) on the MovieLens 1M, LibraryThing and BeerAdvocate datasets.

Dataset	P	Recall	MAP	nDCG	MRR	bpref	infAP
MovieLens1M	2.6	7.0	2.8	**1.4**	15.5	9.9	8.4
LibraryThing	1.5	5.9	3.6	**0.2**	2.9	5.4	3.8
BeerAdvocate	**1.9**	8.3	10.7	4.4	5.8	12.7	4.8

8 CONCLUSIONS AND FUTURE WORK

In this paper, we studied the robustness and discriminative power of several ranking metrics, originally used in IR, when applied to the top-N recommendation task. To this end, we adapted and extended previous methodologies developed in IR for studying robustness against incompleteness and discriminative power.

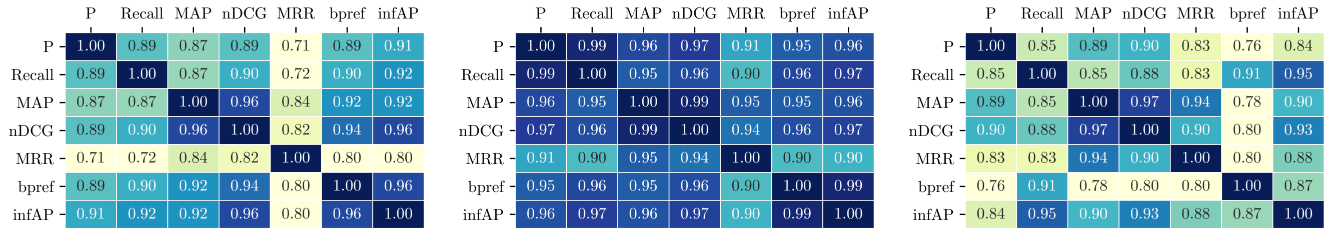
We found that deeper cut-offs (around 100) offer better robustness to sparsity and popularity biases than shallower cut-offs (5-10) which are traditionally used in RS evaluation. Therefore, we conclude that we should employ deeper cut-offs because they are more reliable in terms of robustness and discriminative power in offline evaluations. Although only a few recommendations are displayed to users, the use of deeper cut-offs allow us to perform more robust and discriminative evaluations of recommender systems.

Our findings suggest that precision, a simple binary metric, is very robust to sparsity and popularity biases. Normalised Discounted Cumulative Gain also presented high robustness to the sparsity bias and moderate robustness to the popularity bias. Moreover, in terms of discriminative power, nDCG and to a lesser degree precision showed the best figures of all the tested metrics. We found that bpref and infAP—which were proposed to address incompleteness in IR—as well as MRR perform poorly in RS evaluation.

We envision to extend this work to different types of metrics. Apart from ranking accuracy, diversity and novelty are also important properties of recommender systems [11]. It would be interesting to analyse which diversity and novelty metrics provide better robustness or discriminative power. Furthermore, in this work, we have focused on the AllItems methodology because it is the most similar to IR evaluation [3], but we also plan to study further evaluation procedures. Finally, we intend to analyse the impact of different dataset partitioning schemes such as temporal splits and n-fold cross-validation.

ACKNOWLEDGMENTS

This work was supported by projects TIN2015-64282-R and TIN2016-80630-P (MINECO/ERDF), projects ED431G/01 and GPC ED431B 2016/035 (Xunta de Galicia/ERDF) and grant FPU014/01724 (MECD).



	P	Recall	MAP	nDCG	MRR	bpref	infAP
P	1.00	0.89	0.87	0.89	0.71	0.89	0.91
Recall	0.89	1.00	0.87	0.90	0.72	0.90	0.92
MAP	0.87	0.87	1.00	0.96	0.84	0.92	0.92
nDCG	0.89	0.90	0.96	1.00	0.82	0.94	0.96
MRR	0.71	0.72	0.84	0.82	1.00	0.80	0.80
bpref	0.89	0.90	0.92	0.94	0.80	1.00	0.96
infAP	0.91	0.92	0.92	0.96	0.80	0.96	1.00

a: MovieLens 1M.

