CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Debiased offline evaluation of recommender systems: A weighted-sampling approach
Authors	Carraro, Diego;Bridge, Derek G.
Publication date	2020-03
Original Citation	Carraro, D. and Bridge, D. (2020) 'Debiased offline evaluation of recommender systems: A weighted-sampling approach', in Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 30 March-3 April, pp. 1435-1442. doi: 10.1145/3341105.3375759
Type of publication	Conference item
Link to publisher's version	https://www.sigapp.org/sac/sac2020/ - 10.1145/3341105.3375759
Rights	© 2020 Association for Computing Machinery. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Download date	2026-11-06 04:36:34
Item downloaded from	https://hdl.handle.net/10468/10209

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Debiased Offline Evaluation of Recommender Systems: A Weighted-Sampling Approach

Diego Carraro
Insight Centre for Data Analytics
University College Cork, Ireland
diego.carraro@insight-centre.org

Derek Bridge
Insight Centre for Data Analytics
University College Cork, Ireland
derek.bridge@insight-centre.org

ABSTRACT

Offline evaluation of recommender systems mostly relies on historical data, which is often biased by many confounders. In such data, user-item interactions are Missing Not At Random (MNAR). Measures of recommender system performance on MNAR test data are unlikely to be reliable indicators of real-world performance unless something is done to mitigate the bias. One way that researchers try to obtain less biased offline evaluation is by designing new supposedly unbiased performance estimators for use on MNAR test data. We investigate an alternative solution, a *sampling approach*. The general idea is to use a sampling strategy on MNAR data to generate an *intervened* test set with less bias — one in which interactions are Missing At Random (MAR) or, at least, one that is more MAR-like. An example of this is SKEW, a sampling strategy that aims to adjust for the confounding effect that an item's popularity has on its likelihood of being observed.

In this paper, we propose a novel formulation for the sampling approach. We compare our solution to SKEW and to two baselines which perform a random intervention on MNAR data (and hence are equivalent to no intervention in practice). We empirically validate for the first time the effectiveness of SKEW and we show our approach to be a better estimator of the performance one would obtain on (unbiased) MAR test data. Our strategy benefits from high generality properties (e.g. it can also be employed for training a recommender) and low overheads (e.g. it does not require any learning).

KEYWORDS

Offline evaluation, bias, intervened test sets

1 INTRODUCTION

Offline evaluation of a recommender system is done using an *observed dataset*, which records interactions (e.g. clicks, purchases, ratings) that occur between users and items during a given period in the operation of the recommender system. However, this dataset is *biased*, not only due to the freedom that users have in choosing which items to interact with, but also due to other factors, known as confounders ([5, 27]). For example, the user-interface plays an important role: differences in the ways that items are exposed to users (e.g. position on the screen) influence the likelihood of a user interacting with those items [14]. The recommender itself sets up a feedback loop, which results in another confounder: users are typically more likely to interact with the recommender's suggestions than with other items. The user's preferences are also a confounder: for example, Marlin et al. demonstrate that, in a dataset of numeric ratings, the probability of not observing a specific user-item interaction depends on the value associated with that particular interaction (i.e. the rating value): informally, users tend to rate items that they like [18]. Because of these and other confounders, interactions that are missing from an observed dataset are Missing Not At Random (MNAR) [18].

Classical offline evaluations using such an observed dataset are in effect making the assumption that interactions that are missing from the observed dataset are either Missing Completely At Random (MCAR) or Missing At Random (MAR) [18]. (For the distinction between MCAR and MAR, see Section 2.) Using MNAR data in an evaluation as if it were MCAR or MAR, results in biased estimates of a recommender's performance [18]: for example, such experiments tend to incorrectly reward recommenders that recommend popular items or that make recommendations to the more active users [8, 21].

There are three ways of addressing this problem. The most straightforward approach (in theory, at least) is to collect and employ a MAR dataset instead of an MNAR one for the offline evaluation. Using (unbiased) MAR data for the evaluation would give an unbiased estimate of the recommender's performance. In some domains, there are ways of collecting small MAR-like datasets (see Section 2). But, in many domains it is either impractical or too expensive to obtain MAR-like datasets.

