Received: 11 May 2021 | Accepted: 8 July 2021

DOI: 10.1002/aaai.12051

SPECIAL TOPIC ARTICLE

Offline recommender system evaluation: Challenges and new directions

Pablo Castells[1] | **Alistair Moffat**[2]

[1]Universidad Autónoma de Madrid, Madrid, Spain

[2]The University of Melbourne, Melbourne, Australia

Correspondence
Alistair Moffat, The University of Melbourne, Melbourne, Australia.
Email: ammoffat@unimelb.edu.au

Funding information
Australian Research Council, Grant/Award Number: DP190101113; Spanish Government, Grant/Award Number: PID2019-108965GB-I00

Abstract

Offline evaluation is an essential complement to online experiments in the selection, improvement, tuning, and deployment of recommender systems. Offline methodologies for recommender system evaluation evolved from experimental practice in Machine Learning (ML) and Information Retrieval (IR). However, evaluating recommendations involves particularities that pose challenges to the assumptions upon which the ML and IR methodologies were developed. We recap and reflect on the development and current status of recommender system evaluation, providing an updated perspective. With a focus on offline evaluation, we review the adaptation of IR principles, procedures and metrics, and the implications of those techniques when applied to recommender systems. At the same time, we identify the singularities of recommendation that require different responses, or involve specific new needs. In addition, we provide an overview of important choices in the configuration of experiments that require particular care and understanding; discuss broader perspectives of evaluation such as recommendation value beyond accuracy; and survey open challenges such as experimental biases, and the cyclic dimension of recommendation.

INTRODUCTION

Recommendation technologies started to develop nearly three decades ago, and have grown to a point where they are perceived nowadays as a connatural feature in our daily online experience. We have grown accustomed to recommendations as we are shopping online, listening to music, watching series and movies, reading news, making social connections and browsing through their posts, or planning for vacation. As in any applied science domain, evaluation is central in recommendation technology development and research. After three decades of development, evaluating recommender systems remains a challenging endeavor.

Current offline evaluation methodologies evolved from experimental practice in Machine Learning (ML) and Information Retrieval (IR). However, evaluating recommender systems involves added complexities that challenge the simplifications upon which the ML and IR evaluation methodologies were developed. In particular, the ground truth for evaluating recommendations—required for meaningful experimentation—is difficult to obtain at scale in any controlled environment. This is because the source of ground truth information is people—end-users—in large numbers, who cannot be bypassed or proxied in any meaningful way, since the "truth" being sought is precisely the individual and subjective inclinations and preferences of those people.

Moreover, the scale requirement applies to the input for the evaluated systems as well. An (unsupervised) search algorithm can produce effective results for a single query, whereas even the simplest collaborative filtering algorithms need thousands of users to produce meaningful output. Data collection is thus unavoidably expensive in even the humblest recommendation experiment.

Collecting test data "in the wild" (that is, from a live operational system) introduces substantial complexities, such as fragmentary and/or biased ground truth data. Unlike other ML domains, recommendation does not aim to predict exactly what users will do (in the way that weather prediction aims to forecast what the weather will be tomorrow, or fraud detection aims to recognize untoward data access), but what users *would* do if they were to be offered a particular choice. In this respect, recommendation is more akin to medical research, where recommendation is the "treatment," and the goal is improved "health" (user satisfaction) as a consequence of the recommended treatment. From this perspective, valuable information may lie in what is best referred to as the "unobserved truth"—the choices that users were not offered, never experienced, and hence were not collected in the experimental data. These unexplored truths pose great challenges to recommendation evaluation.

A simplistic representation of the recommendation problem as a pure regression or classification task, disregarding the underpinning motivations of the system's users, or the context of the business in which recommendation is deployed, can render an evaluation approach irrelevant. Evaluating recommendation in a narrow perspective may still be useful nonetheless, and inform partial but important aspects of the effectiveness of an algorithm or a system, or a specialized component with a very specific mission as a component of a larger system. But even in a simplified representation, recommendation has peculiarities of its own, that do not arise in other ML and IR problems, and that need to be reflected in the task representation and the experimental setup.

In this article, we explore offline evaluation of recommender systems, with an emphasis on the techniques and methodologies that might be employed by academic researchers making use of static resources, or practitioners selecting, training, and optimizing models for subsequent online testing. Our purpose is to examine the many design choices required when planning such an experiment, and, at the same time, highlight areas in recommendation measurement that remain vexed, and where innovative solutions continue to be sought.

In particular, after briefly reprising the recommendation task and summarizing the differences between offline and online evaluation, we provide a status report describing the current practice in offline recommendation evaluation. We explore recommendation evaluation through a lens derived from IR evaluation, drawing on more than five decades of work, progress, knowledge, and lessons learned. We then present subtle details of experimental setups, with the goal of guiding other researchers past the many possible pitfalls. After this, we discuss updated perspectives in recommendation evaluation in regard to discovery, bias, exploration, and the interactive recommendation cycle.

FIGURE 1 The rating matrix

THE RECOMMENDATION TASK

In a generic definition of the recommendation task, users are observed interacting with (rating, clicking, playing, purchasing) products and choices in a particular system, and the problem consists of predicting which choices users might enjoy next. User–item interactions can be viewed as a user–item matrix (Figure 1), where interaction data are associated to the corresponding matrix cell (Adomavicius and Tuzhilin 2005). As a useful simplification, the data can be abstracted to a scalar (or binary) value reflecting the degree of enjoyment or utility that a user draws from an item. In this representation, a recommender system should predict values for the unobserved matrix cells.

In early work in the field, the cell values output by recommender systems were intended to literally predict user actions or accurately match rating values (Herlocker et al. 1999). From a practical perspective, the specific values are not of any consequence, as long as they are not displayed to users. Only the item selection and order they induce is important. In typical recommender system applications, the item scores determine where to place the recommended items in the user interface. From this point of view, the recommendation task can be cast as a ranking problem (where we use "ranking" as a simplification to mean selecting and arranging items in some display order)

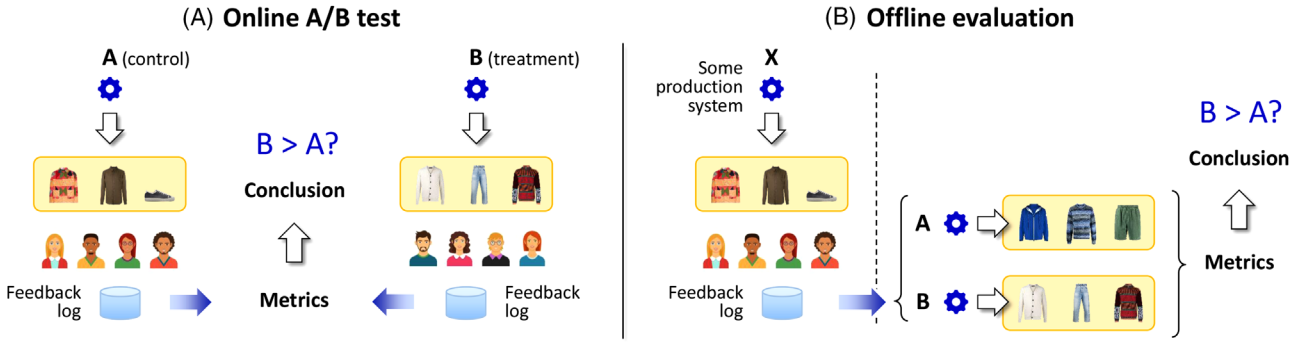

FIGURE 2 Online versus offline evaluation

and—for most purposes—an IR task (Bellogín, Castells, and Cantador 2017; Herlocker et al. 2004).

The list of algorithmic solutions to the recommendation task is endless and keeps growing—a review would far exceed the scope of the present paper. The point to be noted is that suitable evaluation methodologies are critically important to such algorithmic developments, because they provide the basis on which researchers can differentiate amongst the many proposed approaches, and/or find optimal configurations for them.

