Information Retrieval Journal
https://doi.org/10.1007/s10791-020-09377-x

Assessing ranking metrics in top-N recommendation

Daniel Valcarce[1] · **Alejandro Bellogín**[2] · **Javier Parapar**[1] · **Pablo Castells**[2]

Received: 19 August 2019 / Accepted: 20 May 2020

Abstract
The evaluation of recommender systems is an area with unsolved questions at several levels. Choosing the appropriate evaluation metric is one of such important issues. Ranking accuracy is generally identified as a prerequisite for recommendation to be useful. Ranking metrics have been adapted for this purpose from the Information Retrieval field into the recommendation task. In this article, we undertake a principled analysis of the robustness and the discriminative power of different ranking metrics for the offline evaluation of recommender systems, drawing from previous studies in the information retrieval field. We measure the robustness to different sources of incompleteness that arise from the sparsity and popularity biases in recommendation. Among other results, we find that precision provides high robustness while normalized discounted cumulative gain offers the best discriminative power. In dealing with cold users, we also find that the geometric mean is more robust than the arithmetic mean as aggregation function over users.

Keywords Recommender systems · Top-N recommendation · Evaluation · Ranking metrics · Robustness · Discriminative power

This work is supported by the Spanish Ministry of Science, Innovation and Universities and the ERDF (Projects TIN2016-80630-P and RTI2018-093336-B-C22) and by the Regional Government of Galicia and the ERDF (accreditation ED431G/01 and ED431B 2019/03). The authors also acknowledge the very helpful feedback from the anonymous reviewers.

✉ Javier Parapar
javier.parapar@udc.es

Daniel Valcarce
daniel.valcarce@udc.es

Alejandro Bellogín
alejandro.bellogin@uam.es

Pablo Castells
pablo.castells@uam.es

[1] Information Retrieval Lab, Centro de Investigación en Tecnoloxías da Información e as Comunicacións (CITIC), Universidade da Coruña, A Coruña, Spain

[2] Information Retrieval Group, Universidad Autónoma de Madrid, Madrid, Spain

Published online: 08 June 2020

1 Introduction

Recommender Systems assist users in finding their way in massive information spaces, in many application domains. To deal with such an information overload, these systems help users in discovering relevant pieces of information or items (Ricci et al. 2015). The goal of a recommender system is hence to present items that may be of interest to the users. Although the idea is simple and intuitive, the evaluation of a recommendation algorithm is a more difficult endeavour than it may appear (Ferro et al. 2018). In this article, we aim to shed light on a central issue in recommender systems assessment: the evaluation metrics.

In the early days of the field, recommender systems were conceived to act as rating predictors. The early recommendation methods aimed to forecast the ratings that users would assign to each item (Herlocker et al. 2004; Gunawardana and Shani 2015). The evaluation of rating predictors therefore relied on error metrics such as the root mean squared error (RMSE) or the mean absolute error (MAE) (Gunawardana and Shani 2015). The rationale behind this recommendation task is that if a model predicts ratings correctly, then we can provide good recommendations by suggesting the items with the highest predicted scores.

However, later work suggested that the assessment of recommendation methods based on error metrics provides a poor proxy of true user satisfaction in real applications (Cremonesi et al. 2010; Gunawardana and Shani 2015; McNee et al. 2006). In part, this is because recommender systems in production typically offer a short list of recommendations where the predicted ratings are not shown (Herlocker et al. 2004). Producing a short list of N recommendations per user—instead of accurately predicting the rating values that users would assign—is commonly known as top-N recommendation (Cremonesi et al. 2010); the focus in this task is to provide a list with the most interesting items unknown to the user. Additionally, evaluation tailored for rating prediction gives the same importance to all the items; in contrast, top-N recommendation only cares for the top recommended items that each user may browse in the ranking. For these reasons, a paradigm shift from rating prediction to top-N recommendation has taken place in the field in the last years.

Aside from this paradigm shift, accuracy as a broad concept remains a (or may we say the) primary requirement for recommendation to make sense, notwithstanding the increasing interest for additional complementary recommendation properties (such as diversity and novelty Castells et al. 2015; Herlocker et al. 2004; Gunawardana and Shani 2015). For a given user, we might say that a particular suggestion is "correct" if that user likes the recommended item. With this goal in mind, we may conduct online or offline experiments to determine whether a user likes the recommendations she is delivered. Online evaluation (chiefly A/B testing Siroker and Koomen 2013; Kohavi et al. 2009) is generally considered the final word in assessing how effective a recommender system is. The known cost and limitations of online experiments are commonly relieved (or eluded for lack of alternative) by complementary offline evaluation, typically as a first filter before moving to A/B testing. Offline experiments exploit a dataset of previously collected user-item interactions which is usually partitioned into at least two subsets (Gunawardana and Shani 2015): a training set used as input to the recommendation algorithm under evaluation and a test split employed for measuring the performance of the algorithm according to an evaluation metric. Additionally, it is usually appropriate to use an extra validation set for tuning the hyperparameters.

The Information Retrieval (IR) community has built well-established evaluation methodologies—often broadly referred to as the Cranfield paradigm—through careful and iterative design, in-depth analysis, and consensus-building over the decades by

community-wide initiatives such as TREC (Voorhees 2002). IR and Recommender Systems are strongly related fields where both seek to provide relevant pieces of information to the users (Belkin and Croft 1992). One primary difference lies in the representation of the information need: while an IR system typically uses an explicit query specified by the user, recommender systems exploit the user's interaction records as an implicit query. The Cranfield paradigm measures how a retrieval system meets the information needs of the users using ranking metrics. Many of these metrics have also been used to assess recommender systems in the top-N recommendation task. However, in contrast to IR, the evaluation of recommender systems has not undergone a comparable process in terms of standardization and consensus-building. The recommender systems field can benefit from the IR methodology in this area, but the recommendation task has peculiarities that require specific adaptations and analysis.

For instance, in the Cranfield paradigm, documents are manually judged for relevance, which is generally not the case in recommender system evaluation: instead, experimenters hold out a subset of the available user interaction records as a proxy to serve this purpose. These pseudo-judgments are highly incomplete and obtained in a very different way compared to the IR relevance judgments. Since the assumptions of the Cranfield paradigm are substantially different from those in the evaluation of recommendations, the adoption of IR metrics requires at the very least careful consideration.

In this article, we extend our previous work (Valcarce et al. 2018) by several means: (i) we include new evaluation metrics (F-measure and expected reciprocal rank) in the analysis, (ii) we include the use of the geometric mean in addition to the arithmetic mean to aggregate metrics (iii) we change the evaluation protocol using 5-fold cross validation and random splitting and (iv) we designed a new set of experiments analyzing the robustness to:

- items missing at random,
- users missing at random,
- missing users with the largest user profiles.

1.1 Research objective

In this work, we study the applicability of several IR metrics to top-N recommendation. Our research objective is to study the robustness and discriminative power of ranking metrics in top-N recommendation. Most of these metrics are already being used in recommender system evaluation, but they have not been thoroughly studied. We adapt and extend evaluation methodologies from IR to assess the robustness and discriminative power of several ranking metrics in the top-N recommendation task. In this context, a metric is robust when it presents a similar behavior when less relevance judgments are available. Likewise, a metric is discriminative when changes in its values produce statistically significant differences.

As we shall show later, we analyze the robustness of several IR metrics to different incompleteness scenarios: ratings, items, and users missing at random, and when larger items (sensitivity to popular items) and users (sensitivity to active users) are removed. Inspired by the results obtained in the last scenario, we propose to aggregate ranking metrics using the geometric mean instead of the arithmetic mean, where we find a more consistent behavior regarding robustness to incompleteness for some metrics. In general, the

results show that precision offers the best robustness figures whereas NDCG provides the highest discriminative power.

The remainder of the paper is structured as follows: in Sect. 2 we present some background concepts about evaluation in IR and recommender systems, and a summary of the work related to our present research. Section 3 describes the ranking metrics that will be used throughout the paper, including the equivalence between IR and recommender systems evaluation assumptions. In Sect. 4 we introduce our methodology to analyze the robustness and discriminative power of ranking metrics in recommender systems. The results of the analysis are presented in Sects. 6 and 7 based on the experimental settings introduced in Sect. 5, first averaging the metrics according to the arithmetic mean (Sect. 6) and later according to the geometric mean (Sect. 7). Finally, conclusions and future work directions are presented in Sect. 8.

2 Background and related work

Evaluation plays a crucial role in IR and recommender systems: the effectiveness of any retrieval or recommender system is measured empirically (Gunawardana and Shani 2015; Herlocker et al. 2004), but also theoretical results help in improving the evaluation efforts. Therefore, the evaluation methodology has important theoretical and practical applications in both fields. While IR has established the Cranfield paradigm as the standard evaluation methodology (Voorhees 2002), different approaches coexist in the assessment of top-N recommendations.

2.1 Information retrieval evaluation

The Cranfield paradigm is a well-founded evaluation methodology used and evolved over the years in the Information Retrieval field. This paradigm is based on test collections containing documents, topics (describing information needs that would result in a query), and relevance judgments for each topic (Voorhees 2002). Assessors are responsible for judging the documents and discriminating which ones are relevant to each topic. Using these relevance judgments, ranking metrics can evaluate the output of a retrieval system.

The Cranfield paradigm relies on three fundamental assumptions: (i) the information needs of the users—which are specified by the topics—can be approximated by topical similarity; (ii) relevance is independent of the users, which implies that a set of relevance judgments is valid for any user; and (iii) judgments are complete, i.e., all the relevant documents for each topic are known. Although these assumptions do not strictly hold in general, they are reasonable and some deficiencies can be tolerated and/or compensated for Voorhees (2002). For this reason, the Cranfield paradigm has become the standard systematic procedure for offline evaluation in IR.

A retrieval system is evaluated for a particular topic by producing a list of documents sorted by decreasing relevance according to the retrieval model implemented by the system. Then, a metric is computed on this ranking, using the relevance judgments for the topic. The general quality of a retrieval strategy is measured as the average metric score over all the topics in the test collection. Typically, the arithmetic mean is used but the geometric mean has also been explored by some authors (Voorhees 2005; Robertson 2006).

One problem that this evaluation paradigm has to tackle is the enormous volume of information in modern test collections: datasets are way too large to have complete

relevance judgments for each topic. For this reason, a process called *pooling* selects which documents should be assessed by humans (Spärck and Van Rijsbergen 1975; Voorhees 2002). Pooling reduces the evaluation effort of the human assessors by filtering potentially non-relevant documents. Using pooling, all documents that do not appear in the pool are assumed to be non-relevant. The idea is that we should make relative (instead of absolute) evaluations with the test collections. To this end, we need unbiased relevance judgments. Pooling, if performed correctly, has shown to be a good enough approximation (Voorhees 2002; Losada et al. 2017). Nevertheless, large-scale datasets (such as ClueWeb) contain hundreds of millions of documents which are shallow pooled, leaving a significant number of potentially relevant documents unjudged as a result (Lu et al. 2016; Kutlu et al. 2018).

The limitations and biases of the Cranfield paradigm have been extensively studied. For instance, there have been efforts to overcome the bias produced by pooling (Buckley et al. 2007; Büttcher et al. 2007). Also, Buckley and Voorhees (2004) studied how the number of relevance judgments affects different precision-oriented metrics. These authors defined the robustness of a metric with respect to incomplete judgments as how well the metric correlates with itself when the relevance judgments are incomplete. They propose the Bpref metric to address this issue—with incomplete judgments, the metric is shown to have better correlation with itself and with average precision (AP) with all judgments than other standard IR metrics. The same authors also found that Bpref preserves the absolute scores and the relative ranking of systems better than MAP or precision. Later, Yilmaz and Aslam (2008) proposed three estimates of AP for an incomplete judgments scenario. The proposed estimates showed a better correlation between themselves and AP than Bpref. The correlations between system rankings were measured in terms of Kendall's correlation (Kendall 1938). Among different proposed alternatives, InfAP was the metric that provided the best results (Yilmaz and Aslam 2008). To measure the robustness to incomplete judgments in these experiments, the metrics were calculated using random subsets of relevance judgments. Buckley and Voorhees (2004) used stratified random sampling, while Yilmaz and Aslam (2008) employed random sampling. However, both samplings are equivalent in expectation (Yilmaz and Aslam 2008). More recently, Lu et al. (2016) thoroughly studied the effect of the pooling depth in several IR metrics providing a list of advice for IR evaluation.

In addition to the robustness to judgment incompleteness, discriminative power is another property of evaluation metrics that has been extensively studied in IR (Buckley and Voorhees 2000; Lu et al. 2016; Sakai 2006; Sakai and Kando 2008). Discriminative power measures the capability of a metric to discriminate between 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 (2000) proposed a first attempt at analysing the discriminative power of a metric using a *fuzziness value*. Later, Sakai (2006) introduced a more formal method based on the bootstrap test. 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 the Student's t statistic for this purpose. To avoid fixing a particular significance level, Lu et al. (2016) proposed to report the median system-pair p-value as a measure of discriminative power. Sakai and Kando (2008) also studied how the discriminative power of a metric varies when using incomplete judgments.

2.2 Recommender systems evaluation

The move from error metrics to ranking-oriented evaluation was a step towards a more realistic view of the recommendation task, and cast recommendation, for many purposes, as an IR task (Cremonesi et al. 2010; Gunawardana and Shani 2015; Herlocker et al. 2004). The need of Information Retrieval systems to anticipate user needs and provide relevant information without the need of an explicit query (usually called *zero query* search Allan et al. 2012) was also one of the motivations why IR researchers started to look into Recommender Systems.