Because of the difficulty of collecting MAR-like data, the other two ways of addressing the problem focus on using MNAR data (which is usually available and in larger quantities) but mitigating its bias. One way of doing this is to design estimators (i.e. evaluation metrics) which compensate for the bias in the MNAR test data. Although this achieves the desired goal to some extent, unbiased estimators suffer from two potential drawbacks. The first is that they may not be general enough to overcome all sources of bias, i.e. they are often designed to compensate for a specific kind of bias: for example, the accuracy metric that is proposed in [24] is able to correct only for the long-tail popularity bias in a dataset. The second drawback that affects unbiased estimators is that their unbiasedness might be proven only if the data satisfies some specific conditions: the ATOP estimator proposed in [23], for example, is unbiased only if the data satisfies two conditions.

The third approach is to *intervene* on MNAR test data before using it for the evaluation. In practice, such intervention is performed by means of a *sampling strategy* which samples from the available MNAR test data. The sampling strategy is chosen so that the intervened test set which results from the sampling is supposed to be less biased (more MAR-like) and therefore more suitable for

xxx, xxx, xxx
© 2020 xxx

evaluation of the recommender's performance. One such sampling strategy is known as SKEW [13]: it samples user-item interactions in inverse proportion to item popularity, thus producing test data with reduced popularity bias.

In this paper we investigate a new alternative to the SKEW sampling strategy for generating intervened data. We propose a *weighted* sampling strategy in which the weights are calculated by considering the divergence between the distribution of users and items in the MNAR data and their corresponding target (unbiased) MAR distributions.

We compare our sampling approach with SKEW. Our experiments allow us: to empirically evaluate for the first time the effectiveness of SKEW; to verify that both strategies successfully perform the desired debiasing action; but also to demonstrate that our strategy more closely approximates the unbiased performances of different recommender algorithms.

Although in this paper we employ our technique to generate a *test set* for offline recommender evaluation, our approach is general and can also be employed to debias the data used for training a recommender.

The rest of this paper is organized as follows. Section 2 presents related work. In Section 3, we propose a probabilistic framework to study properties of MAR and MNAR datasets. In Section 4 we use the properties presented in Section 3 to derive our weighted sampling strategy, which is used to generate intervened test sets. Section 5 describes the experiments we have run to assess the effectiveness of our approach. We analyse the results of the experiments in Section 6. We discuss our findings in Section 7.

2 RELATED WORK

A distinction is sometimes drawn between Missing Completely At Random (MCAR) and Missing At Random (MAR). The distinction is based on missing data analysis theory and is first proposed by [16] and later introduced into the recommender systems literature by [18]. Indeed, MCAR, MAR and MNAR are terms used to denote different *missing data mechanisms* which describe the process that generates the observation pattern in the data. In work on causal inference, the same process is typically called the *assignment mechanism* instead [10]. In [16, 18], MCAR means that whether a user-item interaction is missing does not depend on interaction values (such as ratings in a recommender) at all, i.e. it depends neither on the observed interaction values nor the missing interaction values. MAR, on the other hand, means that whether a user-item interaction is missing may depend on the observed interaction values, but is independent of the missing interaction values.

In this paper, we use MNAR and MAR in a more informal and general way. We use MNAR to indicate that data is biased (missing interactions depend on some confounders in the data), and we use MAR to mean that data is unbiased (missing interactions do not depend on any confounder in the data, whether it is observed or not). Although these more informal usages are not properly in line with the categorization in [16] and [18], our choice is broadly in line with other work in the recommender systems literature: what we refer to as MAR is also called MAR in papers such as [4, 23] and what we call MAR is referred to as MCAR in, e.g., [22].

A substantial body of work has been done in the last few years to cope with bias in recommenders, both for their training and their offline evaluation. We focus here more on the latter, as it is more relevant to our work in this paper.

As we mentioned in Section 1, one approach is to collect a separate MAR-like dataset (i.e. one that is as devoid of bias as possible) to use for the evaluation of the recommender's performance. This is usually done by means of what we will call a "forced rating approach" [4]. User-item pairs are chosen uniformly at random and for each user-item pair that gets selected the user is required (forced) to provide a rating for the item. In this way, from the data that we collect we remove biases such as the item discovery bias (because items are randomly chosen for users), item consumption bias (because users are forced to consume or interact with the item so that they can rate it, unless the item was already known to the user) and rating decision bias (because users are not free whether to rate the chosen item or not, they are forced to do it) [4].

Datasets collected by the "forced rating approach" are MAR-like, rather than MAR: they may still carry some bias. When building such a dataset, for example, although invitations are sent to users who are chosen uniformly at random, those who agree to participate may be atypical, thus introducing bias. Equally, the fact that, for each user, items to rate are presented sequentially introduces bias: the rating a user assigns to a particular item may be influenced by the items she has rated so far. Although this means that these datasets are less biased, rather than unbiased, to the best of our knowledge, this is still the best way of collecting this type of data.