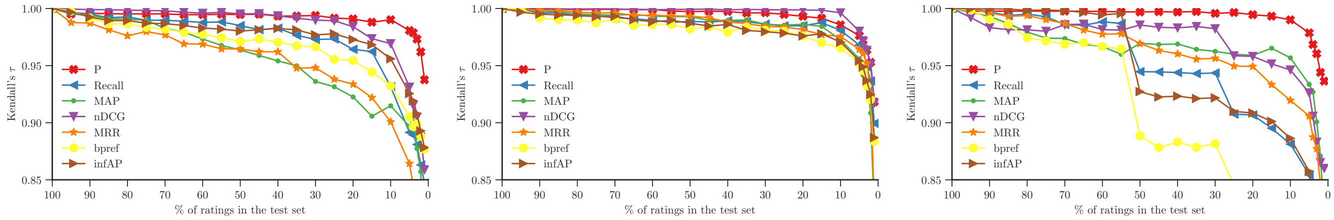
	P	Recall	MAP	nDCG	MRR	bpref	infAP
P	1.00	0.99	0.96	0.97	0.91	0.95	0.96
Recall	0.99	1.00	0.95	0.96	0.90	0.96	0.97
MAP	0.96	0.95	1.00	0.99	0.95	0.95	0.96
nDCG	0.97	0.96	0.99	1.00	0.94	0.96	0.97
MRR	0.91	0.90	0.95	0.94	1.00	0.90	0.90
bpref	0.95	0.96	0.95	0.96	0.90	1.00	0.99
infAP	0.96	0.97	0.96	0.97	0.90	0.99	1.00

b: LibraryThing.

	P	Recall	MAP	nDCG	MRR	bpref	infAP
P	1.00	0.85	0.89	0.90	0.83	0.76	0.84
Recall	0.85	1.00	0.85	0.88	0.83	0.91	0.95
MAP	0.89	0.85	1.00	0.97	0.94	0.78	0.90
nDCG	0.90	0.88	0.97	1.00	0.90	0.80	0.93
MRR	0.83	0.83	0.94	0.90	1.00	0.80	0.88
bpref	0.76	0.91	0.78	0.80	0.80	1.00	0.87
infAP	0.84	0.95	0.90	0.93	0.88	0.87	1.00

c: BeerAdvocate.

Figure 5: Correlation of P, Recall, MAP, nDCG, MRR, bpref and infAP (using a cut-off of 100) with each other on the MovieLens 1M, LibraryThing and BeerAdvocate datasets.

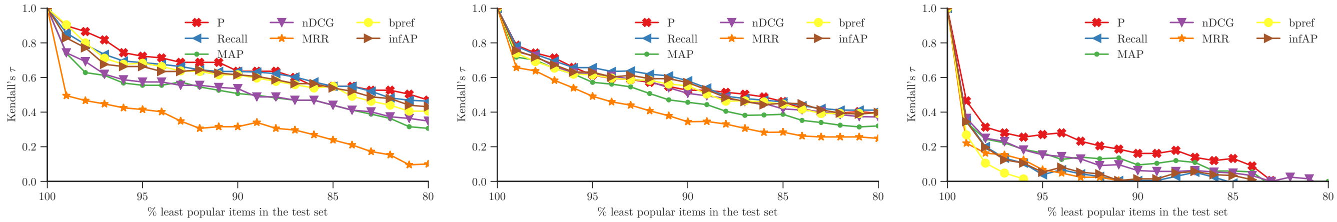


a: MovieLens 1M.

b: LibraryThing.

c: BeerAdvocate.

Figure 6: Kendall's correlation of P, Recall, MAP, nDCG, MRR, bpref and infAP (using a cut-off of 100) with respect to themselves using the test set when increasing the sparsity bias on the MovieLens 1M, LibraryThing and BeerAdvocate collections.

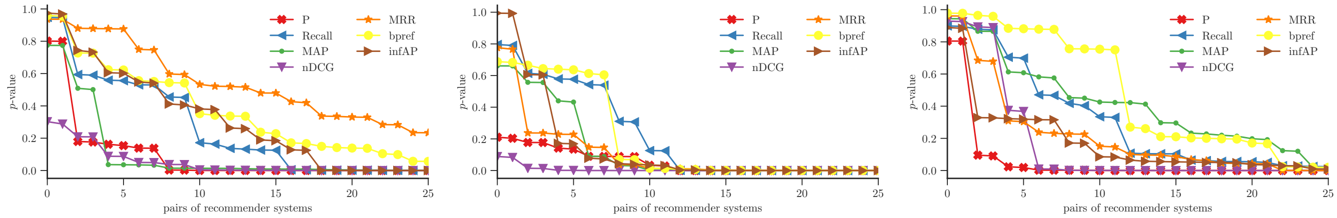


a: MovieLens 1M.

b: LibraryThing.

c: BeerAdvocate.

Figure 7: Kendall's correlation of P, Recall, MAP, nDCG, MRR, bpref and infAP (using a cut-off of 100) with respect to themselves using the test set when increasing the popularity bias on the MovieLens 1M, LibraryThing and BeerAdvocate collections.



a: MovieLens 1M.

b: LibraryThing.

c: BeerAdvocate.

Figure 8: Analysis of the discriminative power of P, Recall, MAP, nDCG, MRR, bpref and infAP (using a cut-off of 100) on the MovieLens 1M, LibraryThing and BeerAdvocate datasets.