Evaluation is still however one of the most active and open research areas to date in the RS field (Ferro et al. 2018; Konstan and Adomavicius 2013). The consistency between offline and online evaluation is, for one, a recurrent and prominent open issue (Rossetti et al. 2016). Recent studies restricted to particular domains, for instance, have shown discrepancies between the click-through rate and the score provided by offline metrics (Garcin et al. 2014; Beel and Langer 2015). A more exhaustive study analyzed seven different recommendation algorithms from a user-centric perspective using two accuracy metrics finding a poor match between the perceived quality and recall and fallout metrics (Cremonesi et al. 2011). 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 (Cremonesi et al. 2013). Overall, online evaluation is strongly sensitive to several variables such as the domain, the demographics of the users, or the user interface of recommendations, which can make it difficult to isolate the effect of the recommendation algorithm at the core of an application from the bias of noisy variables. Moreover, reproducibility is difficult, if not impossible, to achieve when researchers do not have access to the original experimental environment (Said and Bellogín 2014). Even when a production system is available, A/B tests take a long time to run, typically more than 2 weeks, whereas many offline experiments can be run and repeated many times in a few hours, with no impact on the customers' quality of experience. Additionally, a limitation of online evaluation is that the recommender model may change during the A/B test. Most recommendation models adapt themselves to incoming feedback. This makes hard to draw objective conclusions about the effectiveness of different algorithms when they mutate in production. For all these reasons, offline experimentation is used consistently both in academia (as the primary means for research-oriented evaluation) and industry (as a complement), and typically constitutes the first step before online evaluation.

In contrast to the situation described in IR in the previous section, recommender systems test collections do not rely on assessors and pooling. Since RS lack a query and relevance is highly contextual and/or personal, obtaining relevance judgments from assessors is not possible. For this reason, a held-out test set of actual user interactions (such as ratings or clicks) is employed for testing purposes and the rest as input for the recommendation algorithm. However, this does not mean that recommender system evaluation is free from biases. Quite the contrary, Bellogín et al. showed that sparsity and popularity biases impact the evaluation of recommendations (Bellogín et al. 2017), and explored alternative test rating sampling approaches to counter such biases. Later on, Cañamares and Castells (2018) gave a formal explanation for the bias, and found out that the bias might preserve the comparisons between systems if certain conditions hold: the key factor is whether the probability of an item to be rated depends only on its relevance – in which case the bias preserves comparisons –, or depends also on any other characteristics or circumstances of specific items—in which case offline evaluation may return misleading results.

Table 1 Comparison between the evaluation assumptions of the Cranfield paradigm in Information Retrieval and recommender systems

Information retrieval	Recommender systems
Topical similarity can approximate the user's information need	User's information need may be captured in several different ways
Relevance does not depend on the users	Relevance is fully user-dependent
Relevance judgments are almost complete (pooling depth)	Relevance judgments are far from complete

Another recent strand of research has addressed the bias in offline evaluation as an issue of mismatch between the data gathering policy (e.g., users freely interacting with a deployed system) and item selection by the recommendation algorithms to be evaluated. Building on this perspective, techniques such as inverse propensity scoring have been explored to reduce the biases in the evaluation (Gilotte et al. 2018; Gruson et al. 2019; Swaminathan et al. 2017; Yang et al. 2018) and the evaluated algorithms (Joachims et al. 2017; Schnabel et al. 2016). One of the main challenges in these approaches is the development of accurate models of propensity (i.e., the probability that a user is observed interacting with an item). As a consequence, these techniques are not always easy to apply as an off-the-shelve option in evaluation. It is not (yet) fully clear how to reliably debias offline evaluation with a typical public dataset such as, for instance, MovieLens Harper and Konstan (2015). In this paper we shall therefore focus on common widespread offline evaluation scenarios as are in order today. Future progress and consolidation of debiasing practice would certainly motivate an extension of the study we present here, considering the application of the metrics of study in combination with new debiasing techniques.

On the other hand, metrics such as Bpref and InfAP, which have been proposed in IR to address incompleteness of relevance judgments (Buckley and Voorhees 2004; Yilmaz and Aslam 2008), have rarely been used in recommendation (Bellogín et al. 2013; Matos-Junior et al. 2012; Valcarce et al. 2016). To the best of our knowledge, our previous work (Valcarce et al. 2018) and the present continuation are the first systematic review of ranking metrics in recommendation regarding robustness to incompleteness and discriminative power. The results of our research provide important practical implications in how offline recommender systems evaluation should be performed.

3 Metrics

3.1 Cranfield for recommendation

We can establish an analogy between the Cranfield paradigm and top-N recommendation if we consider that the user contexts play the role of queries (since they are both associated with an information need to be satisfied). By this equivalence, we can evaluate item rankings as we evaluate document rankings. Cranfield evaluation assumes that relevance judgments are procured somehow, typically manually assigned by human assessors. Commonly available datasets for recommender system evaluation do not generally include such an explicit provision of manual judgments specifically intended to meet the experimental needs. Instead, a part of the user activity records in the dataset is held out as an approximation to Cranfield judgments. However, when evaluating recommendations, some Cranfield assumptions do not hold (see Table 1 for a summary). Compared to, e.g., TREC relevance

judgments, the held-out test records are considerably noisier, incomplete and biased (Bellogín et al. 2017).

Relevance is highly dependent on end-users: the same item may not be relevant to two different users, and this is, in fact, inherent to the recommendation problem—if items were equally liked or disliked by all users, the recommendation task would be a non-problem. Compared to ad-hoc search tasks, this significantly complicates the construction of Cranfield test collections for offline evaluation: it is impossible to delegate relevance labeling to a group of experts or assessors: no one can judge the personal relevance of items on behalf of the users that recommendations will target in the experiment, whereby (typically thousands of) end-users need to be fully involved in the data collection procedure. The incompleteness of judgments is likewise not only hardly avoidable, but a condition for recommendation to be a meaningful task. If user preferences for all items were known, there would be nothing to predict—no gap to fill—by a recommender system.

In addition, when approximating relevance judgments with a held-out test set, a trade-off arises between how much data is used for the training and the test splits. A larger training subset (in exchange for a smaller test subset) allows for better modeling, at the expense of potentially worse evaluation reliability, and vice versa. Finally, the long tail distribution of ratings in recommender systems impacts the recommendation process (Anderson 2008; Cañamares and Castells 2018). IR collections, in contrast, do not have such a strong imbalance in the number of judgments per document (Bellogín et al. 2017).

On the other hand, while metrics such as mean average precision (MAP) and normalized discounted cumulated gain (NDCG) (Järvelin and Kekäläinen 2002) have been considered as reference metrics in IR (in spite of recent criticism Fuhr 2018), the generalization of the use of ranking metrics for recommender system evaluation is relatively recent, and a comparable consensus is missing as to which metric is the most suitable to measure the ranking quality.

3.2 Adaptation of ranking metrics

We use the following definitions and notation when adapting ranking metrics from IR. When using explicit feedback datasets in the form of ratings, user relevance is estimated by exploiting the ratings. All 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 for that user u and constitute the set that we will denote by $\mathcal{N}_u$. Likewise, we shall denote by $\mathcal{R}_u$ the set of those items rated by u in the test set with a score equal to or greater than the threshold τ, representing the set of relevant items for the user u. For instance, in datasets such as MovieLens Harper and Konstan (2015) or Netflix Bennett and Lanning (2007), with ratings ranging from 1 to 5, it is common to set τ to 4. The 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 (such as Bpref Buckley and Voorhees 2004 and InfAP Yilmaz and Aslam (2008)).

Given a set of users $\mathcal{U}$ and a set of items $\mathcal{I}$, the top-N recommendation task consists in producing a ranking of the N most relevant items in $\mathcal{I}$ for each user in $\mathcal{U}$ (Cremonesi et al. 2010). We represent the ranking of length n for user u as the list L_u^n. We refer to the item in the k-th position of that list by $L_u^n[k]$. Finally, we denote the rating from a user u to an item i by $r(u, i)$.

With this formulation, we select nine ranking metrics to analyze in our study. We select them as representative and commonly used metrics in IR, which we shall adapt for

recommender system evaluation. All of them range from 0 to 1 where the higher the value, the better. The adapted 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 (typically using the arithmetic mean). If a recommender system cannot provide recommendations for a particular user, we assign a value of zero to all the metrics for that user, thus penalizing the inability to provide recommendations to some users (user coverage shortfall). When using IR metrics, we commonly establish a specific depth n – the "cut-off"— at which the ranking is to be evaluated. Since some metrics have multiple versions with slight differences, in this work we adopt the `trec_eval`[1] implementation of the metrics which is the standard evaluation tool of the TREC initiative. Since some metrics are not implemented in `trec_eval`, we developed a fork called `rec_eval`[2] including them.

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

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

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:

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

F-Measure (F_1) The F-measure is the harmonic mean between precision and recall.

$$F1_u@n = \frac{2 \cdot P_u@n \cdot Recall_u@n}{P_u@n + Recall_u@n} \tag{3}$$

Average Precision (AP) Average Precision averages precision at the positions where a relevant item is found. When AP is averaged (using the arithmetic mean) over the set of topics in IR or users in recommender systems, it receives the name of mean AP (MAP).

$$AP_u@n = \frac{1}{\left|\mathcal{R}_u\right|} \sum_{k=1}^{n} \mathbb{I}\left(L_u^n[k] \in \mathcal{R}_u\right) P_u@k \tag{4}$$

where $\mathbb{I}$ denotes the indicator function.

Normalized Discounted Cumulative Gain (NDCG) This metric uses graded relevance (the values of the ratings) as well as positional information of the recommended items (Järvelin and Kekäläinen 2002). 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 is defined as:

$$NDCG_u@n = \frac{\sum_{k=1}^{n} G(u,n,k)\, D(k)}{\sum_{k=1}^{n} G^*(u,n,k)\, D(k)} \tag{5}$$

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

[2] https://github.com/dvalcarce/rec_eval.

A common discount function is $D(k) = \log_2^{-1}(k+1)$. Although there exist multiple options for defining the gain function, our preliminary experiments showed no meaningful differences among them. Therefore, we shall simply take $G(u,n,k) = r(u, L_u^n[k])$ as the gain function hereinafter. This gain function exploits the graded relevance feedback present in the ratings.

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 users, this metric is called Mean RR (MRR).

$$RR_u = \frac{1}{\min_k \mathbb{I}\left(L_u^n[k] \in \mathcal{R}_u\right)} \tag{6}$$

Expected Reciprocal Rank (ERR) This more recent metric is based on the so-called cascade user model and highly correlates with click-through rate in a search engine (Chapelle et al. 2009). ERR seeks to estimate the expected reciprocal amount of time that the user will take to find a relevant document.

$$ERR_u = \sum_{k=1}^{n} \frac{1}{k} \prod_{p=1}^{k-1} (1 - G(u,n,p)) G(u,n,k) \tag{7}$$

The gain function in ERR is as follows:

$$G(u,n,k) = \frac{2^{r(u,L_u^n[k])} - 1}{2^{r_{max}}} \tag{8}$$

where r_{max} denotes the maximum rating in the dataset.

Bpref This metric was designed to have high correlation with AP and, at the same time, be more robust to incomplete relevance judgments than AP (Buckley and Voorhees 2004). Bpref is inversely related to the number of judged non-relevant items that are located above each relevant item in the ranking list:

$$Bpref_u@n = \frac{1}{|\mathcal{R}_u|} \sum_{k=1}^{n} \mathbb{I}\left(L_u^n[k] \in \mathcal{R}_u\right) \left(1 - \frac{\min(|L_u^k \cap \mathcal{N}_u|, |\mathcal{R}_u|)}{\min(|\mathcal{N}_u|, |\mathcal{R}_u|)}\right) \tag{9}$$

Inferred Average Precision (InfAP) InfAP yields the same score MAP provides when the relevance judgments are complete; however, it is also a statistical estimate of MAP when using incomplete judgments (Yilmaz and Aslam 2008). InfAP has shown a better correlation with AP than Bpref under this scenario. This metric is given by:

$$InfAP_u@n = \frac{1}{|\mathcal{R}_u|} \sum_{k=1}^{n} \mathbb{I}\left(L_u^n[k] \in \mathcal{R}_u\right) E[P_u@k] \tag{10}$$

where the expected precision at position k is defined as:

$$E[P_u@k] = \frac{1}{k} + \frac{k-1}{k} \frac{|L_u^{k-1} \cap \mathcal{R}_u| + \varepsilon}{|L_u^{k-1} \cap \mathcal{R}_u| + |L_u^{k-1} \cap \mathcal{N}_u| + 2\varepsilon} \tag{11}$$

and ε is a small constant (we set ε to 0.00001 in our experiments following the `trec_eval` implementation).

4 Methodology

We now describe the proposed methodologies for studying the robustness to incompleteness and the discriminative power of the ranking metrics described in the previous section. Drawing from prior studies in a similar scope in the IR field, we adapt and extend them to the context of top-N recommendation. We start with the analysis of discriminative power and continue with the study of the robustness to different types of incompleteness. When we are to choose one best recommendation algorithm over others, statistically sound guarantees are desirable. A metric with higher discriminative power than another tends to produce statistically significant differences between systems more often. On the other hand, the robustness to incompleteness is an important concern in recommendation because the scarcity of relevance judgments is commonly a challenge to the reliability of metric values.

4.1 Discriminative power

When comparing two recommendation techniques using a particular ranking metric, we pay attention to the value of the metric for each system. For most metrics, we should prefer the recommendation model that produces the highest metric figures. However, the evaluation process is subject to some degree of randomness and uncertainty: the dataset is a sample of the whole data, and its manipulation (e.g., sampling test records) often adds further random variance. Differences in the values of a metric may therefore be subject to some degree of randomness or noise. For ranking metrics that are computed for each user, we can use paired difference tests to discriminate whether the metric means of two systems differ or not. Ideally, we expect that the set of measured values reflect a statistically significant difference. Otherwise, we would hardly be able to conclude much from the experiment.

In our experiments, we will compare the performance of multiple recommender systems in different collections according to the described metrics. We decided to choose the method proposed by Sakai (2006) for measuring the discriminative power of those metrics. In Sakai (2006) the author used the t-test, although in more recent work, he recommended the Tukey HSD test when comparing multiple runs (Sakai 2012). In this article, we use the permutation test (also known as Fisher's randomization test) with the difference in means as the test statistic (Efron and Tibshirani 1993). We took this choice because this test statistic provides a better estimation of the p-value (Efron and Tibshirani 1993; Parapar et al. 2019). 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. We use 100,000 samples which is enough to 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 (Efron and Tibshirani 1993).