Datasets of this kind include Webscope R3 [18] and cm100k [4] in the music domain, and CoatShopping [22] in the clothing domain. The "forced rating approach" can only work in certain domains; for example, it requires that a user who is presented with an item can quickly consume that item so as to provide a rating. In the movie domain, for example, we almost certainly cannot require a user to watch an entire movie (although we could require them to watch a movie trailer).

Therefore, because in some domains obtaining a MAR-like dataset may be impractical, most work on unbiased offline evaluation of recommenders still relies on the use of MNAR datasets. The majority of the literature tries to overcome the bias in an MNAR test set by proposing new estimators (i.e. evaluation metrics) which provide unbiased or nearly unbiased measures of performance on the MNAR test data. Steck describes ATOP, for example, a new ranking estimator which is unbiased under specific mild assumptions about the data employed [23, 25]. Steck also proposes an accuracy metric that is able to correct for the long-tail popularity bias in the data, resulting in a nearly unbiased estimate of the true accuracy under the assumption that no other confounders besides the so-called popularity bias occurs [24]. There is work too on unbiased estimators for implicit MNAR data. An example of this appears in [15], where the authors proposed a missing data model and a novel evaluation measure, i.e. Average Discounted Gain (ADG), built upon the widely used NDCG metric. They show ADG allows unbiased estimation with respect to their missing data model, unlike NDCG. Other work uses Inverse-Propensity-Scoring (IPS) techniques (e.g. [22, 28]). A propensity is the probability that a particular user-item pair is observed. This work on IPS uses propensities as a proxy to build unbiased estimators on explicit ([22]) and implicit ([28]) data

respectively. One drawback of propensities is that their estimation might require an expensive learning step (e.g. [22, 28]).

There are those who use what we are calling an intervention approach. They sample from the MNAR test set to produce a smaller MAR-like test set (the intervened set), which they use in the evaluation in place of the MNAR test set. One such method is Lang et al.'s SKEW method, which samples user-item pairs in inverse proportion to the item popularity. This generates an intervened test set which has roughly uniform exposure distribution across items, thus reducing the item popularity bias in the test set [13]. Lang et al. in [13, 27] and Bonner et al. in [3] use this technique for test set generation to evaluate causal approaches to recommendation. However, none of the three works that we have just cited either explain or verify empirically why SKEW should be effective as a debiasing technique. In this paper we fill the gap by providing such contributions. Also, because of the similarity with our work, we use SKEW as a state-of-the-art strategy to compare against our own approach.

Bellogin et al. also sample an MNAR dataset to try to obtain a fairer evaluation [2]. Their first approach is a form of stratification, in which test items are sampled from a popularity-based partition of the data. Their second approach builds a test set with the same number of ratings for each item. Compared with our work, their approaches are more limited since both have the goal only of reducing popularity bias. Their approaches may also result in quite small tests sets, especially if the popularity curve in the original dataset is quite steep.

To conclude this review, and for completeness, we mention some of the work that has applied debiasing techniques when training recommender systems. In [9, 12, 17], for example, existing algorithms are adapted to include explicit MNAR data models. Others employ unbiased estimators as a loss function to train their model and therefore correct for the bias in the training set (e.g. [15, 23, 24]), while others take a causal inference perspective (e.g. [11, 13, 14, 26]).

3 PROPERTIES OF DATASETS: A PROBABILISTIC FRAMEWORK

In this section, we define a probabilistic framework to analyse properties of MAR and MNAR datasets. Then, in Section 4, we use these properties to design our approach that generates intervened test sets for 'unbiased' evaluation.