OFFLINE VERSUS ONLINE EVALUATION

Algorithm selection and updates in recommender system applications are generally informed by online evaluation, typically consisting of A/B tests (Amatriain and Basilico 2015; Gomez-Uribe and Hunt 2015). In an A/B test (Figure 2A), the system currently in production ("control") is compared to one or more updated variants ("treatments"), by diverting a fraction of live user traffic to the latter and comparing their effectiveness in terms of business metrics (Jannach and Jugovac 2019) commonly related to user engagement (click-through rate, time watching, etc.) and sales (order size, revenue, profit, etc.).

A/B tests provide the most direct assessment of the impact of recommendation in the business performance, but require time, have limited bandwidth, and involve risk as they directly expose system changes to customers. For such reasons, new ideas are tuned and filtered through extensive offline experiments, before bringing them to the final A/B test (Amatriain and Basilico 2015; Gomez-Uribe and Hunt 2015). Academic research, as the far end in the innovation to production funnel, rarely has access to a production system and therefore commonly relies on offline experimentation entirely.

Offline evaluation consists in collecting user interaction data—most commonly from a working system—over a period of time and setting it aside for repeated experimentation (Figure 2B). The offline data are usually divided into two disjoint subsets: the "training" subset is passed as input to the evaluated systems, and the "test" subset is taken as ground truth for metric computation.

Offline experimentation aims to be predictive of online performance, yet the correlation between offline and online evaluation outcomes is often weak (Amatriain and Basilico 2015; Garcin et al. 2014; Gomez-Uribe and Hunt 2015; Jannach and Jugovac 2019). What is more, the outcomes of different offline experiments (on the same systems and data) do not correlate well with each other all too often (Cañamares, Castells, and Moffat 2020). Several causes can be pointed out at the root of this divergence:

- Lack of shared, sufficiently detailed protocols, and shared tools for offline experimentation.
- A sometimes partial understanding of the subtleties and effects of detailed experimental settings.
- Considerable, intrinsic hidden complexity involved in offline recommender system evaluation.

The above challenges are partly a consequence of different fields and views confluencing in the recommendation task, such as ML, IR, and Human–Computer Interaction. The idea of producing personalized recommendations initially arose as a regression/classification problem, and solutions were, therefore, evaluated with corresponding protocols and metrics, focusing on prediction error (Herlocker et al. 1999). As recommender systems grew into an established industry, the view shifted towards an IR perspective (Cremonesi, Koren, and Turrin 2010; Herlocker et al. 2004), more in accordance with real applications. The adoption of IR evaluation methodologies took time to overcome the rating prediction view, ingrained for years of previous research in the field. Evaluation methodology developed in the IR field over decades of community effort towards sound and standardized practice and principles. Its adoption in a new area such as recommendation is, therefore, not necessarily trivial and deserves dedicated study. This motivates an overview of IR evaluation, which we provide next.

OFFLINE EVALUATION IN INFORMATION RETRIEVAL

The field of IR is closely related to Recommender Systems, but also differs in several important ways. This section provides an overview of offline evaluation in IR. Sanderson (2010) provides details of several of the areas covered below. In this section, a *document* is a text object stored in a *retrieval system*, and is one member of a *collection* of such items; a *topic* is a specification of an *information need* as might be regarded as being typical of the users of that retrieval system; a *query* is one user's crystallization of the topic into a short list of *terms* (often, but not always, as a *bag-of-words* statement); and a *relevance judgment* (sometimes also called a *qrel*) is a human-determined assessment of the degree to which (if at all) a particular document in the collection is responsive to (that is, helps address) the information need expressed via the corresponding topic.

Key questions

Critical issues associated with the evaluation of retrieval *quality* for ordered document rankings include: determining which subset of the documents should most usefully be judged for each topic, assuming that only a limited judgment budget is available; deciding how ordered document rankings (each of which is referred to as a *run*) should be numerically scored, and what principles those scores should be based on; and dealing with the possible uncertainty in run scores arising from the likely absence of complete judgments.

Document collections and pooling

In the early 1990s, the US National Institute of Standards and Technology embarked upon a quite remarkable project to provide infrastructure in support of IR experimentation. The ensuing *T*ext *RE*trieval *C*onference (see Harman (1992) and the many subsequent volumes of TREC conference proceeding) initiated a generation of research into "at scale" retrieval systems, with even the first published TREC document collection approaching a gigabyte in size, a rather large amount of storage at the time, and something like 10 times bigger than previously available collections. Other "shared community" efforts followed, including the CLEF initiative in Europe, the NTCIR collections in Japan, and the FIRE project in India.

To build sets of relevance judgments, TREC adopted a depth-*d* *pooling* strategy (Figure 3). The runs submitted by each participating research group were truncated at depth d, with typically the top $d = 100$ documents or top $d = 200$ documents extracted from submitted runs of length 1000 or more, and then merged to get a list (for each topic) of documents that appeared within any group's top-d. Those documents were then shuffled, and presented to human annotators for labeling, with the assumption being that the majority of any documents that were indeed relevant would be identified, and that any unjudged documents could be safely assumed to be nonrelevant (Harman 2005). That is, compared to recommender evaluation, the ground-truth data used in IR evaluations are typically generated post hoc, and in volumes determined by an experimental budget rather than by user behavior.

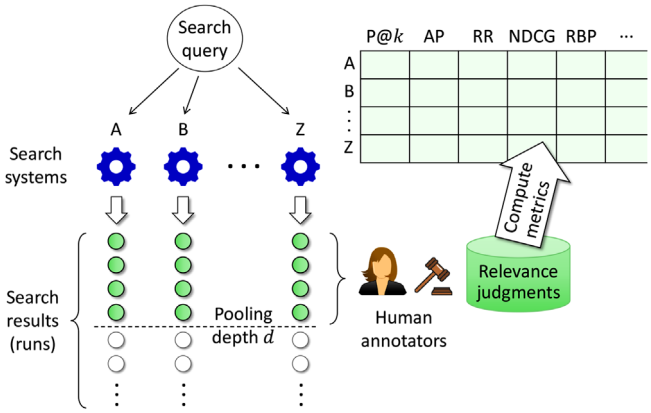

FIGURE 3 Relevance judgment pooling in IR evaluation

A range of investigations into the robustness of pooling have been carried out, concluding that while even deep pooling is unlikely to find all of the relevant documents, the system comparisons that emerge from the partial judgments are sufficient to allow systems to be, by and large, reliably compared (Zobel 1998). Subsequent work has investigated the resilience of evaluations based on pooling and incomplete judgments (for example, Buckley and Voorhees 2000; Buckley and Voorhees 2004; Buckley et al. 2007; Büttcher et al. 2007; Sakai and Kando 2008; Sanderson and Zobel 2005); including the issue of selecting documents into the pool itself, and how to merge the runs in more nuanced or sensitive ways (Buckley et al. 2007; Lipani et al. 2021; Moffat, Webber, and Zobel 2007) so as to improve the usefulness of the eventual measurement and systems comparisons taking place.

An alternative response to the imprecision introduced by incomplete judgments has been to quantify the maximum extent of measurement uncertainty via a *residual* (Moffat and Zobel 2008). If the residual of a effectiveness measurement derived from a run is low, it indicates that the judgment set provided good coverage, and that the score is reliable. If the residual is high, there is the possibility of score imprecision, and hence a need for caution in interpreting the results that were obtained.

Ranked list evaluation

The first metrics applied to ranked lists were variants of two traditional set retrieval measurements: precision@k is the fraction of the first k items in the run that are relevant; and recall@k is the fraction of the available relevant documents that appear in the first k items in the run. While natural extensions of precision and recall, these metrics do not account for the fundamental nature of runs, namely that they represent the system's preference ordering over the collection's documents, an issue that also arises in recommender systems evaluation. For example, precision@10 suffers the same degradation in score if relevant documents at rank 1 and at rank 10 are swapped for nonrelevant ones. This is a correct outcome if users are assumed to pay equal attention to each of the top-10 documents (and to not ever look at any documents outside that top-10 set), but does not sit well with the typical behavior of search system users, who tend to process the ranking from top down.