For assessing how discriminative a given metric is, we compute the p-value of the test among every possible pair for recommenders using that metric. As we are doing paired testing, each comparison takes into account the array of metric values for each user produced by the two systems being tested. We plot the p-values[3] sorted by decreasing value as in Sakai (2006). We call each of those curves the p-value curve of the metric. A highly discriminative metric yields low p-values and, thus, its p-value curve should be closer to the origin.

[3] Actually, we limit the plot to the first 20 p-values.

Moreover, to summarize the p-value curve in one unique value for each metric, we propose to compute the area under the p-curve by summing up all the p-values for the given metric. We shall call this value DP (discriminative power). The lower the value of DP, the higher the discriminative power of the metric. Note that DP depends on the set of systems and the dataset. Therefore, DP can only be used to compare the same systems on the same dataset.

4.2 Robustness to incompleteness

As discussed earlier, incompleteness is a pervasive condition in the evaluation of recommender systems. The relevance judgments are formed by the ratings in the test set which are, by definition, incomplete since they are generated as a held-out subset of the whole dataset. More fundamentally, the dataset itself is incomplete because users have not rated all the items in the system—if they had, they might not need recommendations, to begin with. A reliable metric for recommendation should, therefore, be robust to incompleteness in the test set.

Incompleteness has been simulated in IR using unbiased random sampling techniques (Buckley and Voorhees 2004; Yilmaz and Aslam 2008). We propose a similar approach to play with incompleteness in recommender system evaluation. However, incompleteness in recommender systems can manifest itself in ratings, items, and users. We discuss next these types of incompleteness and present a procedure to simulate each of them.

4.2.1 Rating incompleteness

One of the most common biases in the recommender system evaluation is the sparsity bias which arises when we lack relevance judgments for user-item pairs involved in a recommendation we wish to evaluate (Bellogín et al. 2017). Because of the sparsity bias, the absolute values of the metrics lose meaning, but they are still valid for making relative comparisons (Bellogín et al. 2017), as long as the compared systems are equally affected—in expectation—by sparsity (Cañamares and Castells 2018).

We propose to assess the robustness of different ranking metrics to the ratings incompleteness using random samples of the test set. We define different test sizes starting from 100 to 1% of the size of the original test set, and we take 50 random samples of each size from the test set. Given a set of recommenders and a particular metric, we rank these systems for each test set sample. Then, we compute Kendall's rank correlation coefficient of each system ranking with respect to the ranking obtained using the original test set (Kendall 1938). Finally, by averaging the rank correlation of the samples with the same size, we obtain a final estimate of the robustness of a metric for each test size. The smaller the test size (i.e., the more aggressive the simulated sparsity), the lower the correlation of the system comparisons can be expected to be. We say that a metric is more robust than another if it stands at a higher average correlation with itself as the test set is reduced.

4.2.2 Item incompleteness

The sparsity bias discussed in the previous section produces ratings missing at random over items. But we also wish to test the robustness of metrics when some items lack more relevance judgments than others (Bellogín et al. 2017; Cañamares and Castells 2018). We consider two approaches to study item-dependent incompleteness conditions.

Random-item incompleteness Similar to the uniform rating incompleteness simulation described earlier, we create increasingly smaller random samples of the test set, but instead of sampling ratings at random, we sample whole items from the test set uniformly at random. Again, for each test size (now measured in percentage of items rather than ratings), we take 50 samples. Likewise, we compute the ranking of systems for a particular metric using each test sample, and we compute Kendall's τ coefficient with respect to the system ranking in the original full test set. And again, we average the correlations of the samples with the same size, which we take as an indication of the robustness of the metric to random item incompleteness.

Popular-item incompleteness An important difference between IR and recommender systems evaluation is that missing relevance judgments are not uniformly distributed which is commonly referred to as data *missing not at random* (Marlin et al. 2007; Steck 2010). The distribution of ratings in a recommendation scenario commonly follows a heavily skewed long tail distribution. This bias strongly affects the reliability of several IR metrics (Bellogín et al. 2017; Cañamares and Castells 2018). Previous works on recommender systems remove popular items to deal with the popularity bias (Cremonesi et al. 2010; Bellogín et al. 2017). To study how metrics are skewed towards popularity, we propose to build progressively smaller test sets removing the ratings of the most popular items. Again, we remove entire items from the test set, only by decreasing order of their number of test ratings (popularity). Then, we can study the change in the correlation between systems rankings of different subsets of the test set and the original test set, just as before. The higher the correlation, the higher the robustness of the metric to the popular item incompleteness.

4.2.3 User incompleteness

Finally, we propose to study the robustness of metrics to missing entire users in the relevance judgments. Symmetrically to missing items, we also consider two scenarios: users are missing at random, or users with the largest profiles are missing.

Random-user incompleteness When removing users at random, our assessment of the quality of the recommender systems becomes less reliable. In IR, this would be equivalent to removing the relevance judgments of random topics, which amounts to reducing the test query set—and hence the experiment size. We propose to follow an analogous approach to the random item incompleteness scenario, but removing all the ratings of random users instead of random items.

Large-user incompleteness Rating sparsity is also typically unevenly distributed over users, in a way that evolves over time. Cold start is a particularly challenging situation, where new recent users have to be delivered personalized recommendations with little available information about their tastes (Park et al. 2009). To analyze the robustness of metrics to cold user situations, we consider removing users with the largest profiles from the test set, and study the system ranking correlation with the original test set. This would be equivalent to removing the topics with a high number of relevance judgments in IR, i.e., the easy queries. Robust metrics to large-user incompleteness are ones that focus more on cold start users.

5 Experimental settings

In this section, we describe the experimental settings used in the conducted experiments to address the following questions that stem from our research objective:

Table 2 Datasets statistics

Dataset	Users	Items	Ratings	Density	User Gini	Item Gini
MovieLens 1M	6040	3706	1,000,209	4.468%	0.529	0.634
LibraryThing	7279	37,232	749,401	0.277%	0.493	0.581
BeerAdvocate	33,388	66,055	1,571,808	0.071%	0.868	0.865

RQ1) What is the discriminative power of the ranking metrics presented in Sect. 3?
RQ2) How robust are such metrics to different types of incompleteness (rating, user, item)?

The answers to these questions present important practical applications in the evaluation of top-N recommenders since they will guide the choice of the most appropriate evaluation metrics.

The experimental settings are presented as follows: first, we introduce the datasets (Sect. 5.1); then, in Sect. 5.2 we explain the details of the evaluation protocol; last, we provide a brief description of the recommendation algorithms used in the experiments in Sect. 5.3.

5.1 Datasets

We use three collections with explicit feedback in the form of 1-5 ratings: MovieLens 1M.[4], LibraryThing, and BeerAdvocate.[5] Table 2 shows the number of users, items, ratings, and the rating density this results into, i.e., the ratio of ratings over the total number of user-item pairs. To give an idea of how skewed the rating distribution is in each dataset, we show the Gini coefficient of the number of ratings over users and items. The Gini index measures the inequality of a distribution (a Gini index of 1 represents maximum inequality and 0 a perfect egalitarian systems) (Gini 1912).

We used three very different datasets in this study belonging to different domains (movies, books and beers). As it can be seen in Table 2, the collections exhibit different distributions of ratings, different ratios of ratings per user and item as well as different values of density. We think that this diversity is important to obtain generalizable results.

For evaluation purposes, we use a standard 5-fold cross-validation (Bellogín 2011). We might consider to use a more realistic evaluation that uses some kind of temporal training-test splitting (Campos et al. 2014), however, in the RS literature, random splits are much more commonly used (Bellogín 2011), in part because temporal information is not always available in public datasets, and even when it is available it may not realistically represent the user preferences (Harper and Konstan 2015). Because of this, we prefer to analyze robustness and discriminative power of evaluation metrics under the most common evaluation methodology used by researchers and practitioners for the sake of generality, which translates in performing random splits. Additionally, this methodology allows us to repeat

[4] https://grouplens.org/datasets/movielens.
[5] http://snap.stanford.edu/data/web-BeerAdvocate.html.

the experiments several times by performing cross-validation, which produces results with less variability and better statistical properties.

5.2 Evaluation protocol

There are several protocols for offline evaluation in Recommender Systems (Herlocker et al. 2004; Bellogín 2011; Bellogín et al. 2017). We follow the AllItems protocol which is considered a fair evaluation methodology and is similar to how systems are evaluated in IR (where no hold-out test set is available) (Bellogín 2011; Bellogín et al. 2017). In this protocol, the evaluated systems are required to rank all the items in the test set, except those already rated by the target user in the training set. An ideal recommender system would be able to achieve a perfect score in all the studied metrics. This evaluation protocol is also highly correlated to other variants (Bellogín 2011; Bellogín et al. 2017).

Most of the traditional IR ranking metrics rely on binary relevance: each item is either relevant or non-relevant for a given user. However, more modern metrics such as NDCG and ERR rely on graded relevance judgments. Since we use explicit feedback datasets in this work, we have to specify how to transform the ratings (a form of graded relevance) to binary relevance for those metrics that do not support graded relevance. In those cases, we set the relevance threshold τ to 4, considering as non-relevant every item rated below τ. The 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: their relevance is unknown.

5.3 Recommendation algorithms

To analyze the properties of evaluation metrics, we need a set of recommender systems for the metrics to be computed on their output. The IR community often uses the runs submitted to the TREC tracks for this purpose (Buckley and Voorhees 2004; Yilmaz and Aslam 2008; Lu et al. 2016). Since we do not have an equivalent in recommendation, we implement the following 21 recommendation methods and use their outputs to study the ranking metrics:

- **Random, Popularity**: basic non-personalized baselines.
- **CHI2**, **KLD**, **RSV**, **Rocchio's Weights** Valcarce et al. (2016): neighbourhood-based techniques that stem from Rocchio's feedback model.
- **RM1**, **RM2** Parapar et al. (2013): neighbourhood-based techniques that rely on relevance-based language models.
- **LM-WSR-UB**, **LM-WSR-IB** Valcarce et al. (2016): user-based and item-based approaches that compute neighborhoods with language models.
- **NNCosNgbr-UB**, **NNCosNgbr-IB**Cremonesi et al. (2010): user-based and item-based versions of a neighbourhood-based algorithm.
- **SLIM** Ning and Karypis (2011): sparse linear methods for recommendation.
- **HT** Yin et al. (2012): graph-based technique with emphasis on the long tail.
- **SVD, PureSVD, BPRMF, WRMF**Takács et al. (2009); Cremonesi et al. (2010); Rendle et al. (2009); Hu et al. (2008): matrix factorization techniques.
- **LDA** Blei et al. (2003): recommendation based on Latent Dirichlet Allocation.
- **PLSA** Hofmann (2004): recommendation based on Probabilistic Latent Semantic Analysis.

– **UIR-Item** Wang (2006): probabilistic user-item relevance model.

Since the goal of this paper is understanding the behavior of evaluation metrics on the most typical and representative recommendation algorithms, we believe the listed recommendation algorithms contain a varied selection of techniques from different Collaborative Filtering families that use the same information (the user-item rating matrix) and that proved to be competitive baselines in classical domains, such as movie or e-commerce recommendation.

6 Study of arithmetic mean metrics

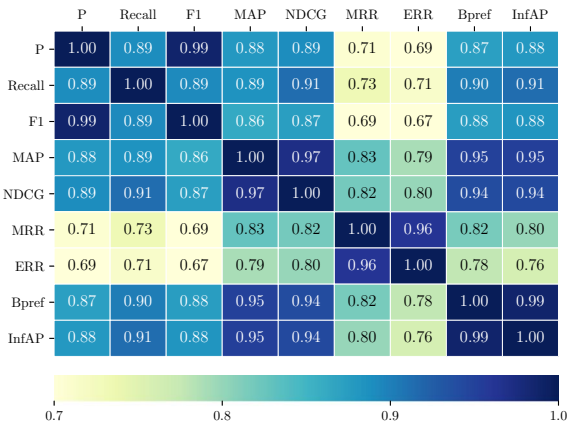
In this section, we study the metrics presented in Sect. 3. Given a metric, in IR a score is computed for each topic whereas in recommendation a score is computed for each user. The most common practice in both fields is to aggregate the user scores into a single number by computing the arithmetic mean. Therefore, in this section, we aggregate the scores for all users using the arithmetic mean.

In our previous work Valcarce et al. (2018), we found that metrics with deep cut-offs are more robust to the sparsity and popularity biases and have better discriminative power than metrics evaluated at shallow cut-offs. 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 (such as $n = 100$ recommendations) should be preferred in offline experiments. Note that such deep cut-off provides better properties even though many of the top n recommended items for each user may (and generally will) lack relevance judgments. Because of this, in the following experiments we use cut-offs of $n = 100$.

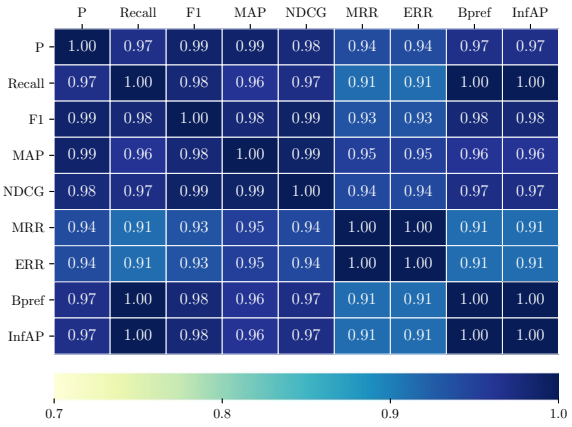
6.1 Correlation among metrics

We study the correlation among the system orderings according to different ranking metrics. Figure 1 shows the Kendall's τ correlation among metrics on the MovieLens 1M, LibraryThing and BeerAdvocate datasets. On the LibraryThing collection, all correlations are above 0.9 which indicates that the metrics produce almost identical rankings. On the other two datasets we observe stronger differences, with some correlations below 0.8.