We consider a user-item space, $U \times I$, of size $|U| \cdot |I|$. We denote with $u \in U = \{1, .., |U|\}$ a generic user and with $i \in I = \{1, .., |I|\}$ a generic item. We denote with $D = \{O \in \{0,1\}^{U \times I}, Y \in \mathbb{R}^{U \times I}\}$ a generic observed dataset. The binary matrix O records which interactions between users and items have been observed: $O_{u,i} = 1$ if an interaction is observed and $O_{u,i} = 0$ otherwise. We also define the associated matrix $Y \in \mathbb{R}^{U \times I}$ which records the value of the interactions of the corresponding observed entries in O: we have $Y_{u,i} \neq 0$ where $O_{u,i} = 1$, $Y_{u,i} = 0$ otherwise. When discussing Y, we use the general term "interaction value", rather than "rating", to emphasize the generality of our framework: Y can take values of any kind in $\mathbb{R}$ whether they denote ratings, number of clicks, number of views, listening frequencies, etc. We also define the binary random variable $\mathcal{O} : U \times I \rightarrow \{0, 1\}$ over the set of user-item pairs in O as $\mathcal{O} = 1$ if the user-item interaction is observed and $\mathcal{O} = 0$ otherwise. (But later we will use abbreviation $P(\mathcal{O})$ in place of $P(\mathcal{O} = 1)$.) Using this notation, we can refer to two kinds of datasets over the same $U \times I$ space, $D_{mnar} = \{O^{mnar}, Y^{mnar}\}$ and $D_{mar} = \{O^{mar}, Y^{mar}\}$, which have MNAR and MAR properties respectively.

3.1 Properties of a MAR dataset

We will formally describe how D^{mar} is generated. We make use of the forced ratings approach that we described in Section 2. First, we need to randomly sample a set of user-item pairs in order to generate O^{mar}. Then, a preference (interaction value) for each pair in O^{mar} is collected so that Y^{mar} is obtained. Note that, in order to satisfy the MAR property, the generation of O^{mar} is totally independent from Y^{mar} and from the particular user-item pair (u, i) as well. We also assume that, once O^{mar} is determined, we can obtain interaction values Y^{mar} for all user-item pairs in O^{mar}. (In practice, of course, users may decline the invitation to participate or may refuse to give some ratings, which is one reason why in reality these datasets are MAR-like and not MAR.)

To achieve the goal, we make use of the probability distribution $P_{mar}(\mathcal{O}|u, i)$, defined over the space $U \times I$, that leads to O^{mar}. A straightforward choice is to set $P_{mar}(\mathcal{O}|u, i) = P(\mathcal{O}) = \rho_{mar}$, where ρ_{mar} represents the desired ratio of observed entries from $U \times I$.

Now, assuming that a dataset D^{mar} has been collected using such an approach, we should empirically verify that user and item posterior probabilities are (roughly) uniformly distributed:

$$P_{mar}(u|\mathcal{O}) = \frac{|O^{mar}_u|}{|O^{mar}|} \approx \frac{1}{|U|} \quad \forall u \in U \tag{1}$$

$$P_{mar}(i|\mathcal{O}) = \frac{|O^{mar}_i|}{|O^{mar}|} \approx \frac{1}{|I|} \quad \forall i \in I \tag{2}$$

where O^{mar}_u and O^{mar}_i are the observed interactions in O^{mar} for user u and item i respectively.

Also, because users and items are drawn independently, we have that their posteriors are independent and we can write:

$$P_{mar}(u, i|\mathcal{O}) = P_{mar}(u|\mathcal{O})P_{mar}(i|\mathcal{O}) \approx \frac{1}{|U||I|} \quad \forall (u, i) \in U \times I \tag{3}$$

for the joint posterior of a specific user-item pair.

3.2 Properties of an MNAR dataset

MNAR data is, of course, usually collected during the operation of a recommender system. But, similarly to the way we modelled the generation of MAR data, we can model the generation of a MNAR dataset $D_{mnar} = \{O^{mnar}, Y^{mnar}\}$ in terms of a drawing process which determines O^{mnar} first and Y^{mnar} subsequently.

Differently from the MAR scenario, due to the presence of bias, we cannot assume the sampling distribution P_{mnar} to be independent from the interaction values Y^{mnar} (or from other confounders too, including, e.g., the specific user and item (u, i)). In other words, in an MNAR dataset the draw is generally guided by some unknown probability $P_{mnar}(\mathcal{O}|u, i, Y, \mathcal{X})$, where $Y \supset Y^{mnar}$ represents the complete set of user-item interactions and $\mathcal{X}$ represents a set of features (covariates, confounders) which influences the sampling

probability (e.g. user demographics, item features, characteristics of the system such as the way it exposes items to users, and so on).