It was thus natural for top-weighted effectiveness metrics to emerge. One important metric that has been closely connected with TREC activities through many years is called *average precision*, or AP, which was codified into the program `trec_eval`. If a topic has R relevant documents, and a system places those relevant documents (in any permutation) into positions $\{p_1, p_2, \ldots, p_R\}$ within the run, then AP is calculated as $\text{AP} = (1/R) \sum_{i=1}^{R} (i/p_i)$, that is, as the average of R separate precision@p_i scores, one for each place at which a relevant document appears.

Average precision is sometimes referred to as a *systems metric* because its value is typically affected by document relevance down to relatively deep positions in the run (indeed, down to depth p_R). At the other end of the spectrum, *reciprocal rank*, RR, is a *user metric*, more likely to reflect the perceptions of a typical shallow-examination user: $\text{RR} = (1/p_1)$. Reciprocal rank ignores the positioning of all relevant documents after the first, modeling users as being fully satisfied when they have found a single answer.

Both AP and RR assume *binary* topic-document query relevance labels drawn from $\{0, 1\}$. A wide range of other scoring formulae have been proposed, including ones that make use of *graded* relevance labels, where each topic-document combination is assigned a fractional *gain* value between zero and one inclusive. Those gain values are then accumulated down the system's ranking, but are also increasingly discounted as a way of ensuring that the metric is more heavily weighted to the top of the run. For example, Järvelin and Kekäläinen (2002) propose *discounted cumulative gain*, computed as $\text{DCG@}k = \sum_{i=1}^{k} (r_i / \log_2(1+i))$, where $0 \le r_i \le 1$ is the gain value associated with the document in position i of the run. Note that DCG is unbounded as k increases, and values greater than one can emerge. As one way of resolving that slight awkwardness, Järvelin and Kekäläinen also proposed a *normalized* version, referred to as NDCG@k, in which the DCG@k score is divided by the "ideal DCG@k" from a run that contains every document in the collection sorted by decreasing gain value r_i. Now the metric values are bounded above by one, with k still giving a sense of the maximum depth to which the user will examine the run, or of the length of the run. The latter is an important concern in recommender systems evaluation.

A different approach to normalization was introduced by Moffat and Zobel (2008). Their *rank-biased precision* (RBP) metric uses a geometric decay function that has a bounded sum, so that the evaluation can be taken to an arbitrary depth. In RBP a parameter ϕ describes the user's persistence when scanning the ranking, and can be tailored to the usage scenario that is anticipated. For example, a user with $\phi = 0.5$ is regarded as being relatively impatient, and has (only) a 50% chance of viewing the $i+1$th document after they have viewed the ith. Their expected viewing depth in the ranking is thus two documents; contrast that with a $\phi = 0.95$ user, who is anticipated as having a average search depth of 20. As foreshadowed above, RBP also allows the computation of a residual, bounding the maximum possible score change that could arise if all unjudged documents were in fact fully relevant (Moffat and Zobel 2008).

Goal/Task sensitivity

Many other metrics have also been proposed, and the ones described in the previous section are but a sample of the most widely used options. As one example, there has been effort put into metrics that attempt to infer relevance labels in the cases where judgments are missing (Buckley and Voorhees 2004; Sakai 2007), rather than simply assume such documents to be nonrelevant. As yet, these approaches have yet to be considered by the recommender community.

Another thread of development has been the emergence of more sophisticated user browsing models. Chapelle et al. (2009) propose *expected reciprocal rank* (ERR) in which users are assumed to be seeking a single relevant document, with each r_i value indicative of the likelihood of users regarding the ith document as being "the one." More generally, Moffat et al. (2017) suggest that three factors should correlate positively with the decision made by the user as to whether or not to continue to the document at depth $i+1$: (i) the current rank i in the ranking; (ii) the amount of total gain desired by the user when they commenced their search, denoted by T; and (iii) the amount

of that gain still unfound by depth i. These desiderata define a family of metrics that are *goal sensitive* and also *adaptive*, and specified in terms of a function $C(i)$, the conditional continuation probability of proceeding from rank i to rank $i + 1$. Moffat et al. (2017) then use their "C/W/L framework" to define a metric INST that has the proposed properties. The adaptive property is one that makes good sense—as users have success in finding in full or even in part what they are looking for, they can be expected to be less likely to continue looking.

Maxwell et al. (2015) also consider what it is that makes users stop examining a ranking; and Azzopardi, Thomas, and Craswell (2018) provide a further basis on which stopping might occur, by considering a localized rate at which relevance is being accrued. Other related work is by Zhang et al. (2017) and Luo et al. (2017), who describe different possible balances between gain and decay; and from Smucker and Clarke (2012), who add document length and document repetition into a time-based measurement regime, arguing that gain should be measured relative to the time spent accumulating it. Many of these ideas are yet to be employed in offline evaluation of recommender systems, and offer directions that might be productive if the question of ground truth data can be addressed.

Finally in this section, note that offline evaluation is only one way in which IR systems can be compared, and that A/B testing is very important in commercial search scenarios. User studies—via a wide range of techniques—can also be very informative (Kelly 2009), with human factors being at least as important in terms of overall "user satisfaction with a search service" as is retrieval quality when measured by an effectiveness metric of the type discussed here. Human factors similarly play an important role in user-focused recommender evaluation.

CHALLENGES IN OFFLINE RECOMMENDATION EXPERIMENT DESIGN

As discussed earlier, recommender system evaluation originally developed as a classification or regression task (Herlocker et al. 1999; Shardanand and Maes 1995). The particularities of recommendation soon made the complexity of evaluation apparent though, motivating specific analysis and research efforts (Cañamares, Castells, and Moffat 2020; Ferrari Dacrema et al. 2021; Herlocker et al. 2004; Sun et al. 2020). Seen as a ML task, recommendation is peculiar in (a) the key importance of ranking (the selection and placement of recommended items in the user display), along with the fact that (b) the system's goal is to predict human satisfaction and/or actions, whereby output "correctness" becomes an elusive notion, in contrast to ML tasks such as image recognition or medical diagnosis where ground truth has a more objective basis.

These particularities are proper of IR problems. Yet compared to a search task, recommendation has singularities of its own such as the absence (or indirect role at best) of an explicit user query, and the difficulty of eliciting relevance judgments without intervention of the end-users to whom the recommendations are to be delivered (Bellogín, Castells, and Cantador 2017; Lu et al. 2021). These particularities bring additional complexity to experimental design. As a consequence, small details in the configuration of experiments can result in substantial differences and contradictions in the outcomes (Cañamares and Castells 2020; Ferrari Dacrema et al. 2021). This calls for an improved awareness and understanding of such fine details by experimenters, and a detailed account of experiment configurations when communicating evaluation results (Cañamares, Castells, and Moffat 2020). We discuss next some of the most important aspects in this scope.

Collecting and splitting the data

The data for offline recommender system experimentation are most typically collected from a working system, for example, by a dump of a certain period of time's worth of logged user interaction records in the system (Bertin-Mahieux et al. 2011; Harper and Konstan 2016). When specific data characteristics are sought, data can also be sampled through surveyed (as opposed to spontaneous) user feedback (Cañamares and Castells 2018; Marlin et al. 2007; Schnabel et al. 2016).

As in any supervised ML task, the data are split into training and test subsets. The former is given as input to the evaluated systems, and the latter is used to compute offline evaluation metrics on the system outputs. In some datasets, the training and test data are collected separately, with different sampling protocols (Marlin et al. 2007; Schnabel et al. 2016). In most cases though, a single set is supplied to the experimenter, who is responsible for appropriately partitioning the data (Gunawardana and Shani 2015).