REFERENCES

[1] Joeran Beel and Stefan Langer. 2015. A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems. In *TPDL '15*. Springer, 153–168. https://doi.org/10.1007/978-3-319-24592-8_12

[2] Nicholas J. Belkin and W. Bruce Croft. 1992. Information Filtering and Information Retrieval: Two Sides of the Same Coin? *Commun. ACM* 35, 12 (1992), 29–38. https://doi.org/10.1145/138859.138861

[3] Alejandro Bellogín, Pablo Castells, and Iván Cantador. 2011. Precision-Oriented Evaluation of Recommender Systems. In *RecSys '11*. ACM, New York, NY, USA, 333–336. https://doi.org/10.1145/2043932.2043996

[4] Alejandro Bellogín, Pablo Castells, and Iván Cantador. 2017. Statistical biases in Information Retrieval metrics for recommender systems. *Information Retrieval Journal* 20, 6 (2017), 606–634. https://doi.org/10.1007/s10791-017-9312-z

[5] Alejandro Bellogín, Jun Wang, and Pablo Castells. 2013. Bridging Memory-Based Collaborative Filtering and Text Retrieval. *Information Retrieval* 16, 6 (2013), 697–724. https://doi.org/10.1007/s10791-012-9214-z

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.

[7] Chris Buckley, Darrin Dimmick, Ian Soboroff, and Ellen Voorhees. 2007. Bias and the Limits of Pooling for Large Collections. *Information Retrieval* 10, 6 (2007), 491–508. https://doi.org/10.1007/s10791-007-9032-x

[8] Chris Buckley and Ellen M. Voorhees. 2000. Evaluating evaluation measure stability. In *SIGIR '00*. ACM, New York, NY, USA, 33–40. https://doi.org/10.1145/345508.345543

[9] Chris Buckley and Ellen M. Voorhees. 2004. Retrieval Evaluation with Incomplete Information. In *SIGIR '04*. ACM, New York, NY, USA, 25–32. https://doi.org/10.1145/1008992.1009000

[10] Stefan Büttcher, Charles L A Clarke, Peter C K Yeung, and Ian Soboroff. 2007. Reliable Information Retrieval Evaluation with Incomplete and Biased Judgements. In *SIGIR '07*. ACM, New York, NY, USA, 63–70. https://doi.org/10.1145/1277741.1277755

[11] Pablo Castells, Neil J. Hurley, and Saúl Vargas. 2015. Novelty and Diversity in Recommender Systems. In *Recommender Systems Handbook* (2nd ed.), Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, Boston, MA, 881–918. https://doi.org/10.1007/978-1-4899-7637-6_26

[12] Paolo Cremonesi, Franca Garzotto, Sara Negro, Alessandro Vittorio Papadopoulos, and Roberto Turrin. 2011. Looking for "Good" Recommendations: A Comparative Evaluation of Recommender Systems. In *INTERACT '11*. Springer, Berlin, Heidelberg, 152–168. https://doi.org/10.1007/978-3-642-23765-2_11

[13] Paolo Cremonesi, Franca Garzotto, and Roberto Turrin. 2013. User-Centric vs. System-Centric Evaluation of Recommender Systems. In *INTERACT '13*. Springer, Berlin, Heidelberg, 334–351. https://doi.org/10.1007/978-3-642-40477-1_21

[14] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of Recommender Algorithms on Top-N Recommendation Tasks. In *RecSys '10*. ACM, New York, NY, USA, 39–46. https://doi.org/10.1145/1864708.1864721

[15] Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, Boca Raton, Florida, USA.

[16] Norbert Fuhr. 2017. Some Common Mistakes In IR Evaluation, And How They Can Be Avoided. *SIGIR Forum* 51, 3 (2017), 32–41. https://doi.org/10.1145/3190580.3190586

[17] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. 2014. Offline and Online Evaluation of News Recommender Systems at swissinfo.ch. In *RecSys '14*. ACM, New York, NY, USA, 169–176. https://doi.org/10.1145/2645710.2645745

[18] Asela Gunawardana and Guy Shani. 2015. Evaluating Recommender Systems. In *Recommender Systems Handbook* (2nd ed.), Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, Boston, MA, 265–308. https://doi.org/10.1007/978-1-4899-7637-6_8

[19] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems* 22, 1 (2004), 5–53. https://doi.org/10.1145/963770.963772

[20] Thomas Hofmann. 2004. Latent Semantic Models for Collaborative Filtering. *ACM Transactions on Information Systems* 22, 1 (2004), 89–115. https://doi.org/10.1145/963770.963774

[21] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM '08*. IEEE, Washington, DC, USA, 263–272. https://doi.org/10.1109/ICDM.2008.22

[22] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446. https://doi.org/10.1145/582415.582418

[23] M. G. Kendall. 1938. A New Measure of Rank Correlation. *Biometrika* 30, 1-2 (1938), 81–93. https://doi.org/10.1093/biomet/30.1-2.81

[24] Xiaolu Lu, Alistair Moffat, and J. Shane Culpepper. 2016. The Effect of Pooling and Evaluation Depth on IR Metrics. *Information Retrieval Journal* 19, 4 (2016), 416–445. https://doi.org/10.1007/s10791-016-9282-6

[25] Benjamin M. Marlin, Richard S. Zemel, Sam Roweis, and Malcolm Slaney. 2007. Collaborative Filtering and the Missing at Random Assumption. In *UAI'07*. AUAI Press, Arlington, Virginia, United States, 267–275.