If a MNAR dataset D^{mnar} has been collected, we can examine user and item posterior probabilities in O^{mnar}, as we did for the MAR dataset but now, in general, we will find:

$$P_{mnar}(u|\mathcal{O}) = \frac{|O^{mnar}_u|}{|O^{mnar}|} \neq \frac{1}{|U|} \quad \forall u \in U \tag{4}$$

$$P_{mnar}(i|\mathcal{O}) = \frac{|O^{mnar}_i|}{|O^{mnar}|} \neq \frac{1}{|I|} \quad \forall i \in I \tag{5}$$

In general, the users and items are not uniformly distributed and thus, given that a specific entry is observed, i.e. $\mathcal{O} = 1$, we cannot assume the user and item posterior independence for the joint posterior $P_{mnar}(u, i|\mathcal{O})$, i.e.

$$P_{mnar}(u, i|\mathcal{O}) \neq P_{mnar}(u|\mathcal{O})P_{mnar}(i|\mathcal{O}) \quad \forall (u, i) \in U \times I \tag{6}$$

However, the formulation that we have given here provides us with a solid framework to design our debiasing strategy in the next section.

4 INTERVENED TEST SETS

To conduct unbiased evaluation from biased data, we generate and use intervened test sets in place of classical random heldout test sets. We begin by presenting this approach in general (Section 4.1), and then we present the specifics of our approach (Sections 4.2 and 4.3).

4.1 The sampling approach

The sampling approach consists in performing a debiasing intervention on MNAR data D^{mnar} by means of a given sampling strategy, denoted with S. The result of the intervention is the dataset $D_S = \{O^S \subset O^{mnar}, Y^S \subset Y^{mnar}\}$, with the objective that D_S has unbiased-like properties. We follow the same reasoning adopted to study properties of MAR and MNAR datasets. Thus, we generate O^S first and then we obtain Y^S accordingly.

The sampling is performed on the space O^{mnar}, ignoring interaction values in Y^{mnar}. We denote with $\mathcal{S} : U \times I \rightarrow \{0, 1\}$ the binary random variable that guides the sampling. $\mathcal{S} = 1$ when a particular user-item pair is sampled from O^{mnar}, 0 otherwise. (Again, we will use abbreviation $P(\mathcal{S})$ in place of $P(\mathcal{S} = 1)$.) In practice, a particular strategy S is characterized by the expression of the probability $P_S(\mathcal{S}|u, i), \forall (u, i) \in O^{mnar}$, which is the probability distribution responsible for guiding the sampling on O^{mnar}. We present our sampling approach in the next subsection. In Section 5, we will also define P_S for SKEW and for two baseline approaches that we compare against in the experiments.

4.2 Our approach: weights for the sampling

In the presentation of our approach, we will start by assuming the availability of some MAR-like data O^{mar} in addition to MNAR data O^{mnar}. In fact, we will see in Section 4.3 that we can use our approach even in cases where we do not have any MAR data.

Our main idea is to make the posterior probability distribution of each user-item pair in the sampled O^S, i.e. $P_S(u, i|\mathcal{S})$, approximately the same as the posterior probability distribution observed for the corresponding user-item pair in O^{mar}, i.e. $P_{mar}(u, i|\mathcal{O})$. In other words, we want to make O^S similar to O^{mar} in terms of its posteriors. Writing this as a formula, we want:

$$P_S(u, i|\mathcal{S}) \approx P_{mar}(u, i|\mathcal{O}) \quad \forall (u, i) \in O^S \tag{7}$$

To obtain this approximation, we adjust the posterior distributions of the sampling space O^{mnar}, i.e. $P_{mnar}(u, i|\mathcal{O})$, using user-item weights $w = (w_{ui})_{u \in U, i \in I}$ (similarly to [19]). We denote the modified weighted MNAR posteriors by $P_{mnar}(u, i|\mathcal{O}, w)$. The goal is to find weights w so that:

$$P_{mnar}(u, i|\mathcal{O}, w) = P_{mar}(u, i|\mathcal{O}) \quad \forall (u, i) \in O^{mnar} \tag{8}$$

From the fact that a typical MAR dataset is uniformly distributed over users and items, we use the independence of formula 3 to re-write the right-hand side of formula 8 to obtain:

$$P_{mnar}(u, i|\mathcal{O}, w) = P_{mar}(i|\mathcal{O})P_{mar}(u|\mathcal{O}) \quad \forall (u, i) \in O^{mnar} \tag{9}$$

Similarly to formula 6 which considers user and item MNAR posteriors, user and item *weighted* MNAR posteriors will not in general be independent. However, we are going to treat them as if they were independent, to obtain the following:

$$P_{mnar}(u, i|\mathcal{O}, w) = P_{mnar}(i|\mathcal{O}, w)P_{mnar}(u|\mathcal{O}, w) \quad \forall (u, i) \in O^{mnar} \tag{10}$$

While formula 10 is not true in general, we justify it by showing empirically in Section 6 that it does obtain good results.