The way the data are split can make a difference in the experiment outcomes, not just in the metric values (Bellogín, Castells, and Cantador 2017), but also in the qualitative system comparisons (Cañamares, Castells, and Moffat 2020; Meng et al. 2020). Data partitioning options include random versus temporal sampling, global versus per-user sampling, the choice of a split ratio, and more (Cañamares, Castells, and Moffat 2020; Said and Bellogín 2014). The right approach may depend on what is feasible to begin with (considering the data density, the availability of meaningful timestamps, etc.), and the

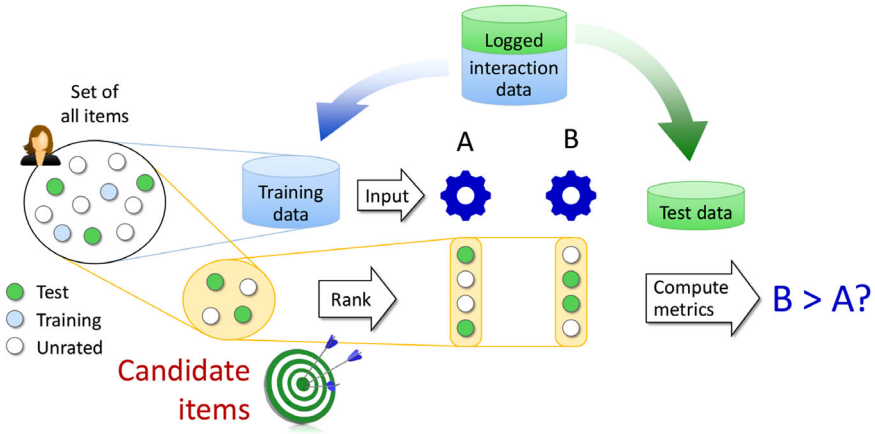

FIGURE 4 Data split and candidate set in offline evaluation

specific goal of the evaluation. As a general rule, the preferred approach should maximize the similarity to the conditions where the system is to be deployed—or seek maximum generality and minimum assumptions when no specific deployment is targeted. At any rate, detailed transparency about the experiment configuration should be the best precept to make evaluation results most useful and meaningful (Cañamares, Castells, and Moffat 2020).

Candidate item set subsampling

Given a data partition, the evaluated systems should output a ranking of recommended items for every user. A question the experimenter faces at this point is what items should the system be requested to rank for each user (Figure 4). While a first naive answer might be "rank all items in the dataset," one can find reasons to consider smaller sets. For instance, in most cases, we may not want an evaluated system to recommend the user choices that the system was given as training data—the same as we would not ask a classifier to classify the training examples. Koren (2008) was first to bring this idea further by restricting the system's output to an arbitrary subset of candidate item. The idea was carried on by many other researchers (see e.g., the experiments reviewed by Ferrari Dacrema et al. 2021). The motivation for this design option was not always explicit, but might be related to a purpose of conceptual simplicity, and potential savings in computational cost for some algorithms.

The selection of target items has a direct impact on metric values (see e.g. Bellogín, Castells, and Cantador 2017), and can even flip the comparison between systems, as proved by Krichene and Rendle (2020) and Cañamares and Castells (2020), and observed earlier by Steck (2013) and Cañamares, Castells, and Moffat (2020). Cañamares and Castells (2020) provided some insights to these discrepancies, and found further reason for selecting a larger or smaller number of candidate items: maximizing the statistical power of experiments, and minimizing the evaluation bias. At a minimum, all items with a test rating should be arguably included in the candidate set. Experiment power and fidelity were found to be highest at an intermediate point, where a certain amount of unrated candidate items are included, but not all. A rule of thumb is hinted: this ideal point may be determined as the candidate set size with which the experiment produces the fewest ties between the compared systems.

A potential drawback of reduced candidate sets is a loss in similarity to the problem that a real system needs to solve. But this is not exactly the case: recommender systems in industry typically work as a chain of algorithms that progressively narrow down the set of items to be recommended (Amatriain and Basilico 2015; Gomez-Uribe and Hunt 2015). Thus, the ideal candidate item set would be the one that is most alike to what the evaluated algorithm will handle at its specific point in the recommendation pipeline. An algorithm may be very effective at filtering large candidate sets early in the chain, while another may be much better as a late-stage ranker of small lists.

An ideal dataset would log what the candidate items were when the data were collected, so that this can be used as the target set in subsequent offline experiments. Knowing which among those items were actually *impressed* within the user's sight would enable additional offline evaluation possibilities. Publicly available datasets including information of this kind would be certainly welcome by the community (Pérez Maurera et al. 2020).

Computing offline metrics

By equating users to queries, items to documents, and test data to relevance judgments (Bellogín, Castells, and Cantador 2017), any IR metric (such as the ones discussed

earlier in the Information Retrieval section) can be used to evaluate recommender systems. Precision, recall, MRR or NDCG are indeed used routinely in recommender system evaluation (Gunawardana and Shani 2015; Valcarce et al. 2020). False-positive metrics (measuring unpleasing recommendations) are less frequent in the literature but seem important in industry, and might deserve wider consideration as a complementary perspective (Mena-Maldonado et al. 2021).

When analyzing evaluation results, statistical power can be considered as important in recommendation as in IR evaluation at large. As a noteworthy difference, the number of data points (number of users) is typically much larger in recommendation datasets than it is in public search evaluation benchmarks (number of queries)—as a consequence, statistical significance is typically more easily achieved and less often an issue in the recommendation literature. To this respect, Cañamares and Castells (2020) suggest measuring the number of ties between systems as a complementary measure of discriminative power that can surface nuances not captured by statistical tests.

Typical metric values in offline recommendation experiments are very often orders of magnitude smaller than in search evaluation. This raised doubts in the early days when IR methodologies started to be adopted, but is now better understood as just a consequence of a much higher sparsity of test data in recommendation compared to pooled relevance judgments for search evaluation (Bellogín, Castells, and Cantador 2017), combined with the lack of explicit direction in the user's need, in contrast to retrieval tasks involving an explicit need description (a query). Low metric values therefore do not mean that systems are ineffective, but that most of the data to measure their effectiveness is missing. Statistical significance tests usually confirm that such low metric values are meaningful in comparing systems, as much as they can be in search experiments with TREC data, even if the values are not meaningful as an absolute measure of individual system performance.

Incomplete rankings

When data sparsity heavily affects the available training data for some user, some algorithms may find it difficult to produce a reliable recommendation, or to include as many items in the list as the evaluation metric expects: if we wish to measure, for instance, precision@k, the algorithm may return less than k items (or none at all) for some users. This problem has been barely discussed in the literature, and may come unnoticed to the unwary experimenter, inadvertently distorting the experiment results (Cañamares and Castells 2020; Cañamares, Castells, and Moffat 2020). Incomplete rankings make the metric technically undefined, and require a nonobvious decision as to how the recommendation shortage should be reflected in the metric scores. Choices one may consider to cope with this situation include:

1. Penalizing the algorithm for not filling the required rank positions, counting them as equivalent to nonrelevant recommendations.
2. Forgiving the algorithm, lifting the metric cutoff to the number of items the algorithm was able to rank.
3. Filling in the missing positions with some fallback algorithm, such as random items, popular items, or another recommender system.

Option 1 can be harsh, as it is sometimes wise to abstain from making recommendations that might be more harmful than beneficial—we might want to recognize the algorithm's ability to "quit while ahead" (Liu et al. 2016). However, option 2 can be unfair to algorithms that make their best to recommend as many items as they are requested, as opposed to others that refuse to rank all but the easiest items. Option 3 can make especial sense when the fallback algorithm is some appropriate baseline, or a system that the evaluated algorithm competes against.