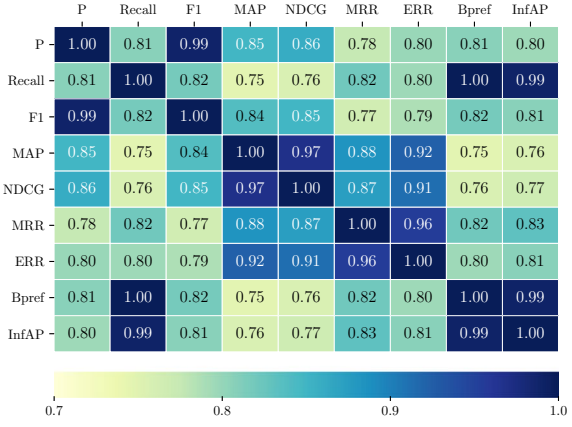
We can see that MRR and ERR differ noticeably from the rest of the metrics, especially on MovieLens 1M and LibraryThing. These two metrics are correlated with each other, which is consistent with the fact that ERR is a generalization of MRR based on a cascade user model (Chapelle et al. 2009). Bpref also shows a low correlation with the other metrics on the BeerAdvocate dataset. It is worth noting that Bpref is poorly correlated with MAP on this collection, which is a surprising result since Bpref was designed to correlate with MAP in IR (Buckley and Voorhees 2004). We suspect that this may be produced by the highly skewed long-tailed rating distribution over items in this dataset. Instead, MAP is strongly correlated with NDCG on the three datasets. Nevertheless, the rankings produced by the rest of the metrics show a fairly strong correlation between them.



a: MovieLens 1M.

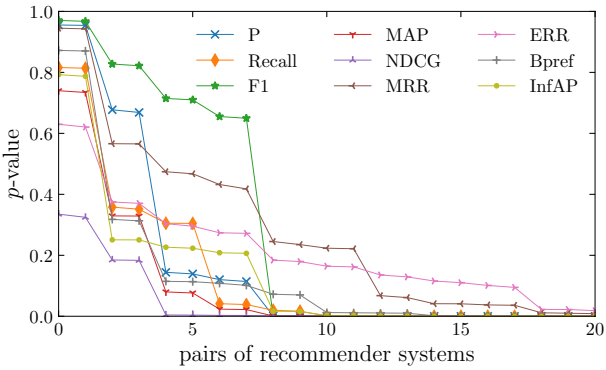


b: LibraryThing.

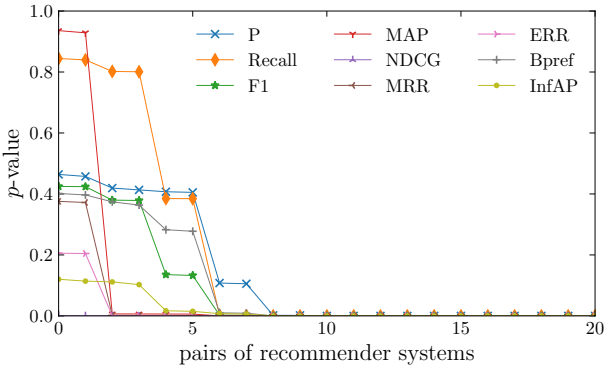


c: BeerAdvocate.

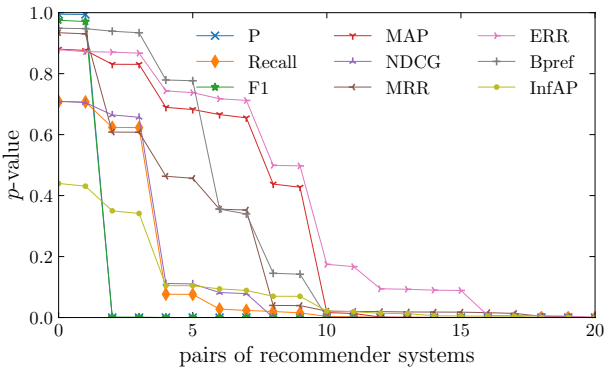
Fig. 1 Pairwise Kendall's τ correlation of P, Recall, F1, MAP, NDCG, MRR, ERR, Bpref and InfAP (at a cut-off of 100) on the MovieLens 1M, LibraryThing and BeerAdvocate datasets. Blue indicates higher correlation values whereas yellow corresponds to lower correlations (Color figure online)



a: MovieLens 1M.



b: LibraryThing.



c: BeerAdvocate.

Fig. 2 Analysis of the discriminative power of P, Recall, F1, MAP, NDCG, MRR, ERR, Bpref and InfAP (at a 100 cut-off) on the MovieLens 1M, LibraryThing and BeerAdvocate datasets

6.2 Discriminative power

Figure 2 shows our findings in terms of the discriminative power of the different studied metrics to provide an answer to RQ1. We only plot the 20 largest p-values (out of 210 pairs) to facilitate the visualization of the curves. We also present the values of DP

Table 3 DP values (lower is better) of P, Recall, F1, MAP, NDCG, MRR, ERR, Bpref and InfAP (at a 100 cut-off) on the MovieLens 1M (ML), LibraryThing (LT) and BeerAdvocate (BA) datasets

Col.	P	Recall	F1	MAP	NDCG	MRR	ERR	Bpref	InfAP
ML	3.774	3.066	6.316	2.335	1.044	6.063	4.603	3.000	2.984
LT	2.783	4.059	1.879	1.889	0.001	0.753	0.411	2.115	0.493
BA	1.989	2.909	1.949	7.009	3.119	4.940	8.127	6.310	2.181
Total	8.546	10.034	10.144	11.233	4.164	11.756	13.141	11.425	5.658

(which amounts to the area under the p-value curve) in Table 3. Although the results vary across datasets, we can find some general trends.

Overall, NDCG shows the highest discriminative power. NDCG is ranked first on MovieLens and LibraryThing. Although it is fifth in the BeerAdvocate dataset, the actual value of DP is quite small and very similar to more discriminative metrics. Note that the sparser the dataset the less reliable the discriminative power measurement is. In this case, NDCG is ranked lower on BeerAdvocate, which is the sparsest dataset.

Precision, Bpref and InfAP have an overall average behavior in terms of discrimination, though Bpref suffers some discrimination loss in BeerAdvocate—perhaps by the difficulty of this metric to cope with the high sparsity of this dataset, as we discuss in Sect. 4.2.1. Finally, MAP, Recall, and MRR display a somewhat erratic performance in terms of discriminative power depending on the dataset.

To check whether these results are dependent of the random samples produced by the 5-fold cross-validation approach, we generated different folds and computed the DP values. The results between different runs of 5-fold cross-validation were very similar and we only found small differences that do not contradict the findings presented in this section.

The fair discriminative power of Precision is somewhat better than what was reported in prior work (Buckley and Voorhees 2000; Sakai 2006). This can be partly explained by a difference in metric depth: we study P@100, while prior work often took a full depth of 1000, and a drop in the discriminative power of Precision beyond depth 200 has been indeed reported by other authors (see e.g., Webber et al. 2010). Buckley and Voorhees Buckley and Voorhees (2000) did find a slightly lower discriminative power for P@100 compared to MAP. Even though they used a quite different methodological approach (not involving statistical significance tests) than has been used in later work (Sakai 2006; Webber et al. 2010), our findings are quite in line with theirs: MAP shows overall a slightly higher discriminative power than Precision. We see an exception in BeerAdvocate, which we attribute to the extremely low judgment density of this dataset, to which Precision seems to be more robust than most other metrics—a property that we will further analyze in the next section. On the other hand, the slightly lower discriminative power of Bpref compared to MAP and nDCG is in line with prior studies on TREC data (Sakai and Kando 2008).

The metrics that use graded relevance (NDCG and ERR) seem to discriminate better among systems on collections with more "*hard users*". We call hard users (as in hard queries Voorhees 2005) to those for whom it is difficult to locate relevant items because of the small size of their test split profiles. On datasets with a greater concentration of those users, the graded information tips more the balance to the metrics exploiting the grades. In this situation, graded metrics can produce more differentiate results. For instance, when two systems a and b find one relevant item each on position 5 for the user, only the graded metrics will produce different performance values when the item found by system a was

rated higher than the one found by *b*. This can be observed to some degree on the LT dataset, where the concentration of test users with short profiles is higher than in the other datasets. Moreover, LT presents a grading pattern more prone to high values than the other collections (31,48% of grades higher than 4 compared to 22.63% of ML and 26.10% of BA), which also seems to contribute to ERR and NDCG producing low DP values. This type of data also affects MRR but in a different way. In the case of MRR, having very few relevant items on test split makes the position of the first relevant item on the systems rankings more variable. In that way, MRR produces more diverse metric values and so more discriminative results

6.3 Robustness to incompleteness

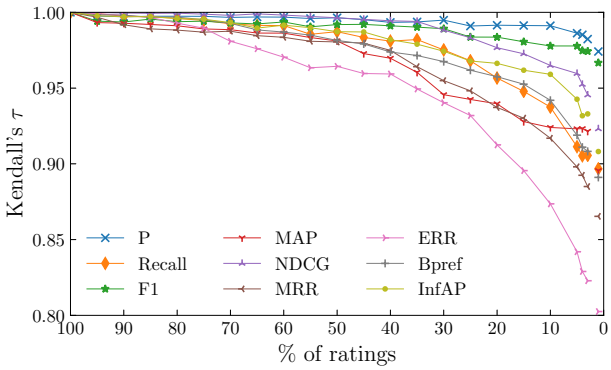
We now study the robustness of the metrics to different types of incompleteness, to answer our second research question.

6.3.1 Rating incompleteness

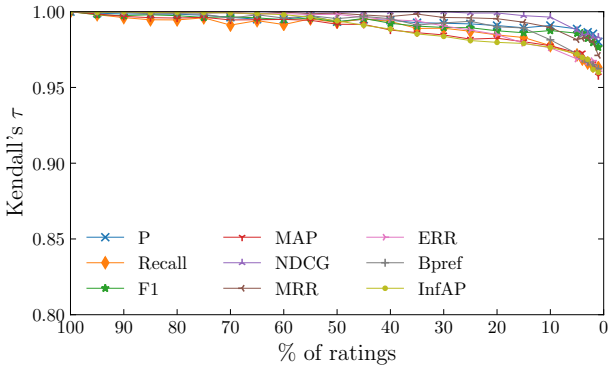
Figure 3 presents the results of the experiments regarding robustness to rating incompleteness. We can see that all the metrics are fairly robust to random rating sparsity since the correlation is above 0.9 even when removing half of the test set. Precision, F1, and NDCG show the best robustness trends on the three datasets (Precision especially on BeerAdvocate). In contrast, ERR, Bpref, InfAP, Recall, and MRR show poor robustness. Some of these results are interesting because they present a different situation to what has been commonly observed in IR. Regarding Precision, we can relate its good robustness to the fact that the amount of variation of this metric seems to be constant even at extreme sparsity levels. This is further illustrated in Fig. 4, where we see that the absolute value of Precision decreases quite slowly and linearly with the amount of removed ratings, whereas other metrics decrease much faster, and in a sublinear fashion. A moderate room for loss of self-correlation in Precision results from this, compared to other metrics.

On the other hand, Bpref and InfAP were conceived for dealing with incomplete judgments (Buckley and Voorhees 2004; Yilmaz and Aslam 2008), but in top-N recommendation they would seem less robust than other metrics. Bpref and InfAP were designed for approximating MAP with incomplete judgments, while in recommendation MAP is not such a golden standard. Still, it is surprising that MAP shows better robustness than Bpref and InfAP on LibraryThing and BeerAdvocate.

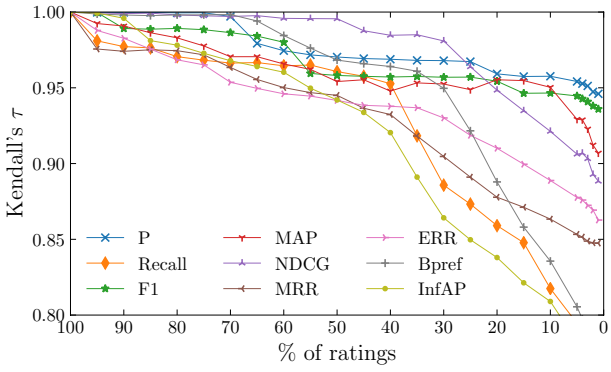
We believe this is because Bpref and InfAP ignore unjudged documents, and this is virtually equivalent to having all judged documents packed at the top of the ranking, whereas MAP, for instance, is sensitive to the number of unjudged documents in between relevant ones (as this determines the position of relevants). The removal of highly ranked judgments produces larger relative variations in these metrics than the removal at lower ranks. When judgments (test ratings) are highly sparse (as is the case in common recommendation data), judged items can get very far apart from each other in the ranking, and the virtual rank for Bpref and InfAP can highly differ from the actual ranking that all other metrics take into account—this may explain why Bpref and InfAP diverge faster than MAP in Fig. 3. Moreover, the removal of non-relevant judgments affects these two metrics, but none of the others, which results in an additional source of higher variance (and hence correlation loss) for Bpref and InfAP compared to MAP and other metrics.



a: MovieLens 1M.



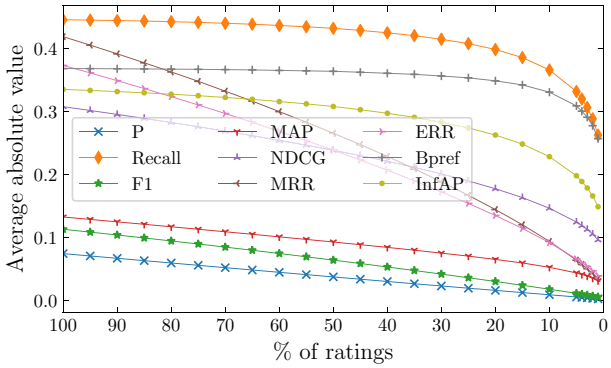
b: LibraryThing.