Now, using 10, we can split formula 9 into the two following equations:

$$P_{mnar}(u|\mathcal{O}, w) = P_{mar}(u|\mathcal{O}) \quad \forall u \in U \tag{11}$$

$$P_{mnar}(i|\mathcal{O}, w) = P_{mar}(i|\mathcal{O}) \quad \forall i \in I \tag{12}$$

As a consequence of formulas 11 and 12 for the weighted MNAR posteriors, we can define and calculate user-specific weights $w = (w_u)_{u \in U}$ and item-specific weights $w = (w_i)_{i \in I}$ instead of weights that are user-item specific.[1]

We propose the most straightforward solution to model the weighted MNAR posteriors, i.e. $P_{mnar}(.|\mathcal{O}, w) = w_. P_{mnar}(.|\mathcal{O})$. We plug this into formulas 11 and 12 and we obtain $w_u P_{mnar}(u|\mathcal{O}) = P_{mar}(u|\mathcal{O})$, $w_i P_{mnar}(i|\mathcal{O}) = P_{mar}(i|\mathcal{O})$ for each user and item weighted distribution respectively. Simply reversing these last two formulas, we have the expressions for calculating the weights:

$$w_u = \frac{P_{mar}(u|\mathcal{O})}{P_{mnar}(u|\mathcal{O})} \quad \forall u \in U \tag{13}$$

$$w_i = \frac{P_{mar}(i|\mathcal{O})}{P_{mnar}(i|\mathcal{O})} \quad \forall i \in I \tag{14}$$

We can think of the calculated weights as quantities that measure the divergence between the MNAR distributions of the sampling space and the target MAR distribution. Because a specific weight adjusts the corresponding MNAR distribution, we directly use weights

[1] Having independent user and item weights also has an advantage in terms of scalability. We need to calculate only $|U| + |I|$ weights instead of $|U \times I|$. This is good for scalability because $|U \times I| >> |U| + |I|$ for the values of $|U|$ and $|I|$ that we find in recommender domains.

to model the sampling distribution, i.e. $P_S(\mathcal{S}|u,i) = w_u w_i$. During the sampling, the effect of the weights is to increase or decrease the probability that a particular user-item pair is sampled depending on how divergent are the user and item posterior probabilities in the MNAR sampling space with respect to the MAR distributions.

In fact, based on preliminary experiments, we use $P_S(\mathcal{S}|u,i) = w_u (w_i)^2$ instead. This variant, denoted by WTD in the rest of this paper, raises the importance of the item-weight relative to the user weight. Specifically, $(w_i)^2$ will be bigger than w_i if w_i is greater than one, and $(w_i)^2$ will be smaller than w_i if w_i is less than one. This choice makes sense in the light of previous research reported in the literature which identifies item popularity as one of the most impactful confounders in MNAR data, e.g. [21, 24].

4.3 Hypothesized distributions for the weights

Up to this point, we assumed the availability of some MAR-like data in order to give us the posteriors that we need to approximate. But MAR-like data is expensive or impossible to collect, as we discussed when presenting the "forced rating approach" earlier. Furthermore, in those cases where we do have a reasonable amount of MAR-like data at hand, we could use it directly as an unbiased test set. Using it to calculate weights so that we can intervene on MNAR data to produce a more MAR-like test set would then be pointless.

In fact, when we do not have any MAR-like data, we can still use our approach. We know that the posterior probability distribution for MAR data is uniform ($P_{mar}(u|\mathcal{O}) = 1/|U|$, $P_{mar}(i|\mathcal{O}) = 1/|I|$), and this is all we need for our sampling approach. Therefore, we can use this hypothesized distribution when calculating the weights, avoiding the need for a MAR-like dataset. We call this strategy, WTD_H (where the H stands for "hypothesized").

5 EXPERIMENTS

We have assessed WTD and WTD_H in offline experiments, which we describe in this section.