In general, none of these options is necessarily better than the others; the best option is the one that better matches the specific experiment purpose. Whatever the choice is, we suggest reporting recommendation coverage@k (the rate of filled rank positions for a cutoff k across users) as a complementary measure, to put the main metric in perspective and better understand the potential fluctuations due to coverage issues (Cañamares and Castells 2020; Cañamares, Castells, and Moffat 2020).

Beyond relevance

While relevance is a basic condition for recommendation to be useful, matching the user's tastes may not be enough to provide valuable suggestions. For instance, recommendation often comes along with a purpose of discovery. Recommending well-known user favorites, no matter how relevant, may then be rather pointless. Relevant but less obvious suggestions that users may not have even thought of searching for is likely to be far more useful (Castells, Hurley, and Vargas 2015). Recommending less widely known items is also important to overcome cold-start stages and surface the potential value of new and underexposed choices. With many e-commerce and online services evolving into online marketplaces, avoiding overconcentration around a small set of choices becomes

also a requirement for effective and fair recommendation (Abdollahpouri et al. 2020; Mehrotra et al. 2018).

Specific metrics have been thus developed to measure novelty and diversity in different angles, and have become common in offline evaluation. These include, for instance, the average pairwise dissimilarity between recommended items (Ziegler et al. 2005), the average pairwise dissimilarity between recommendations and previously consumed choices (Vargas and Castells 2011), the "unpopularity" (scarcity of past interaction) of recommended items (Zhou et al. 2010), or the Gini index of recommendations over items (Chaney, Stewart, and Engelhardt 2018; Fleder and Hosanagar 2009). The reader is referred to Castells, Hurley, and Vargas (2015) for a comprehensive survey.

BIAS AND LOOPS IN EVALUATION

Closely related to novelty and diversity, bias is one of the major challenges of offline evaluation. In particular, offline data are subject to a strong selection bias, as interaction is much more likely to be observed for some user–item pairs than others, regardless of how much the user likes each item (Marlin and Zemel 2009). We may broadly consider two components in the formation of such biases: item exposure and user selection.

Self-selection bias

When presented with an item, users are more prone to engage with some items than others (Marlin et al. 2007). Reasons include the perception by the user that one item will be more interesting to them than others, or will suit their needs better, or the item simply draws the user's attention or curiosity (e.g., a shocking video).

Exposure bias

Users are more likely to discover some items than others (Cañamares and Castells 2018). These differences are introduced by both internal biases created by the system, and external biases. External bias factors include, for instance, advertisement, fashion, mouth-to-mouth communication, or virality in social media, that boost the popularity of particular products, news, artists, brands, and so forth, outside the system. These biases may leak into offline data when users actively seek such popular items in the data logging system, or are influenced by their external environment in their choices in the system. Internal exposure biases are generally stronger than external ones, and are produced by the algorithms (such as search, browsing and recommendation functionalities) that decide which items are presented when collecting offline data. The placement of retrieved items in the user interface is an additional major internal source of bias: items at prominent positions receive a disproportionately higher user attention. Of course, time is another major factor in item exposure: the longer an item exists, the more opportunity it gets to attract people's attention—in this respect, item cold start can be seen as a natural case of observation bias.

From the items point of view, the sampling bias is often referred to as the *popularity* bias (Bellogín, Castells, and Cantador 2017; Cañamares and Castells 2018; Jannach et al. 2015; Steck 2011). Bias is typically more visible when aggregated over users, but bias can also be user-specific: items can be more popular over (or more exposed to) some groups of users than others.

The effects of bias in recommendation

Sampling bias can distort offline evaluation considerably: when the test data are biased, systems are rewarded for learning and reproducing the bias in the data, besides just pleasing users. Without any intervention, popularity thus gets amplified by recommendation; for internal biases, new systems may find resistance to change, as offline experiments will evaluate—along with recommendation relevance—how similar the evaluated algorithms are to the system with which test data were collected. In an application context, established system decisions may tend to perpetuate themselves, as models are selected and trained on the data that the current system collects: system variants that agree with the hypotheses that the deployed models build upon have higher chances to be successful in offline experiments. From a business perspective, self-reinforced biases result in missing opportunities, by failing to surface underexposed but potentially valuable items.

Broader reinforcement loops might affect the field as a whole: algorithmic research is evaluated with data collected from applications that draw from algorithmic research. Researchers have found indeed how strongly state-of-the-art collaborative filtering algorithms are biased towards recommending majority choices (Cañamares and Castells 2017; Jannach et al. 2015). However, whether this challenges the status quo as to what the best algorithms really are is an open question (Cañamares and Castells 2018)—we briefly touch on this in the next subsection.

Coping with bias

The realization of the strong biases in evaluation came along with efforts to mitigate them (Jannach et al. 2015; Marlin and Zemel 2009; Steck 2010; 2011), in order to achieve more reliable, undistorted measurements of the relevance of recommendations, better matching online

performance. Simple approaches can be conceived as test data sampling and subsampling procedures. For instance, test data can be segmented into head, torso, and tail items—or any number of strata—according to their amount of test data (Bellogín, Castells, and Cantador 2017; Cremonesi, Koren, and Turrin 2010). Popularity bias is then equalized across the compared algorithms, reducing the advantage of popularity-biased algorithms. Another simple approach in this line is to sample an equal amount of test data for all items (Bellogín, Castells, and Cantador 2017). These ideas can be effective but come with their own issues and limitations (Castells and Cañamares 2018), and tend to further increase the gap between the offline and the online settings, by the introduction of additional data manipulation steps.

More principled solutions have been developed by defining metrics that correct for the bias (Steck 2010; 2011). In this line, so-called *counterfactual* evaluation consists in modeling the bias, and correcting the metrics accordingly (Gilotte et al. 2018; Gruson et al. 2019; Jadidinejad, Macdonald, and Ounis 2021; Swaminathan et al. 2017; Yang et al. 2018). A widely considered method in this scope is *inverse propensity scoring* (IPS), which divides the value (e.g., relevance) procured by a recommended item to a user by the probability (propensity) that an interaction between this user and this item is present in the test data. The IPS correction guarantees an unbiased metric estimate in expectation.

One important challenge in IPS is estimating propensity, which can be as much of a challenge as predicting user tastes in the first place. This is feasible, however when we have some information about the test data logging policy (such as a controlled retrieval environment): propensity can be modeled knowing, for instance, the number of times an item has been presented, and the probability that users actually noticed the item according to a position bias model. IPS is known to suffer from other issues such as high variance on underexposed items, and specific elaborations have been devised to this avail (Gilotte et al. 2018). Counterfactual evaluation is still an open area and IPS may not always work as expected in all cases (Gruson et al. 2019). It is currently actively researched as a promising direction in dealing with bias in offline evaluation and feedback loops in model training.

An alternative to bias neutralization is to avoid the bias altogether when collecting test data, by sampling the data uniformly at random, thereby enabling unbiased metric estimates. Example datasets of this kind include Yahoo! R3 (Marlin et al. 2007), Coat (Schnabel et al. 2016), and CM100k (Cañamares and Castells 2018). Collecting random data is, however, expensive as it requires explicit effort and time from users, and is not trivial to scale as a sustained solution. Unbiased data can nonetheless be a useful element in developing improved debiasing methods on top of it (Wang et al. 2021), and is certainly a valuable resource for research.

On the other hand, recent studies show that evaluation with biased data may still agree (in terms of system comparisons) with unbiased evaluation, when the sampling biases agree with relevance distributions in appropriate ways (Cañamares and Castells 2018; Mena-Maldonado et al. 2021). This is not to say that biases are not a problem, but these studies indicate that many experiments can still be informative even if they are subject to bias. This notwithstanding, bias can introduce imprecision in measurements (Bellogín, Castells, and Cantador 2017), impoverished recommendations (Chaney, Stewart, and Engelhardt 2018; Fleder and Hosanagar 2009), and uncertainty as to what the degree of potential distortion in evaluation really is.