[26] Osvaldo Matos-Junior, Nivio Ziviani, Fabiano Botelho, Marco Cristo, Anisio Lacerda, and Altigran Soares da Silva. 2012. Using Taxonomies for Product Recommendation. *Journal of Information and Data Management* 3, 2 (2012), 85–100.

[27] Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being Accurate is Not Enough: How Accuracy Metrics have hurt Recommender Systems. In *CHI EA '06*. ACM, New York, NY, USA, 1097. https://doi.org/10.1145/1125451.1125659

[28] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *ICDM '11*. IEEE Computer Society, Washington, DC, USA, 497–506. https://doi.org/10.1109/ICDM.2011.134

[29] Javier Parapar, Alejandro Bellogín, Pablo Castells, and Álvaro Barreiro. 2013. Relevance-based Language Modelling for Recommender Systems. *Information Processing & Management* 49, 4 (2013), 966–980. https://doi.org/10.1016/j.ipm.2013.03.001

[30] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI '09*. AUAI Press, Arlington, VA, US, 452–461.

[31] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. *Recommender Systems Handbook* (2nd ed.). Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7637-6

[32] Tetsuya Sakai. 2006. Evaluating Evaluation Metrics Based on the Bootstrap. In *SIGIR '06*. ACM, New York, NY, USA, 525–532. https://doi.org/10.1145/1148170.1148261

[33] Tetsuya Sakai and Noriko Kando. 2008. On Information Retrieval Metrics Designed for Evaluation with Incomplete Relevance Assessments. *Information Retrieval* 11, 5 (2008), 447–470. https://doi.org/10.1007/s10791-008-9059-7

[34] Mark D Smucker, James Allan, and Ben Carterette. 2007. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *CIKM '07*. ACM, New York, New York, USA, 623. https://doi.org/10.1145/1321440.1321528

[35] Karen Spärck Jones and Cornelis J. Van Rijsbergen. 1975. *Report on the Need for and Provision of an Ideal Information Retrieval Test Collection*. University Computer Laboratory.

[36] Harald Steck. 2010. Training and Testing of Recommender Systems on Data Missing Not at Random. In *KDD '10*. ACM, New York, NY, USA, 713–722. https://doi.org/10.1145/1835804.1835895

[37] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. 2009. Scalable Collaborative Filtering Approaches for Large Recommender Systems. *Journal of Machine Learning Research* 10 (2009), 623–656. https://doi.org/10.1145/1577069.1577091

[38] Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. 2016. Efficient Pseudo-Relevance Feedback Methods for Collaborative Filtering Recommendation. In *ECIR '16*. Springer, Berlin, Heidelberg, 602–613. https://doi.org/10.1007/978-3-319-30671-1_44

[39] Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. 2016. Item-based Relevance Modelling of Recommendations for Getting Rid of Long Tail Products. *Knowledge-Based Systems* 103 (2016), 41–51. https://doi.org/10.1016/j.knosys.2016.03.021

[40] Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. 2016. Language Models for Collaborative Filtering Neighbourhoods. In *ECIR '16*. Springer, Berlin, Heidelberg, 614–625. https://doi.org/10.1007/978-3-319-30671-1_45

[41] Ellen M. Voorhees. 2001. Evaluation by Highly Relevant Documents. In *SIGIR '01*. ACM, New York, NY, USA, 74–82. https://doi.org/10.1145/383952.383963

[42] Ellen M. Voorhees. 2002. The Philosophy of Information Retrieval Evaluation. In *CLEF 2001*. Springer, Berlin, Heidelberg, 355–370. https://doi.org/10.1007/3-540-45691-0_34

[43] Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. 2006. A User-Item Relevance Model for Log-Based Collaborative Filtering. In *ECIR '06*. Vol. 3936. Springer, London, UK, 37–48. https://doi.org/10.1007/11735106_5

[44] Emine Yilmaz and Javed A. Aslam. 2008. Estimating Average Precision when Judgments are Incomplete. *Knowledge and Information Systems* 16, 2 (2008), 173–211. https://doi.org/10.1007/s10115-007-0101-7

[45] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. 2012. Challenging the Long Tail Recommendation. *Proceedings of the VLDB Endowment* 5, 9 (2012), 896–907. https://doi.org/10.14778/2311906.2311916