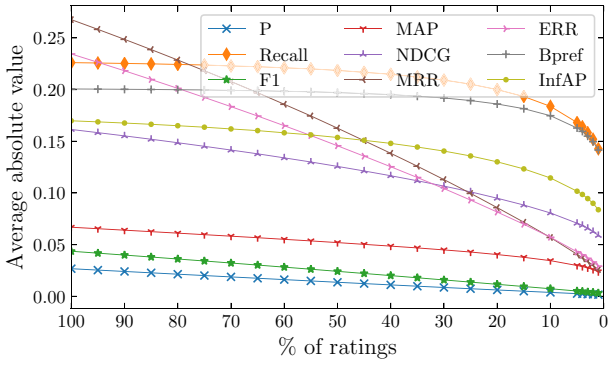
c: BeerAdvocate.

Fig. 3 Kendall's τ self-correlation of P, Recall, F1, MAP, NDCG, MRR, ERR, Bpref and InfAP (at a 100 cut-off) when removing ratings uniformly at random from the test set on the MovieLens 1M, LibraryThing and BeerAdvocate collections

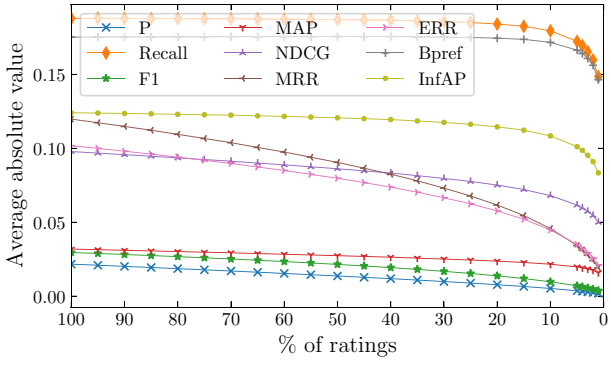
This can explain why our results differ from earlier experiments with pooled TREC judgments (Buckley and Voorhees 2004; Yilmaz and Aslam 2008). In our experiments, the "100%" point in the x axis for robustness (in Fig. 3) represents a highly sparse situation already: the average test rating coverage of the top 100 of the evaluated systems is 5%, 2.2% and 1.4% respectively in MovieLens, LibraryThing and BeerAdvocate. In contrast, in the experiments on missing judgments with TREC data (Buckley and Voorhees 2004;



a: MovieLens 1M.



b: LibraryThing.



c: BeerAdvocate.

Fig. 4 Averaged absolute values of P, Recall, F1, MAP, NDCG, MRR, ERR, Bpref and InfAP (at a 100 cut-off) when removing ratings uniformly at random from the test set on the MovieLens 1M, LibraryThing and BeerAdvocate collections

Yilmaz and Aslam 2008) the starting point is 100% complete judgments. Hence the virtual (Bpref and InfAP) and actual (MAP) ranking positions take much longer to get far apart as judgments are removed, compared to our experiments, where they are very far apart from the beginning. Furthermore, prior work often uses smoothing options in the metrics or the setup (Buckley and Voorhees 2004; Yilmaz and Aslam 2008), which may likely reduce their variance (and hence increase their robustness) in low density conditions. Bpref and

InfAP display their best robustness in the MovieLens dataset, where Bpref is comparable to MAP, and InfAP is slightly better. This is the denser dataset of all three, which may account for a milder instability in these metrics.

On the other hand, utility-based metrics such as MRR and ERR are found to rank among the least robust metrics (except in LibraryThing) in our experiments. This is an interesting finding that complements previous studies (Lu et al. 2016) where ERR was found to be quite stable but in a different dimension from what we are studying here: these metrics are stable regarding the cutoff depth but, as we find here, they are not particularly robust to missing ratings in recommendation.

6.3.2 Item incompleteness

We now study the robustness of the metrics regarding item incompleteness. First, we remove random items and, second, we remove the most popular items from the test set.

Random item incompleteness When removing random items from the test set, we obtain similar results as when removing random ratings (compare Fig. 5 against Fig. 3). However, the drop in correlation is more pronounced when removing whole items instead of individual ratings from the test set.

Popular item incompleteness We report the results of the experiments of robustness to the popularity bias (i.e., removing the most popular items) in Fig. 6. 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-tailed ratings distribution in 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 and ERR are the worst. F1 and NDCG also present moderately good results. The robustness to the popularity bias of the rest of the metrics depends heavily on the dataset and it is difficult to draw conclusions about them.

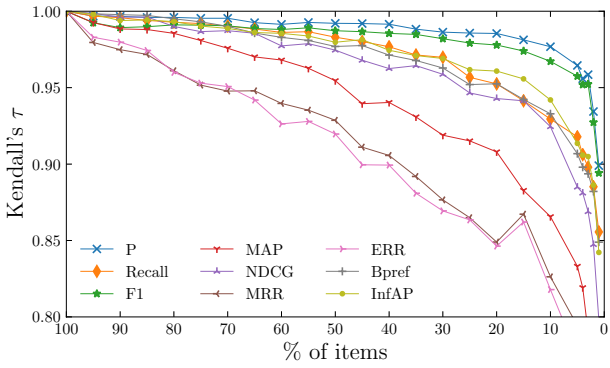
6.3.3 User incompleteness

The last set of experiments regarding robustness to incompleteness consist in removing users from the test set. As described in Sect. 4.2, on the one hand, we remove random users to test how the number of users in the test set affects the robustness of the metrics. On the other hand, we remove users with the largest profiles. In this way, we can study how metrics assess the performance of algorithms for users who are difficult to recommend to.

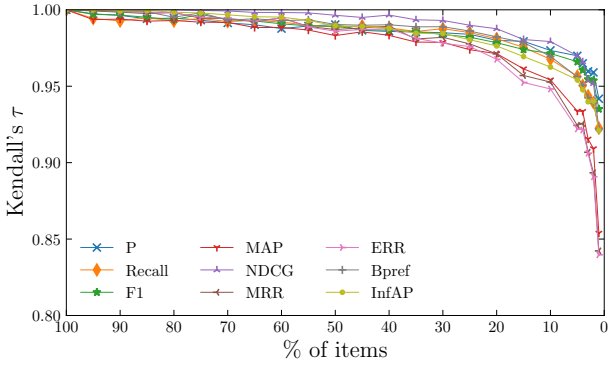
Random user incompleteness Figure 7 shows the robustness of the metrics to random user incompleteness. We can observe similar results to those obtained when removing random ratings (Fig. 3) and random items (Fig. 5), but with a much smoother decrease in correlation. Overall, the studied metrics are very robust to test sets with fewer users.

Large user incompleteness We now study the robustness of the metrics to the removal of users with large profiles. We present the results of these experiments in Fig. 8. In this scenario, MRR, ERR and, to a lesser degree, InfAP and Bpref show the best robustness. In contrast, metrics such as Precision, F1 or NDCG, that showed good results in the previous experiments, present now lower robustness. Therefore, these metrics are more sensitive to the behavior of users with large profiles. In the next section, we study an alternative formulation of these metrics that is more sensitive to users with small profiles.

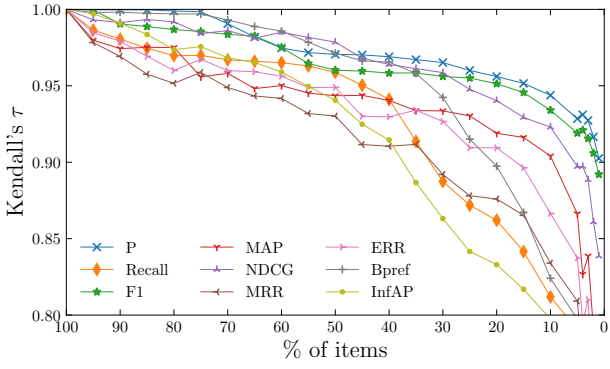
We think that the reason for MRR (and ERR as a highly correlated metric) to gain robustness with this missing rating distribution—compared to the ones analyzed in the



a: MovieLens 1M.



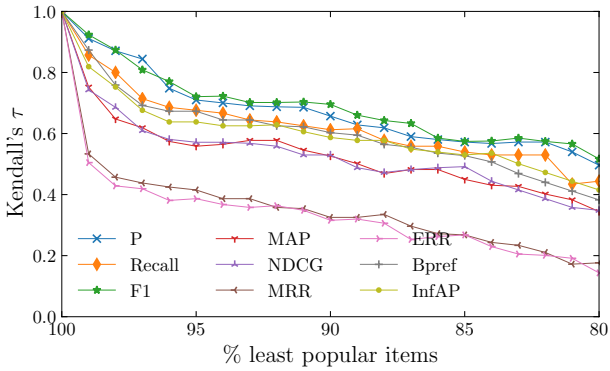
b: LibraryThing.



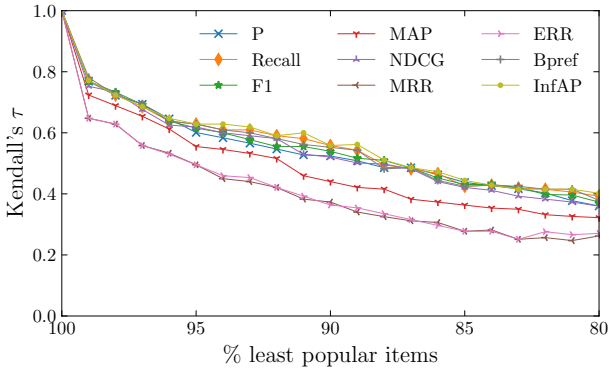
c: BeerAdvocate.

Fig. 5 Kendall's τ self-correlation of P, Recall, F1, MAP, NDCG, MRR, ERR, Bpref and InfAP (at a cut-off of 100) when removing items uniformly at random from the test set on the MovieLens 1M, LibraryThing and BeerAdvocate collections

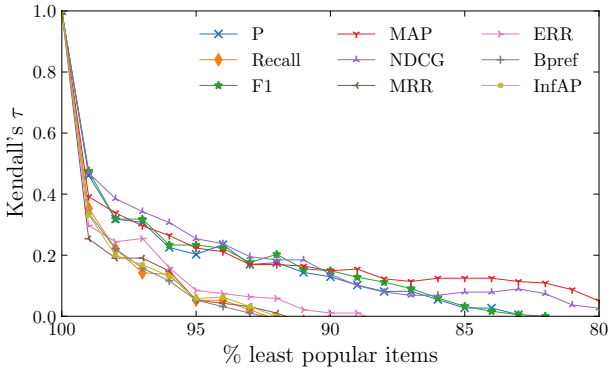
previous sections—is that a large user profile (providing many relevance judgments) leaves less room for differences in the position of the top ranked relevant item, compared to sparser profiles. The comparison between two systems in MRR is therefore played in the small user profiles. As a consequence, removing large user profiles changes fewer system comparisons, and MRR catches up on stability compared to other metrics that are either sensitive to the position of all relevant documents (not just the top one), or



a: MovieLens 1M.



b: LibraryThing.

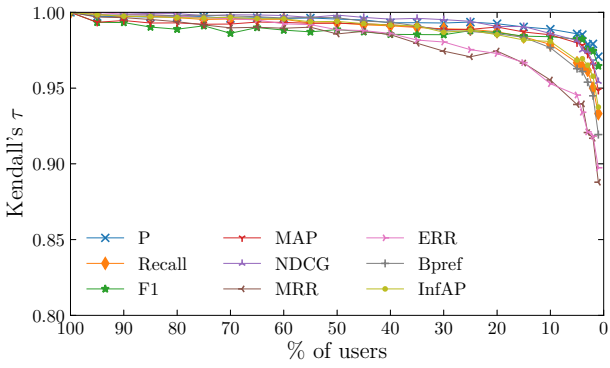


c: BeerAdvocate.

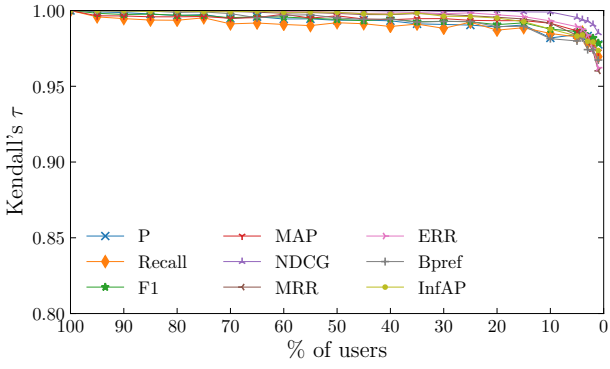
Fig. 6 Kendall's τ self-correlation of P, Recall, F1, MAP, NDCG, MRR, ERR, Bpref and InfAP (at a cut-off of 100) when removing the most popular items from the test set on the MovieLens 1M, LibraryThing and BeerAdvocate collections

are insensitive to all positions (e.g., Precision). This effect is less clear in LibraryThing, where the user profile size distribution is "flatter" (as evidenced by a lower User Gini index in Table 2).

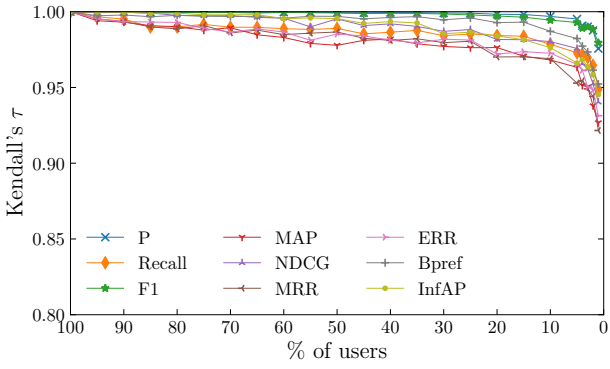
Based on these results, since we cannot fully answer RQ2, we propose to change how metrics are averaged and analyze the resulting behavior in the next section.



a: MovieLens 1M.



b: LibraryThing.