Evaluating recommendation cycles

An additional perspective in mitigating bias from feedback loops has gained traction in recent years, consisting in incorporating the cyclic nature of recommendation in the task definition. In this view, the goal of recommendation shifts from seeking local optima to procuring sustained value over a continued relationship with users. When longer-term optima are sought, one realizes that the effectiveness of recommendation at a given time step builds on the data collected from users' reactions to the previous recommendations. The recommendation problem at each step becomes fundamentally twofold: (a) pleasing the user now, and (b) improving the system knowledge about user interests, for future use. That is: making the most of the evidence of user preferences collected so far (*exploitation*), and optimizing data acquisition to maximize the value of future recommendations (*exploration*).

This realization casts recommendation as a reinforcement learning problem (Sutton and Barto 2018). In this area, multiarmed bandits (MAB) have become a popular problem representation, for which specific algorithmic solutions have been explored in the last few years (Li et al. 2010; Wang et al. 2019). Under this perspective, common recommendation algorithms are referred to as "greedy" or "myopic:" by making deterministic decisions based on incomplete user preference observations, greedy algorithms are, in statistical terms, mistaking the sample for the population. MAB, in contrast, explicitly acknowledge the uncertainty involved in the collected evidence and make stochastic recommendations, handling observations as samples from unknown distributions. In addition to long-term relevance improvements, MAB approaches enhance recommendation novelty and diversity, as a

collateral effect of their exploratory side (Sanz-Cruzado, Castells, and López 2019).

This task redefinition raises new challenges for offline evaluation. Reinforcement learning algorithms engage in cyclic interaction with users, which does not fit easily in a purely offline setting. We briefly discuss the main approaches reported to this date.

Full user simulation.
User interaction with recommended items can be modeled as a parameterized probabilistic model of user behavior (browsing, clicking, etc., see e.g., Rohde et al. 2018). The model can then be used to simulate user reactions in response to the evaluated algorithms' output. The algorithms feed and are evaluated on the data produced by such synthetic users, and cyclic recommendation experiments can be thus run. The obvious limitation of this approach is that the user model is artificial and does not necessarily represent the behavior of real users. Full user simulation can be useful, however, to analyze and compare general properties of the evaluated algorithms, such as learning convergence, parameter sensitivity, computational cost, and so forth.

Looping through offline datasets.
Some authors have simulated cyclic recommendation by splitting an offline dataset into initial training and test subsets, and then running the evaluated system repeatedly, adding to the input training subset every test data record that the system "discovers" by recommendation (Huang et al. 2020; Kawale et al. 2015; Sanz-Cruzado, Castells, and López 2019). Real test data are thus used to simulate what the users' response might have been in a live situation. The step is repeated until all test data are discovered, or after a certain number of steps. Metrics of interest, such as the cumulative recall (ratio of discovered positive test user preferences) can be monitored during the iteration.

The main problem with this approach is that experiments get biased by the sampling policy (e.g., a working system) that was used in collecting the data—the same issue we discussed earlier. To this respect, recent work has explored debiasing techniques on top of this procedure to improve this aspect (Huang et al. 2020).

Replay
The so-called "replay" approach was proposed by Li et al. (2011) to avoid the bias from offline data. The procedure consists in, first, collecting a large amount of online user feedback to randomly presented items. In the feedback collection procedure, the items are sampled, one at a time, from a small pool of options, that changes over time. In the original paper, the items were news, and the pools were hourly refreshed sets of headline stories. The resulting dataset thus consists of a set of triplets ⟨ pool/sampled item/user feedback ⟩, where the feedback is binary (click/no click).

Systems (e.g., bandits) are evaluated using this data by iterating over the triplets, and requesting the system to recommend one of the items in the corresponding pool. If the recommended item is the same as the one that was randomly sampled, the user feedback (positive or negative) is revealed to the evaluated system—otherwise, the triplet is ignored. This is not very data-efficient, as most data records get discarded, but it can be shown that this procedure enables an unbiased offline estimate of online performance without running an online test (Li et al. 2011).

CONCLUSIONS

We have summarized the current situation with regard to recommender system evaluation, and described a number of aspects of experimentation to which researchers must pay careful attention if they are to avoid the possible pitfalls. Improved replicability and reproducibility of experimental outcomes is a worthwhile goal in all areas of computing, and recommender systems research is no exception to that observation. Despite the many and varied concerns that we have noted, our overall message is one of hope, rather than despair. Forewarned is forearmed, and we believe that if researchers do indeed pay attention to these risk areas, they will have more confidence in the robustness and resilience of their experimental outcomes, and will thus be more sure-footed when making claims about new and improved algorithmic approaches.

ACKNOWLEDGMENTS
This work was partially supported by the Spanish Government (project PID2019-108965GB-I00) and by the Australian Research Council (project DP190101113).

CONFLICT OF INTEREST
The authors declare that there is no conflict.

ORCID
Pablo Castells https://orcid.org/0000-0003-0668-6317
Alistair Moffat https://orcid.org/0000-0002-6638-0232

REFERENCES
Abdollahpouri, H., G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, and L. A. Pizzato. 2020. "Multistakeholder Recommendation: Survey and Research Directions." *User Modeling and User-Adapted Interaction* 30(1): 127–58.

Adomavicius, G., and A. Tuzhilin. 2005. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art

and Possible Extensions." *IEEE Transactions on Knowledge and Data Engineering* 17(6): 734–49.

Amatriain, X., and J. Basilico. 2015. "Recommender Systems in Industry: A Netflix Case Study." In *Recommender Systems Handbook*, edited by F. Ricci, L. Rokach, and B. Shapira, 385–419. Boston, MA: Springer.

Azzopardi, L., P. Thomas, and N. Craswell. 2018. "Measuring the Utility of Search Engine Result Pages: An Information Foraging Measure." In *Proceedings of the SIGIR*, 605–14. New York, NY, USA: ACM.

Bellogín, A., P. Castells, and I. Cantador. 2017. "Statistical Biases in Information Retrieval Metrics for Recommender Systems." *Information Retrieval* 20(6): 606–34.

Bertin-Mahieux, T., D. P. W. Ellis, B. Whitman, and P. Lamere. 2011. "The Million Song Dataset." In *Proceedings of the ISMIR*, 591–6.

Buckley, C., D. Dimmick, I. Soboroff, and E. Voorhees. 2007. "Bias and the Limits of Pooling for Large Collections." *Information Retrieval* 10(6): 491–508.

Buckley, C., and E. M. Voorhees. 2000. "Evaluating Evaluation Measure Stability." In *Proceedings of the SIGIR*, 33–40. New York, NY, USA: ACM.

Buckley, C., and E. M. Voorhees. 2004. "Retrieval Evaluation with Incomplete Information." In *Proceedings of the SIGIR*, 25–32. New York, NY, USA: ACM.

Büttcher, S., C. L. A. Clarke, P. C. K. Yeung, and I. Soboroff. 2007. "Reliable Information Retrieval Evaluation with Incomplete and Biased Judgements." In *Proceedings of the SIGIR*, 63–70. New York, NY, USA: ACM.

Cañamares, R., and P. Castells. 2017. "A Probabilistic Reformulation of Memory-based Collaborative Filtering: Implications on Popularity Biases." In *Proceedings of the SIGIR*, 215–24. New York, NY, USA: ACM.

Cañamares, R., and P. Castells. 2018. "Should I Follow the Crowd? A Probabilistic Analysis of the Effectiveness of Popularity in Recommender Systems." In *Proceedings of the SIGIR*, 415–24. New York, NY, USA: ACM.

Cañamares, R., and P. Castells. 2020. "On target item sampling in offline recommender system evaluation." In *Proceedings of the RecSys*, 259–68. New York, NY, USA: ACM.

Cañamares, R., P. Castells, and A. Moffat. 2020. "Offline evaluation options for recommender systems." *Information Retrieval* 23(4): 387–411.