5.1 Datasets

We use two publicly available datasets: CoatShopping[2] from the clothing domain [22] and Webscope R3[3] from the music domain [18]. Both of them are ideal for our purposes because they are composed of two parts, one having MAR properties ($D_{mar} = \{O^{mar}, Y^{mar}\}$), and the other having MNAR properties ($D_{mar} = \{O^{mnar}, Y^{mnar}\}$). For both of them, interactions are in the form of ratings, so that $Y \in \{1,2,3,4,5\}^{U \times I}$. We consider a rating to be positive if it is above 3, and negative otherwise. Both the D_{mar} parts are collected using the forced ratings approach described earlier, therefore they are almost but not completely unbiased, for the reasons we gave earlier. The D_{mnar} portions are collected during the operation of a recommender system. Note that we did mention earlier that we know of one other MAR-like dataset, collected by the forced ratings approach, namely cm100k from the music domain [4], but we cannot use this in our experiments because it does not have any corresponding MNAR data.

[2]Available from https://www.cs.cornell.edu/~schnabts/mnar/
[3]Available on request from https://webscope.sandbox.yahoo.com

Table 1: Dataset statistics

	CoatShopping		Webscope R3	
	MAR	MNAR	MAR	MNAR
# ratings	4640	6960	54k	129k
# users	290	290	5400	5400
# items	300	300	1000	1000
avg # ratings per user	16	24	10	23
avg # ratings per item	15	23	54	129
avg rating	2.22	2.61	1.81	2.87
sparsity	0.94	0.92	0.99	0.97

For each dataset, we apply a preprocessing step to ensure both D_{mar} and D_{mnar} having a common user-item space $U \times I$: specifically, we keep those users and items that belong to the intersection of the two portions. Table 1 gives statistics of the final resulting datasets that we used in the experiments.

5.2 Methodology

The goal of the experiments is to assess the 'goodness' of different ways of producing intervened test sets. The measure of 'goodness' is how much results obtained by evaluating a recommender on an intervened test set resemble the results we would obtain on an unbiased test set.

In order to do that, in our experiments, we randomly split O^{mnar} in each dataset into a training set O^{tr} and a heldout set O^{he} with proportions 60%-40% respectively. Since the split is random, MNAR distributions are preserved. For both of them, we take the corresponding ratings from Y^{mnar} and we produce Y^{tr} and Y^{he}. Y^{he} is what one would use as a traditional test set. In our case, we use O^{he} as the sampling space: we sample it to obtain intervened test sets. There is one intervened test set per sampling strategy (REG, SKEW, WTD, WTD_H, explained in Section 5.3). We make the REG, SKEW, WTD, WTD_H intervened test sets to be 50% of the size of O^{he}. (Smaller values than 50% can result in intervened test sets that are too small to give reliable results; larger values than 50% can mean that intervened test sets are not appreciably different from O^{he}.)

We also randomly split O^{mar} into three, i.e. O^{w}, O^{val} and O^{gt} with proportions 15%-15%-70% respectively. Since the split is random, MAR distributions are preserved. We obtain Y^{w}, Y^{val} and Y^{gt} accordingly, as before. O^{w} is used to calculate the weights for WTD (see Section 5.3 for more details of the calculation). We use Y^{val} as the validation set to optimize recommender system hyperparameter values (Section 5.4). (In reality, the ratings one would use to optimize hyperparameter values would either be a portion of Y^{tr} or a portion of an intervened test set produced from Y^{he}. We decided it was better in the experiments that we report in this paper to minimise the effect of hyperparameter selection on our results. Hence, we selected hyperparameter values using 'unbiased' data, Y^{val}.)

We use Y^{gt} as an unbiased test set. In other words, the performance of a given recommender on Y^{gt} can be considered to be its "true", unbiased performance (the ground-truth). We want the performance of a recommender on an intervened test set to be close to its performance on this unbiased test set. The best intervention

strategy is the one that produces test sets where performance most closely resembles performance on Y^{gt}.

We train the five recommender systems presented in Section 5.4 using ratings in Y^{tr}. Each recommender produces a ranked list of recommendations which are tested on the unbiased test set Y^{gt} and the intervened test sets. We have computed Precision, Recall, MAP and NDCG on the top-10 recommendations. Results are averaged over 10 runs with different random splits.

5.3 Sampling strategies

We formally present here the sampling strategies that we use to produce the intervened test sets in our experiments. Each strategy samples an intervened test set O^S from O^{he} (and the corresponding ratings from Y^{he}, i.e. Y^S). For each strategy we give the corresponding probability sampling distribution, i.e. $P_S(\mathcal{S}|u,i)$. In addition to SKEW, WTD and WTD_H, we also employ two baselines. REG is a random sample from O^{he}, corresponding to an intervention that does not try to compensate for bias. FULL represents the classical test set generation in the evaluation, where the test set is O^{he} (therefore no intervention).