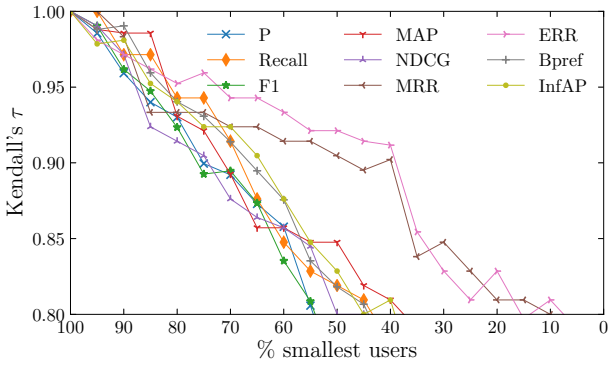


c: BeerAdvocate.

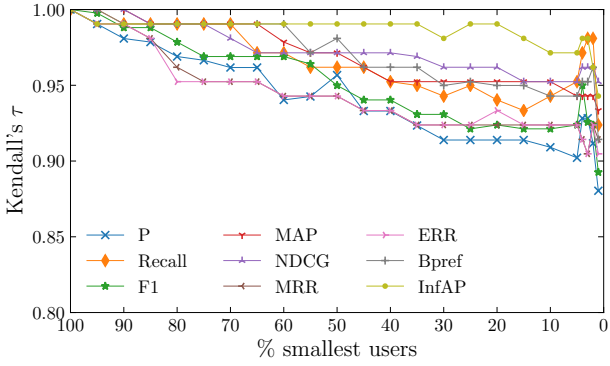
Fig. 7 Kendall's τ self-correlation of P, Recall, F1, MAP, NDCG, MRR, ERR, Bpref and InfAP (at a cut-off of 100) when removing random users from the test set on the MovieLens 1M, LibraryThing and BeerAdvocate collections

7 Study of geometric mean metrics

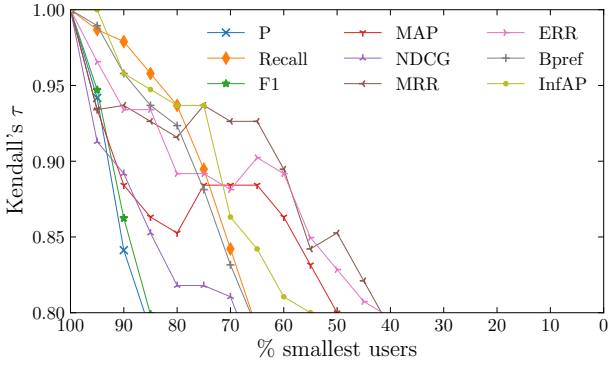
In the previous section, we studied the metrics presented in Sect. 3 using the arithmetic mean to aggregate the metric values for all the users in the dataset. We have seen that some metrics (especially Precision, F1, and NDCG) are more robust than others in most of the scenarios except for the case when we remove the users with the largest profiles. In that



a: MovieLens 1M.



b: LibraryThing.



c: BeerAdvocate.

Fig. 8 Kendall's τ self-correlation of P, Recall, F1, MAP, NDCG, MRR, ERR, Bpref and InfAP (at a cut-off of 100) using the test set when removing the users with the largest profiles on the MovieLens 1M, LibraryThing and BeerAdvocate collections

case, most metrics fall short in robustness. We now re-examine this property when the metric values are aggregated over users using the geometric mean.

The geometric mean has been used in IR to average the values of a metric for different topics; for example, in the TREC Robust track (Voorhees 2005), GMAP (the geometric version of MAP Robertson 2006) is used as the reference metric. The problem of the arithmetic mean is that differences in the scores for well-performing (i.e., easy) topics

overshadow differences in the scores for difficult topics. In contrast, the geometric mean emphasizes the scores close to zero while it minimizes the contribution of larger scores. An intuitive definition of the geometric mean is that it is equivalent to calculating the arithmetic mean of the logs of the metric scores (thus dampening the importance of higher scores).

We, therefore, revisit the robustness (and for completeness, all of the) analysis of the previous section under the light of the nuances that may be uncovered by the geometric mean. Note that the discriminative power of the metrics is the same since the averaging function does not affect paired difference tests. This is because these statistical tests look at the individual values of the metrics in a user basis and disregard the aggregated value.

7.1 Correlation among metrics

Figure 9 shows the pairwise correlation between the ranking metrics when they are averaged using the geometric mean. We observe that all the (geometric) metrics are strongly correlated between each other except for GF1 and GMRR, which tend to obtain low values with respect to any other metric except with each other. For example, in LibraryThing, GF1 even shows negative correlations with respect to all metrics but GMRR. This result contrasts with what we observed in Sect. 6.1 with the arithmetic mean, where F1 strongly correlated with all metrics except MRR (and ERR).

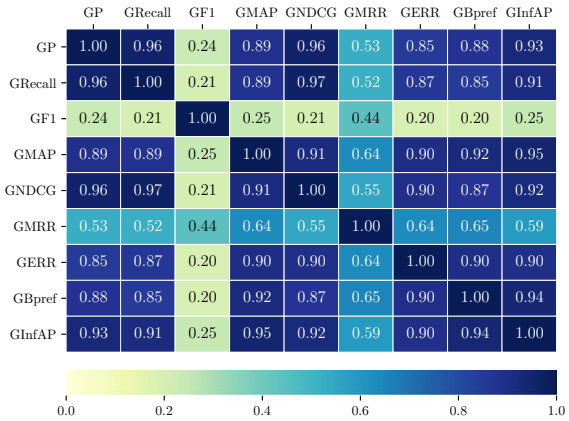
We hypothesize that this behavior is related to the reciprocal components in these metrics. When applying the geometric mean to such reciprocal components the obtained results show very different trends than for the rest of the metrics. In fact, we computed the pairwise correlation between the arithmetic-mean and geometric-mean version of each metric. Whereas most of the metrics show a strong correlation between their arithmetic and geometric variations (for instance, MAP and GMAP have a correlation of 0.92 in MovieLens) this is not true for GF1 and GMRR (e.g., the correlation between F1 and GF1 is only 0.42 in MovieLens). These results evidence a drastic change in the behavior of these metrics when using the geometric mean.

7.2 Robustness to incompleteness

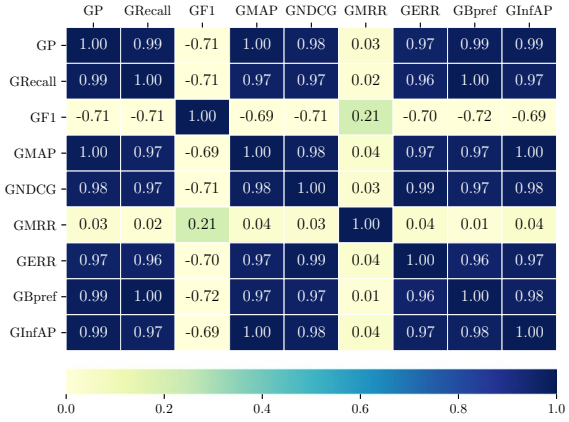
We repeat the experiments of robustness to incompleteness with the metrics averaged with the geometric mean. We study the robustness to incomplete ratings (Fig. 10), to missing items at random (Fig. 11), to missing popular items (Fig. 12), to users missing at random (Fig. 13), and to missing largest users (Fig. 14).

In all the scenarios, GF1 and GMRR are now the least robust metrics. Overall, we can see similar trends in missing ratings, items missing at random and users missing at random. When removing the most popular items, the correlations quickly drop. Geometric Precision and GNDCG exhibit good robustness in all those cases. Additionally, we observe that GP and GNDCG are more robust than P and NDCG when removing the users with large user profiles. Therefore, averaging these metrics with the geometric mean can be useful when we aim to be robust to this phenomenon.

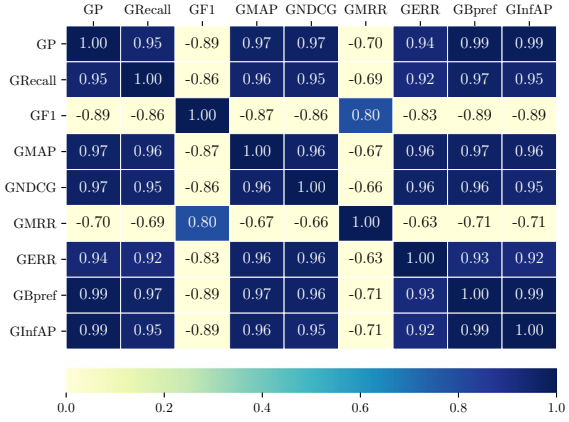
Therefore, as an answer to RQ2, we can summarize our results as follows: standard Precision and NDCG metrics (averaged by the arithmetic mean) obtain consistently good robustness values in all the analyzed scenarios, except when users are removed according to their profile size, where their geometric counterparts show very good robustness. On the other hand, the least robust metrics tend to be those that in some way "discard" some amount of information, namely MRR (which does not care about



a: MovieLens 1M.

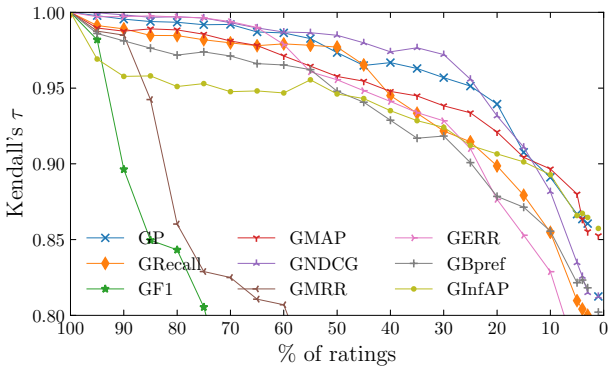


b: LibraryThing.

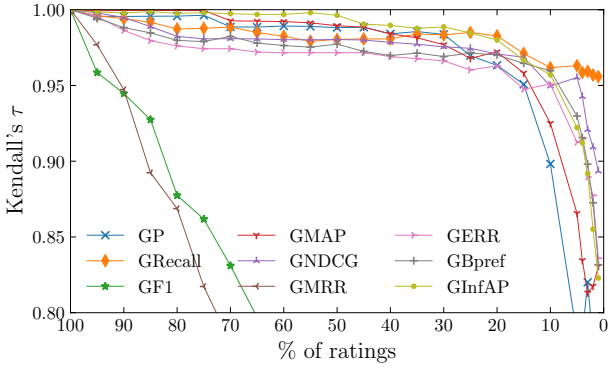


c: BeerAdvocate.

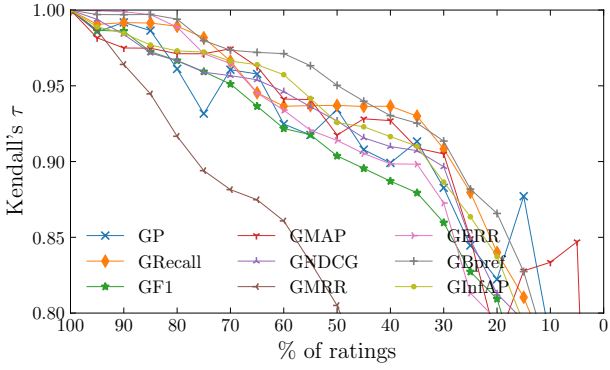
Fig. 9 Pairwise Kendall's τ correlation between GP, GRecall, GF1, GMAP, GNDCG, GMRR, GERR, GBpref, and GInfAP (at a cut-off of 100) on the MovieLens 1M, LibraryThing and BeerAdvocate datasets. Blue indicates higher correlation values whereas yellow corresponds to lower correlations (Color figure online)



a: MovieLens 1M.



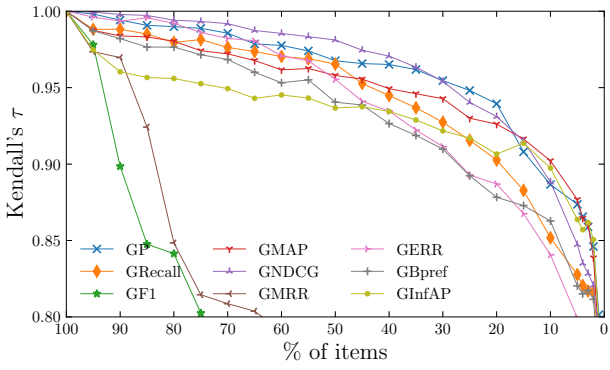
b: LibraryThing.



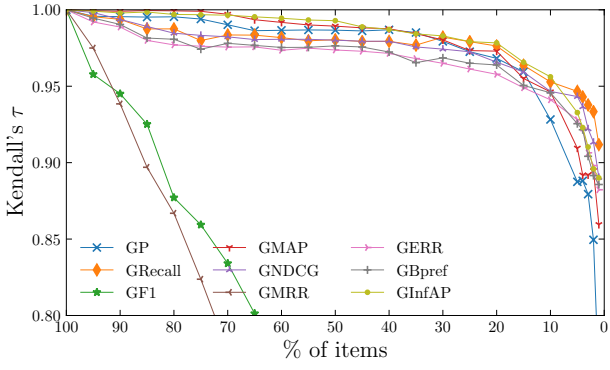
c: BeerAdvocate.