Castells, P., and R. Cañamares. 2018. "Characterization of Fair Experiments for rRecommender System Evaluation: A Formal Analysis." In *Proceedings of the RecSys 2018 Workshop on Offline Evaluation for Recommender Systems*, REVEAL 2018.

Castells, P., N. J. Hurley, and S. Vargas. 2015. "Novelty and Diversity in Recommender Systems." In *Recommender Systems Handbook*, 2nd ed., edited by F. Ricci, L. Rokach, and B. Shapira, 881–18. New York, NY, USA: Springer.

Chaney, A. J. B., B. M. Stewart, and B. E. Engelhardt. 2018. "How Algorithmic Confounding in Recommendation Systems Increases Homogeneity and Decreases Utility." In *Proceedings of the RecSys*, 224–32. New York, NY, USA: ACM.

Chapelle, O., D. Metzler, Y. Zhang, and P. Grinspan. 2009. "Expected R eciprocal Rank for Graded Relevance. In *Proceedings of the CIKM*, 621–30. New York, NY, USA: ACM.

Cremonesi, P., Y. Koren, and R. Turrin. 2010. "Performance of Recommender Algorithms on Top-n Recommendation Tasks." In *Proceedings of the RecSys*, 39–46. New York, NY, USA: ACM.

Ferrari Dacrema, M., S. Boglio, P. Cremonesi, and D. Jannach. 2021. "A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research." *ACM Transactions on Information Systems* 39(2): 20:1–20:49.

Fleder, D. M., and K. Hosanagar. 2009. "Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity." *Management Science* 55(5): 697–712.

Garcin, F., B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, and A. Huber. 2014. "Offline and Online Evaluation of News Recommender Systems at swissinfo.ch." In *Proceedings of the RecSys*, 169–76. New York, NY, USA: ACM.

Gilotte, A., C. Calauzènes, T. Nedelec, A. Abraham, and S. Dollé. 2018. 'Offline A/B Testing for Recommender Systems." In *Proceedings of the WSDM*, 198–206. New York, NY, USA: ACM.

Gomez-Uribe, C. A., and N. Hunt. 2015. "The Netflix Recommender System: Algorithms, Business Value, and Innovation." *ACM Transactions on Management Information Systems* 6(4): 13.

Gruson, A., P. Chandar, C. Charbuillet, J. McInerney, S. Hansen, D. Tardieu, and B. Carterette. 2019. 'Offline Evaluation to Make Decisions About Playlist Recommendation." In *Proceedings of the WSDM*, 420–8. New York, NY, USA: ACM.

Gunawardana, A., and G. Shani. 2015. "Evaluating Recommender Systems." In *Recommender Systems Handbook*, 2nd ed., edited by F. Ricci, L. Rokach, and B. Shapira, 265–308. New York, NY, USA: Springer.

Harman, D. K., ed. 1992. "Overview of the First Text REtrieval Conference (TREC-1)." *Proceedings of the TREC, Volume 500-207 of NIST Special Publication*. Gaithersburg, MD, USA: NIST.

Harman, D. K. 2005. "The TREC Test Collections." In *TREC: Experiment and Evaluation in Information Retrieval. Chapter 2*, edited by E. M. Voorhees, and D. K. Harman, 21–52. Cambridge, MA: MIT Press.

Harper, F. M., and J. A. Konstan. 2016. "The Movielens Datasets: History and Context." *ACM Transactions on Interactive Intelligent Systems* 5(4): 19:1–19:19.

Herlocker, J. L., J. A. Konstan, A. Borchers, and J. Riedl. 1999. "An Algorithmic Framework for Performing Collaborative Filtering." In *Proceedings of the SIGIR*, 230–7. New York, NY, USA: ACM.

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

Huang, J., H. Oosterhuis, M. de Rijke, and H. van Hoof. 2020. "Keeping Dataset Biases Out of the Simulation: A Debiased Simulator for Reinforcement Learning Based Recommender Systems. In *Proceedings of the RecSys*, 190–9. New York, NY, USA: ACM.

Jadidinejad, A. H., C. Macdonald, and I. Ounis. 2021. "The Simpson's Paradox in the Offline Evaluation of Recommendation Systems." *ACM Transactions on Information Systems* 40(1): 4:1–4:22.

Jannach, D., and M. Jugovac. 2019. "Measuring the Business Value of Recommender Systems." *ACM Transactions on Management Information Systems* 10(4): 1–23.

Jannach, D., L. Lerche, I. Kamehkhosh, and M. Jugovac. 2015. "What Recommenders Recommend: An Analysis of Recommendation Biases and Possible Countermeasures." *User Modeling and User-Adapted Interaction* 25(5): 427–91.

Järvelin, K., and J. Kekäläinen, 2002. "Cumulated Gain-based Evaluation of IR Techniques." *ACM Transactions on Information Systems* 20(4): 422–46.

Kawale, J., H. H. Bui, B. Kveton, L. Tran-Thanh, and S. Chawla. 2015. "Efficient Thompson Sampling for Online Matrix-Factorization

Recommendation." In *Proceedings of the NIPS*, 1297–305. Cambridge, MA, USA: MIT Press.

Kelly, D. 2009. "Methods for Evaluating Interactive Information Retrieval Systems with Users." *Foundations and Trends in Information Retrieval* 3(1-2): 1–224.

Koren, Y. 2008. "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model." In *Proceedings of the KDD*, 426–34. New York, NY, USA: ACM.

Krichene, W., and S. Rendle. 2020. "On Sampled Metrics for Item Recommendation." In *Proceedings of the KDD*, 1748–57. New York, NY, USA: ACM.

Li, L., W. Chu, J. Langford, and R. E. Schapire. 2010. "A Contextual-Bandit Approach to Personalized News Article Recommendation." In *Proceedings of the WWW*, 661–70. New York, NY, USA: ACM.

Li, L., W. Chu, J. Langford, and X. Wang. 2011. "Unbiased Offline Evaluation of Contextual-Bandit-based News Article Recommendation Algorithms." In *Proceedings of the WSDM*, 297–306. New York, NY, USA: ACM.

Lipani, A., D. E. Losada, G. Zuccon, and M. Lupu. 2021. "Fixed-cost Pooling Strategies." *IEEE Transactions on Knowledge and Data Engineering* 33(4): 1503–22.

Liu, F., A. Moffat, T. Baldwin, and X. Zhang. 2016. "Quit While Ahead: Evaluating Truncated Rankings." In *Proceedings of the SIGIR*, 953–6. New York, NY, USA: ACM.

Lu, H., W. Ma, M. Zhang, M. de Rijke, Y. Liu, and S. Ma. 2021. "Standing in Your Shoes: External Assessments for Personalized Recommender Systems." In *Proceedings of the SIGIR*, 415–24. New York, NY, USA: ACM.

Luo, C., Y. Liu, T. Sakai, F. Zhang, M. Zhang, and S. Ma. 2017. "Evaluating Mobile Search with Height-biased Gain." In *Proceedings of the SIGIR*, 435–44. New York, NY, USA: ACM.

Marlin, B. M., and R. S. Zemel. 2009. "Collaborative Prediction and Ranking with Non-random Missing Data." In *Proceedings of the RecSys*, 5–12. New York, NY, USA: ACM.

Marlin, B. M., R. S. Zemel, S. T. Roweis, and M. Slaney. 2007. "Collaborative Filtering and the Missing at Random Assumption." In *Proceedings of the UAI*, 267–75.

Maxwell, D., L. Azzopardi, K. Järvelin, and H. Keskustalo. 2015. "Searching and Stopping: An Analysis of Stopping Rules and Strategies." In *Proceedings of the CIKM*, 313–22. New York, NY, USA: ACM.

Mehrotra, R., J. McInerney, H. Bhouchard, M. Lalmas, and F. Diaz. 2018. "Towards a Fair Marketplace: Counterfactual Evaluation of the Trade-off between Relevance, Fairness & Satisfaction in Recommendation Systems." In *Proceedings of the CIKM*, 2243–51. New York, NY, USA: ACM.

Mena-Maldonado, E., R. Cañamares, P. Castells, Y. Ren, and M. Sanderson. 2021. "Popularity Bias in False-positive Metrics for Recommender Systems Evaluation." *ACM Transactions on Information Systems* 39(3): 36:1–36:43.

Meng, Z., R. McCreadie, C. Macdonald, and I. Ounis. 2020. "Exploring Data Splitting Strategies for the Evaluation of Recommendation Models." In *Proceedings of the RecSys*, 681–6. New York, NY, USA: ACM.

Moffat, A., and J. Zobel. 2008. "Rank-biased Precision for Measurement of Retrieval Effectiveness."*ACM Transactions on Information Systems* 27(1): 2.1–2.27.

Moffat, A., P. Bailey, F. Scholer, and P. Thomas. 2017. "Incorporating User Expectations and Behavior Into the Measurement of Search Effectiveness." *ACM Transactions on Information Systems* 35(3): 24:1–24:38.

Moffat, A., W. Webber, and J. Zobel. 2007. "Strategic System Comparisons via Targeted Relevance Judgments." In *Proceedings of the SIGIR*, 375–82. New York, NY, USA: ACM.

Pérez Maurera, F. B., M. Ferrari Dacrema, L. Saule, M. Scriminaci, and P. Cremonesi. 2020. "Contentwise Impressions: An Industrial Dataset with Impressions Included." In *Proceedings of the CIKM*, 3093–100. New York, NY, USA: ACM.

Rohde, D., S. Bonner, T. Dunlop, F. Vasile, and A. Karatzoglou. 2018. "Recogym: A Reinforcement Learning Environment for the Problem of Product Recommendation in Online Advertising." In *Proceedings of the RecSys Workshop on Offline Evaluation for Recommender Systems*, REVEAL 2018.

Said, A., and A. Bellogín. 2014. "Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks." In *Proceedings of the RecSys*, 129–36. New York, NY, USA: ACM.

Sakai, T. 2007. "Alternatives to BPref." In *Proceedings of the SIGIR*, 71–8. New York, NY, USA: ACM.

Sakai, T., and N. Kando. 2008. "On Information Retrieval Metrics Designed for Evaluation with Incomplete Relevance Assessments." *Information Retrieval* 11(5): 447–70.

Sanderson, M., and J. Zobel. 2005. "Information Retrieval System Evaluation: Effort, Sensitivity, and Reliability." In *Proceedings of the SIGIR*, 162–9. New York, NY, USA: ACM.

Sanderson, M. 2010. "Test Collection Based Evaluation of Information Retrieval Systems." *Foundations and Trends in Information Retrieval* 4(4): 247–375.

Sanz-Cruzado, J., P. Castells, and E. López. 2019. "A Simple Multiarmed Nearest-neighbor Bandit for Interactive Recommendation." In *Proceedings of the RecSys*, 358–62. New York, NY, USA: ACM.

Schnabel, T., A. Swaminathan, A. Singh, N. Chandak, and T. Joachims. 2016. "Recommendations as Treatments: Debiasing Learning and Evaluation." In Proceedings of the ICML, 1670–9. Sheffield, UK.

Shardanand, U., and P. Maes. 1995. "Social Information Filtering: Algorithms for Automating "Word of Mouth"." In *Proceedings of the CHI*, 210–7. New York, NY, USA: ACM.

Smucker, M. D., and C. L. A. Clarke. 2012. "Time-based Calibration of Effectiveness Measures." In *Proceedings of the SIGIR*, 95–104. New York, NY, USA: ACM.

Steck, H. 2010. "Training and Testing of Recommender Systems on Data Missing Not at Random." In *Proceedings of the KDD*, 713–22. New York, NY, USA: ACM.

Steck, H. 2011. "Item Popularity and Recommendation Accuracy." In *Proceedings of the RecSys*, 125–32. New York, NY, USA: ACM.

Steck, H. 2013. "Evaluation of Recommendations: Rating Prediction and Ranking." In *Proceedings of the RecSys*, 213–20. New York, NY, USA: ACM.

Sun, Z., D. Yu, H. Fang, J. Yang, X. Qu, J. Zhang, and C. Geng. 2020. "Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison." In *Proceedings of the RecSys*, 23–32. New York, NY, USA: ACM.

Sutton, R. S., and A. G. Barto. 2018. *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: Bradford Books.

Swaminathan, A., A. Krishnamurthy, A. Agarwal, M. Dudík, J. Langford, D. Jose, and I. Zitouni. 2017. "Off-policy Evaluation for Slate Recommendation." In *Proceedings of the NIPS*, 3635–45. Red Hook, NY, USA: Curran Associates, Inc.

Valcarce, D., A. Bellogín, J. Parapar, and P. Castells. 2020. "Assessing Ranking Metrics in Top-n Recommendation." *Information Retrieval* 23(4): 411–48.

Vargas, S., and P. Castells. 2011. "Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems." In *Proceedings of the RecSys*, 109–16. Chicago, Illinois, USA: ACM.

Wang, Q., C. Zeng, W. Zhou, T. Li, S. S. Iyengar, L. Shwartz, and G. Y. Grabarnik. 2019. "Online Interactive Collaborative Filtering using Multi-armed Bandit with Dependent Arms." *IEEE Transactions on Knowledge and Data Engineering* 31(8): 1569–80.

Wang, X., R. Zhang, Y. Sun, and J. Qi. 2021. "Combating Selection Biases in Recommender Systems with a Few Unbiased Ratings." In *Proceedings of the WSDM*, 427–35. New York, NY, USA: ACM.

Yang, L., Y. Cui, Y. Xuan, C. Wang, S. Belongie, and D. Estrin. 2018. "Unbiased Offline Recommender Evaluation for Missing-not-at-random Implicit Feedback." In *Proceedings of the RecSys*, 279–87. New York, USA: ACM

Zhang, F., Y. Liu, X. Li, M. Zhang, Y. Xu, and S. Ma. 2017. "Evaluating Web Search with a Bejeweled Player Model." In *Proceedings of the SIGIR*, 425–34. New York, NY, USA: ACM.

Zhou, T., Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang. 2010. "Solving the Apparent Diversity-accuracy Dilemma of Recommender Systems." *The Proceedings of the National Academy of Sciences* 107(10): 4511–5.

Ziegler, C.-N., S. M. McNee, J. A. Konstan, and G. Lausen. 2005. "Improving Recommendation Lists through Topic Diversification." In *Proceedings of the WWW*, 22–32. New York, NY, USA: ACM.

Zobel, J. 1998. "How Reliable are the Results of Large-scale Information Retrieval Experiments?" In *Proceedings of the SIGIR*, 307–14. New York, NY, USA: ACM.

AUTHOR BIOGRAPHIES

Pablo Castells is an Associate Professor at the Autónoma University of Madrid, and an Amazon Scholar. His research interests are in the fields of information retrieval and recommender systems, dealing with models, theory, algorithms, and evaluation. His recent research focuses on the design of evaluation experiments, algorithmic and evaluation bias, and novelty and diversity. Pablo has coauthored over 100 research publications in top tier outlets in these areas, and received the best paper award at SIGIR 2018.

Alistair Moffat graduated with a PhD from the University of Canterbury in New Zealand in 1986, and has been a faculty member at the University of Melbourne since then. He has published extensively in the area of information retrieval evaluation, and the connected areas of algorithms for text indexing, for text compression, and for keyword-based search; and is an author of three books and more than 250 refereed papers. Alistair was inducted into the SIGIR Academy in 2021.

How to cite this article: Castells, P., and A. Moffat. 2022. "Offline recommender system evaluation: Challenges and new directions." *AI Magazine* 43: 225–38.
https://doi.org/10.1002/aaai.12051