Fig. 10 Kendall's τ self-correlation of GP, GRecall, GF1, GMAP, GNDCG, GMRR, GERR, GBpref and GInfAP (at a cut-off of 100) when removing ratings uniformly at random from the test set on the MovieLens 1M, LibraryThing and BeerAdvocate collections

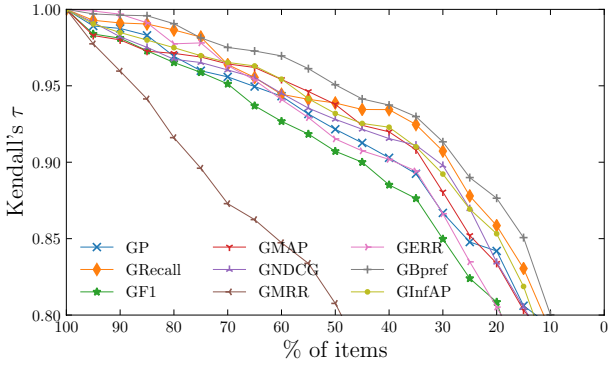
relevant documents ranked after the first one) and ERR (which is smoother than MRR aggressively top-heavy). It is interesting to observe that, except for robustness to large user profiles, the results tend to be quite consistent in all the tested scenarios, regarding robustness with respect to either ratings, or items, or users.



a: MovieLens 1M.



b: LibraryThing.

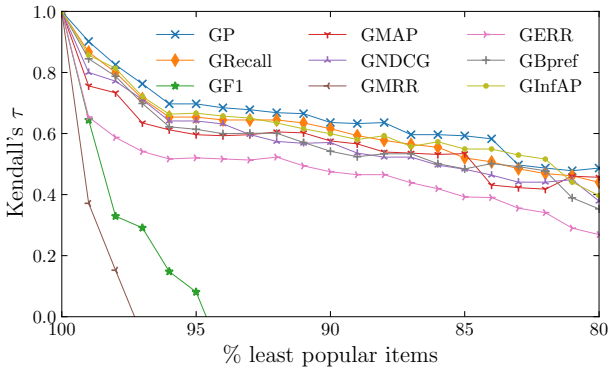


c: BeerAdvocate.

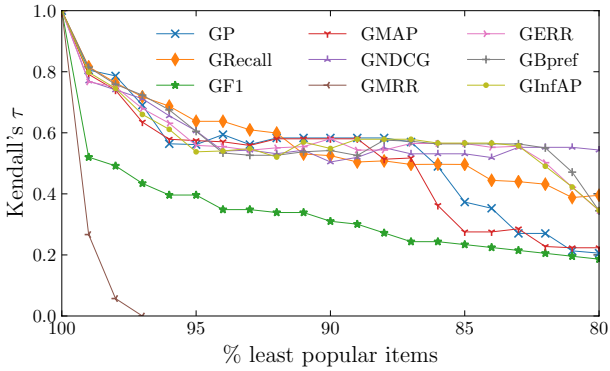
Fig. 11 Kendall's τ self-correlation of GP, GRecall, GF1, GMAP, GNDCG, GMRR, GERR, GBpref and GInfAP (at a cut-off of 100) when removing items uniformly at random from the test set on the MovieLens 1M, LibraryThing and BeerAdvocate collections

8 Conclusions

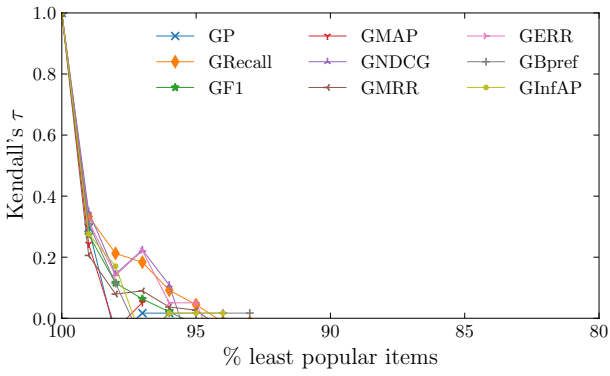
We have studied the properties of common IR metrics when applied to the offline evaluation of recommender systems, under the perspective of recommendation as a ranking task. Our research focused, in particular, on discriminative power and the robustness against incomplete relevance knowledge, as desirable properties of evaluation metrics



a: MovieLens 1M.



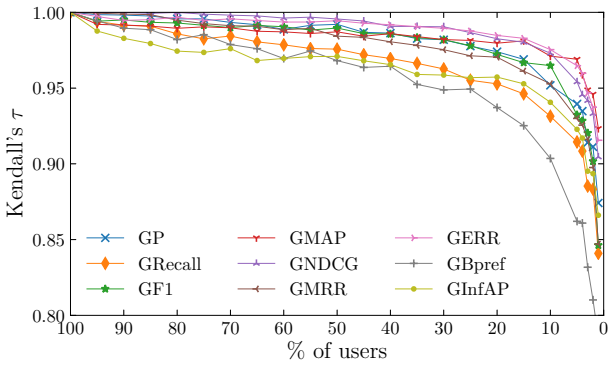
b: LibraryThing.



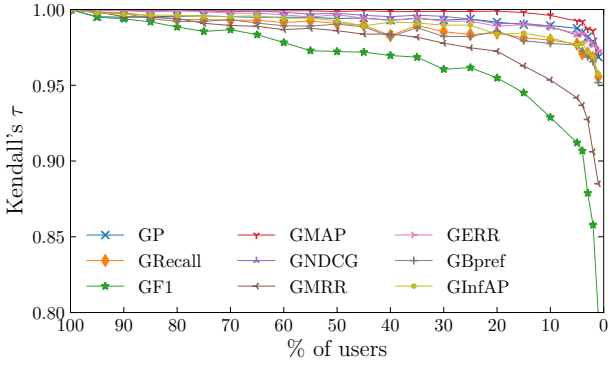
c: BeerAdvocate.

Fig. 12 Kendall's τ self-correlation of GP, GRecall, GF1, GMAP, GNDCG, GMRR, GERR, GBpref and GInfAP (at a cut-off of 100) when removing the most popular items from the test set on the MovieLens 1M, LibraryThing and BeerAdvocate collections

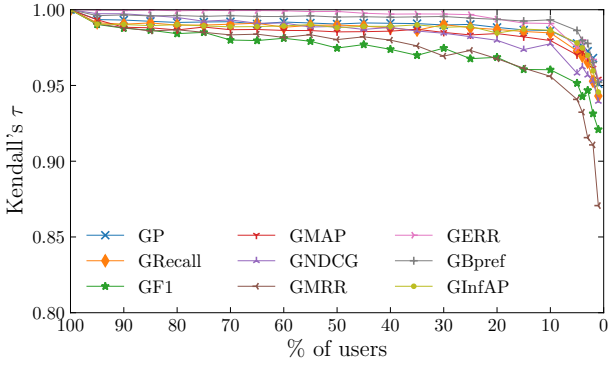
that have been the object of attention in the IR field, mainly in the context of search tasks (Lu et al. 2016; Sakai 2006). The dimensions of our comparative analysis include a set of common and representative metrics, and the metric averaging function over users. Along with providing insights into the properties and behavior of different metrics and



a: MovieLens 1M.



b: LibraryThing.

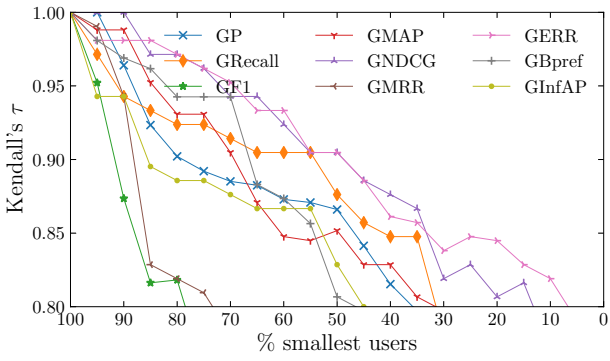


c: BeerAdvocate.

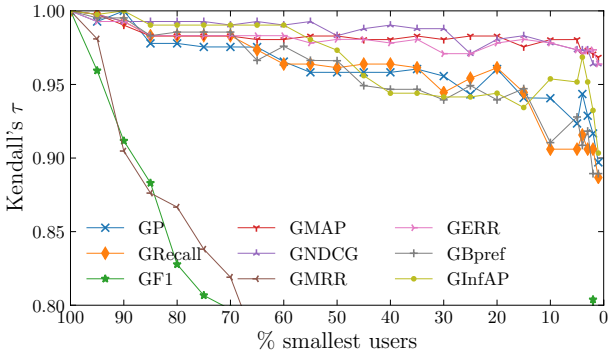
Fig. 13 Kendall's τ self-correlation of GP, GRecall, GF1, GMAP, GNDCG, GMRR, GERR, GBpref and GInfAP (at a cut-off of 100) when removing users uniformly at random from the test set on the MovieLens 1M, LibraryThing and BeerAdvocate collections

their specifics in the recommendation task, our findings have practical implications in the evaluation of recommender systems using ranking-oriented metrics.

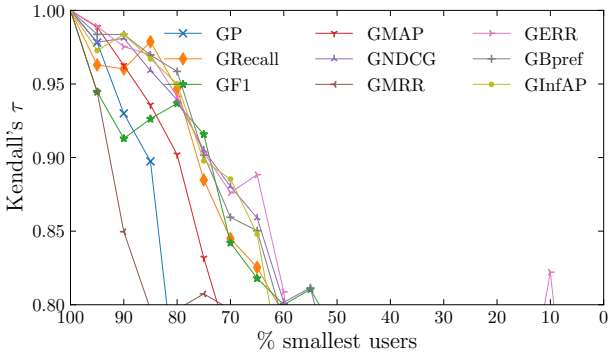
Our analysis suggests that Precision, a simple binary metric, is very robust to sparsity and popularity biases. Normalized Discounted Cumulative Gain also displays high robustness to the sparsity bias and moderate robustness to the popularity bias. In terms of discriminative power, NDCG returns the most consistent results across the tested datasets,



a: MovieLens 1M.



b: LibraryThing.



c: BeerAdvocate.

Fig. 14 Kendall's τ self-correlation of GP, GRecall, GF1, GMAP, GNDCG, GMRR, GERR, GBpref and GInfAP (at a cut-off of 100) using the test set when removing the users with the largest profiles on the MovieLens 1M, LibraryThing and BeerAdvocate collections

while Precision ranks as a mid-packer but reasonably stable metric in this aspect. On the other end, MRR and ERR rank among the lowest in our analysis of robustness, perhaps as a consequence of their extreme top-heaviness. We found, on the other hand, that Bpref and InfAP—which were proposed to address incompleteness in IR—would seem to fail their purpose (robustness to missing ratings) when evaluating recommendations over common publicly available datasets: they can happen to be even more inconsistent (e.g., on the sparse BeerAdvocate dataset) than the metric (MAP) they are intended to improve. Finally,

geometric means are a convenient way to identify systems that are effective with all users, not only the easy ones with large profiles. These results have notable practical implications in the choice of the appropriate ranking metrics since we have compared the goodness of each metric with respect to robustness to incompleteness and discriminative power.

Recommender system evaluation is still an open area and many directions lie ahead to continue our research. Metrics beyond ranking accuracy can be studied: for instance, diversity and novelty (Castells et al. 2015) have been recognized as an important dimension in both information retrieval and recommender systems (Hosanagar et al. 2014; Yu et al. 2017), and it would be interesting to analyze which diversity and novelty metrics provide better robustness or discriminative power. On the other hand, recommender system experiments have a considerable number of design options that can be varied (Bellogín 2011; Bellogín et al. 2017). In this paper, we have taken specific settings, such as the AllItems approach and random splitting. It would be interesting to explore whether the outcomes of our analysis might change in any way for different settings, such as temporal rating splits, or different target item selection approaches (Campos et al. 2014; Bellogín 2011). This work also opens avenues of future investigation on the theoretical properties on the studied metrics that justify the observed behavior.

References

Allan, J., Croft, B., Moffat, A., & Sanderson, M. (2012). Frontiers, challenges, and opportunities for information retrieval: Report from SWIRL 2012 the second strategic workshop on information retrieval in lorne. *SIGIR Forum*, *46*(1), 2–32. https://doi.org/10.1145/2215676.2215678.

Anderson, C. (2008). *The Long Tail: Why the Future of Business Is Selling Less of More*. New York: Hachette Books.

Beel, J., & Langer, S. (2015). A comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems. In S. Kapidakis, C. Mazurek, M. Werla (eds.) *Proceedings of the 19th international conference on theory and practice of digital libraries*, TPDL '15 (pp. 153–168). Springer, Cham. https://doi.org/10.1007/978-3-319-24592-8_12.

Belkin, N. J., & Croft, W. B. (1992). Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, *35*(12), 29–38. https://doi.org/10.1145/138859.138861.

Bellogín, A., Castells, P., & Cantador, I. (2011). Precision-oriented evaluation of recommender systems. In *Proceedings of the 5th ACM conference on recommender systems, RecSys '11* (p. 333). ACM, New York, NY, USA. https://doi.org/10.1145/2043932.2043996.

Bellogín, A., Castells, P., & Cantador, I. (2017). Statistical biases in information retrieval metrics for recommender systems. *Information Retrieval Journal*, *20*(6), 606–634. https://doi.org/10.1007/s10791-017-9312-z.

Bellogín, A., Wang, J., & Castells, P. (2013). Bridging memory-based collaborative filtering and text retrieval. *Information Retrieval*, *16*(6), 697–724. https://doi.org/10.1007/s10791-012-9214-z.

Bennett, J., & Lanning, S (2007). The netflix prize. In *Proceedings of the KDD cup workshop 2007* (pp. 3–6). ACM, New York, NY, USA.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Buckley, C., Dimmick, D., Soboroff, I., & Voorhees, E. (2007). Bias and the limits of pooling for large collections. *Information Retrieval*, *10*(6), 491–508. https://doi.org/10.1007/s10791-007-9032-x.

Buckley, C., & Voorhees, E.M (2000). Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '00* (pp. 33–40). ACM, New York, NY, USA. https://doi.org/10.1145/345508.345543.

Buckley, C., & Voorhees, E.M. (2004). Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '04* (pp. 25–32). ACM, New York, NY, USA. https://doi.org/10.1145/1008992.1009000.

Büttcher, S., Clarke, C.L.A., Yeung, P.C.K., & Soboroff, I. (2007). Reliable information retrieval evaluation with incomplete and biased judgements. In *Proceedings of the 30th annual international ACM*

SIGIR conference on research and development in information retrieval, SIGIR '07, (pp. 63–70). ACM, New York, NY, USA. https://doi.org/10.1145/1277741.1277755.

Campos, P. G., Díez, F., & Cantador, I. (2014). Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Model. User-Adapt. Interact.*, *24*(1–2), 67–119. https://doi.org/10.1007/s11257-012-9136-x.

Cañamares, R., & Castells, P. (2018). Should i follow the crowd? A probabilistic analysis of the effectiveness of popularity in recommender systems. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18* (pp. 415–424). ACM, New York, NY, USA. https://doi.org/10.1145/3209978.3210014.

Castells, P., Hurley, N.J., & Vargas, S. (2015). Novelty and diversity in recommender systems. In F. Ricci, L. Rokach, B. Shapira (eds.) *Recommender Systems Handbook*, 2nd edn. (pp. 881–918). Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7637-6_26.

Chapelle, O., Metlzer, D., Zhang, Y., & Grinspan, P. (2009). Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on information and knowledge management, CIKM '09* (p. 621). ACM, New York, NY, USA. https://doi.org/10.1145/1645953.1646033.

Cremonesi, P., Garzotto, F., Negro, S., Papadopoulos, A.V., & Turrin, R. (2011). Looking for "Good" recommendations: A comparative evaluation of recommender systems. In *Proceedings of the 13th IFIP TC 13 international conference on human-computer interaction–Part III, INTERACT '11* (pp. 152–168). Springer, Berlin. https://doi.org/10.1007/978-3-642-23765-2_11.

Cremonesi, P., Garzotto, F., & Turrin, R. (2013). User-centric vs. system-centric evaluation of recommender systems. In *Proceedings of the 14th IFIP TC 13 international conference on human-computer interaction–Part III, INTERACT '13* (pp. 334–351). Springer, Berlin. https://doi.org/10.1007/978-3-642-40477-1_21.

Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-N recommendation tasks. In *Proceedings of the 4th ACM conference on recommender systems, RecSys '10* (pp. 39–46). ACM, New York, NY, USA. https://doi.org/10.1145/1864708.1864721.

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap, monographs on statistics and applied probability*. Boca Raton, Florida, USA: Chapman & Hall/CRC.

Ferro, N., Fuhr, N., Grefenstette, G., Konstan, J. A., Castells, P., Daly, E. M., et al. (2018). The dagstuhl perspectives workshop on performance modeling and prediction. *SIGIR Forum*, *52*(1), 91–101. https://doi.org/10.1145/3274784.3274789.

Fuhr, N. (2018). Some common mistakes in IR evaluation, and how they can be avoided. *SIGIR Forum*, *51*(3), 32–41. https://doi.org/10.1145/3190580.3190586.

Garcin, F., Faltings, B., Donatsch, O., Alazzawi, A., Bruttin, C., & Huber, A. (2014). Offline and online evaluation of news recommender systems at swissinfo.ch. In *Proceedings of the 8th ACM conference on recommender systems, RecSys '14* (pp. 169–176). ACM, New York, NY, USA. https://doi.org/10.1145/2645710.2645745.

Gilotte, A., Calauzènes, C., Nedelec, T., Abraham, A., & Dollé, S. (2018). Offline A/B testing for recommender systems. In *Proceedings of the 11th ACM international conference on web search and data mining, WSDM '18* (pp. 198–206). ACM.

Gini, C. (1912). *Variabilità e Mutuabilità*. Cuppini: Contributo allo Studio delle Distribuzioni e delle Relazioni Statistiche.

Gruson, A., Chandar, P., Charbuillet, C., McInerney, J., Hansen, S., Tardieu, D., & Carterette, B. (2019). Offline evaluation to make decisions about playlist recommendation. In *Proceedings of the 12th ACM international conference on web search and data mining, WSDM '19* (pp. 420–428). ACM.

Gunawardana, A., & Shani, G. (2015). Evaluating recommender systems. In: F. Ricci, L. Rokach, B. Shapira (eds.) *Recommender systems handbook*, 2nd edn. (pp. 265–308). Springer, Boston, MA, USA. https://doi.org/10.1007/978-1-4899-7637-6_8.

Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *Acm Transactions on Interactive Intelligent Systems*, *5*(4), 19:1–19:19. https://doi.org/10.1145/2827872.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, *22*(1), 5–53. https://doi.org/10.1145/963770.963772.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, *22*(1), 89–115. https://doi.org/10.1145/963770.963774.

Hosanagar, K., Fleder, D., Lee, D., & Buja, A. (2014). Will the global village fracture into tribes: Recommender systems and their effects on consumers. *Management Science*, *60*(4), 805–823. https://doi.org/10.2139/ssrn.1321962.

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 eighth IEEE international conference on data mining, ICDM '08* (pp. 263–272). IEEE, Washington, DC, USA. https://doi.org/10.1109/ICDM.2008.22.

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, *20*(4), 422–446. https://doi.org/10.1145/582415.582418.

Joachims, T., Swaminathan, A., & Schnabel, T. (2017). Unbiased learning-to-rank with biased feedback. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining, WSDM '17* (pp. 781–789). ACM. https://doi.org/10.1145/3018661.3018699.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, *30*(1–2), 81–93. https://doi.org/10.1093/biomet/30.1-2.81.

Kohavi, R., Longbotham, R., Sommerfield, D., & Henne, R. M. (2009). Controlled experiments on the web: Survey and practical guide. *Data Mining and Knowledge Discovery*, *18*(1), 140–181. https://doi.org/10.1007/s10618-008-0114-1.

Konstan, J.A., & Adomavicius, G. (2013). Toward identification and adoption of best practices in algorithmic recommender systems research. In *Proceedings of the international workshop on reproducibility and replication in recommender systems evaluation, RepSys '13* (pp. 23–28). ACM, New York, NY, USA. https://doi.org/10.1145/2532508.2532513.

Kutlu, M., Elsayed, T., & Lease, M. (2018). Intelligent topic selection for low-cost information retrieval evaluation: A New perspective on deep vs. shallow judging. *Information Processing & Management*, *54*(1), 37–59. https://doi.org/10.1016/j.ipm.2017.09.002.

Losada, D. E., Parapar, J., & Barreiro, A. (2017). Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems. *Information Processing & Management*, *53*(5), 1005–1025. https://doi.org/10.1016/j.ipm.2017.04.005.

Lu, X., Moffat, A., & Culpepper, J. S. (2016). The effect of pooling and evaluation depth on IR metrics. *Information Retrieval Journal*, *19*(4), 416–445. https://doi.org/10.1007/s10791-016-9282-6.

Marlin, B. M., Zemel, R. S., Roweis, S., & Slaney, M. (2007). Collaborative filtering and the missing at random assumption. In *Proceedings of the 23rd conference on uncertainty in artificial intelligence, UAI'07* (pp. 267–275). Arlington, VA: AUAI Press.

Matos-Junior, O., Ziviani, N., Botelho, F., Cristo, M., Lacerda, A., & da Silva, A. S. (2012). Using taxonomies for product recommendation. *Journal of Information and Data Management*, *3*(2), 85–100.

McNee, S. M., Riedl, J. T., & Konstan, J. A. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06 extended abstracts on human factors in computing systems, CHI EA '06* (p. 1097). ACM, New York, NY, USA. https://doi.org/10.1145/1125451.1125659.

Ning, X., & Karypis, G. (2011). SLIM: sparse linear methods for top-N recommender systems. In *Proceedings of the 2011 IEEE 11th international conference on data mining, ICDM '11* (pp. 497–506). IEEE, Washington, DC, USA. https://doi.org/10.1109/ICDM.2011.134.

Parapar, J., Bellogín, A., Castells, P., & Barreiro, Á. (2013). Relevance-based Language modelling for recommender systems. *Information Processing & Management*, *49*(4), 966–980. https://doi.org/10.1016/j.ipm.2013.03.001.

Parapar, J., Losada, D.E., PresedoQuindimil, M.A., & Barreiro, Á. (2019). Using score distributions to compare statistical significance tests for information retrieval evaluation. Journal of the Association for Information Science and Technology. https://doi.org/10.1002/asi.24203

Park, S.-T., Chu, W., Park, S.T., & Chu, W. (2009). Pairwise preference regression for cold-start recommendation. In *Proceedings of the 3rd ACM conference on recommender systems, RecSys '09* (pp. 21–28). ACM, New York, NY, USA. https://doi.org/10.1145/1639714.1639720.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence, UAI '09* (pp. 452–461). AUAI Press, Arlington, VA, US.

Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender systems handbook*, 2nd edn. Boston, MA: Springer. https://doi.org/10.1007/978-1-4899-7637-6.

Robertson, S. (2006). On GMAP: And other transformations. In *Proceedings of the 15th ACM international conference on information and knowledge management, CIKM '06* (pp. 78–83). ACM, New York, NY, USA. https://doi.org/10.1145/1183614.1183630.

Rossetti, M., Stella, F., & Zanker, M. (2016). Contrasting offline and online results when evaluating recommendation algorithms. In *Proceedings of the 10th ACM conference on recommender systems, RecSys '16* (pp. 31–34). ACM, New York, NY, USA. https://doi.org/10.1145/2959100.2959176.

Said, A., & Bellogín, A. (2014). Comparative recommender system evaluation: Benchmarking recommendation frameworks. In *Proceedings of the 8th ACM conference on recommender systems, RecSys '14* (pp. 129–136). ACM, New York, NY, USA. https://doi.org/10.1145/2645710.2645746.

Sakai, T. (2006). Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '06* (pp. 525–532). ACM, New York, NY, USA. https://doi.org/10.1145/1148170.1148261.

Sakai, T. (2012). Evaluation with informational and navigational intents. In *Proceedings of the 21st international conference on world wide web, WWW '12* (pp. 499–508). New York, NY: ACM. https://doi.org/10.1145/2187836.2187904.

Sakai, T., & Kando, N. (2008). On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval*, *11*(5), 447–470. https://doi.org/10.1007/s10791-008-9059-7.

Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., & Joachims, T. (2016). Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 33rd international conference on machine learning, ICML '16* (pp. 1670–1679).

Siroker, D., & Koomen, P. (2013). *A/B testing: The most powerful way to turn clicks into customers*. New York: Wiley.

Spärck Jones, K., & Van Rijsbergen, C.J. (1975). Report on the need for and provision of an ideal information retrieval test collection. British Library Research and Development Reports. University Computer Laboratory

Steck, H. (2010). Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '10* (pp. 713–722). New York, NY: ACM. https://doi.org/10.1145/1835804.1835895.

Swaminathan, A., Krishnamurthy, A., Agarwal, A., Dudík, M., Langford, J., Jose, D., Zitouni, I. (2017). Off-policy evaluation for slate recommendation. In *Proceedings of the 31st annual conference on neural information processing systems, NIPS '17* (pp. 3635–3645).

Takács, G., Pilászy, I., Németh, B., & Tikk, D. (2009). Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, *10*, 623–656. https://doi.org/10.1145/1577069.1577091.

Valcarce, D., Bellogín, A., Parapar, J., & Castells, P. (2018). On the robustness and discriminative power of information retrieval metrics for top-N recommendation. In *Proceedings of the 12th ACM conference on recommender systems, RecSys '18* (pp. 260–268). New York, NY: ACM. https://doi.org/10.1145/3240323.3240347.

Valcarce, D., Parapar, J., & Barreiro, Á. (2016). Efficient pseudo-relevance feedback methods for collaborative filtering recommendation. In *Proceedings of the 38th European conference on information retrieval, ECIR '16* (pp. 602–613). Berlin: Springer. https://doi.org/10.1007/978-3-319-30671-1_44.

Valcarce, D., Parapar, J., & Barreiro, Á. (2016). Item-based relevance modelling of recommendations for getting rid of long tail products. *Knowledge-Based Systems*, *103*, 41–51. https://doi.org/10.1016/j.knosys.2016.03.021.

Valcarce, D., Parapar, J., Barreiro, Á. (2016). Language models for collaborative filtering neighbourhoods. In *Proceedings of the 38th European conference on information retrieval, ECIR '16* (pp. 614–625). Berlin: Springer. https://doi.org/10.1007/978-3-319-30671-1_45.

Voorhees, E.M. (2002). The philosophy of information retrieval evaluation. In *Evaluation of cross-language information retrieval systems: Second workshop of the cross-language evaluation forum, CLEF 2001* (pp. 355–370). Berlin: Springer. https://doi.org/10.1007/3-540-45691-0_34.

Voorhees, E.M. (2005). Overview of the TREC 2004 Robust Track. In *ACM SIGIR forum*

Wang, J., de Vries, A.P., & Reinders, M. J. T. (2006). A user-item relevance model for log-based collaborative filtering. In *Proceedings of the 28th European conference on IR research, ECIR '06* (pp. 37–48). London: Springer. https://doi.org/10.1007/11735106_5.

Webber, W., Moffat, A., & Zobel, J. (2010). The effect of pooling and evaluation depth on metric stability. In *Proceedings of the 3rd international workshop on evaluating information access, EVIA '10* (pp. 7–15).

Yang, L., Cui, Y., Xuan, Y., Wang, C., Belongie, S., & Estrin, D. (2018). Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM conference on recommender systems, RecSys '18* (pp. 279–287). ACM.

Yilmaz, E., & Aslam, J. A. (2008). Estimating average precision when judgments are incomplete. *Knowledge and Information Systems*, *16*(2), 173–211. https://doi.org/10.1007/s10115-007-0101-7.

Yin, H., Cui, B., Li, J., Yao, J., & Chen, C. (2012). Challenging the long tail recommendation. *Proceedings of the VLDB Endowment*, *5*(9), 896–907. https://doi.org/10.14778/2311906.2311916.

Yu, H. T., Jatowt, A., Blanco, R., Joho, H., & Jose, J. M. (2017). An in-depth study on diversity evaluation: The importance of intrinsic diversity. *Information Processing & Management*, *53*(4), 799–813. https://doi.org/10.1016/j.ipm.2017.03.001.