- **FULL**: $P_S(\mathcal{S}|u,i) = 1$ so that O^{he} is fully sampled and no intervention is performed.
- **REG:** $P_S(\mathcal{S}|u,i) = 1/|O^{he}|$. Every (u,i) has a constant probability to be sampled and we obtain a test set that is a random subset of O^{he}. We would expect this to behave very similarly to FULL.
- **SKEW:** $P_S(\mathcal{S}|u,i) = 1/pop(i)$, where $pop(i)$ counts the number of ratings that item i has in O^{tr} [3, 27].
- **WTD**, **WTD_H**: $P_S(\mathcal{S}|u,i) = w_u(w_i)^2$. These are the two alternatives of our approach, presented in Sections 4.2 and 4.3. Weights are calculated using formulas 13 and 14. WTD uses formulas 1 and 2 to calculate the actual MAR posteriors from O^w. WTD_H uses the hypothesized MAR posteriors instead. They both use formulas 4 and 5 to calculate exact MNAR posteriors from O^{tr}.

Note that, in each of SKEW, WTD and WTD_H, if the distribution P_S does not sum to 1 (necessary for a probability distribution), we include a normalization step on P_S to ensure that this property is achieved.

5.4 Recommender systems

We train five recommender models, all of them producing a ranked list of recommended items. AvgRating and PosPop are non-personalized recommenders which rank items in descending order of their mean rating and number of positive ratings in the training set, respectively. UB_KNN and IB_KNN are user-based and item-based nearest-neighbour algorithms [8]. MF is the Matrix Factorization algorithm proposed by Pilaszy and Tikk [20]. For UB_KNN, IB_KNN and MF we use the implementations available in the RankSys library[4]. We used our own implementations of AvgRating and PosPop.

The UB_KNN, IB_KNN and MF algorithms have hyperparameters. We select hyperparameter values that maximize Recall for top-10 recommendations on Y^{val} (Section 5.2). For UB_KNN, IB_KNN, we choose the number of neighbors from $\{10, 20, .., 100\}$. For MF, we choose the number of latent factors from $\{20, 40, .., 200\}$ and the regularization term from $\{0.001, 0.006, 0.01, 0.06, 0.1, 0.6\}$.

[4] https://github.com/RankSys

6 RESULTS

We report the results of our experiments in Table 2. For each recommender, we show its ground-truth Recall@10 performance on the unbiased test set Y^{gt} and the relative performance (in terms of percentage difference) for the baselines and intervened test sets with respect to this ground-truth. Results for Precision, NDCG and MAP are omitted because the percentage differences are very similar to the Recall ones.

Results for CoatShopping show that the baselines and all intervened test sets overestimate ground-truth performances for all recommenders with just one exception: PopPos on WTD_H. In general, our new approaches are superior in approximating ground-truth performances. WTD is very close for non-personalized recommenders performances, while WTD_H is the best for the personalized ones. Although both of them outperform all the other strategies, WTD_H would probably be the best choice due to its 'balance', i.e. its percentage differences are not more than around 50% from the ground-truth for all the recommenders except MF, which anyway has the best approximation on WTD_H among all the strategies.

Results on Webscope R3 show something slightly different. First of all, for the AvgRating recommender, ground-truth performances are underestimated by all strategies. For this recommender, SKEW, WTD and WTD_H are equally good, but superior to FULL and REG anyway. We then find SKEW superior to WTD and WTD_H for the PosPop recommender. But WTD and WTD_H are better for the personalized recommenders. This fact is expected to some extent because SKEW is a popularity-bias specific intervention strategy. Comparing only WTD and WTD_H, we find that both are close to each other, but we also find that the former more closely approximates the ground truth for PosPop, UB_KNN and IB_KNN, while the latter does it for MF and AvgRating (but slightly in this case).

Finally, FULL and REG are very far from the ground-truth, showing that 'intelligent' intervention strategies (such as SKEW, WTD and WTD_H) provide an effective debiasing technique in offline evaluations. Indeed, FULL and REG have very similar results, regardless of the fact that REG is 50% smaller in size. This means that what matters is the strategy that performs the sampling, rather than the sampling itself.

Table 3 reports an additional investigation on the results of Table 2. An offline evaluation typically ranks recommender algorithms from best to worst. This helps to narrow the number of different recommender algorithms that needs to be evaluated in costly user trials and online experiments. In our case then, it is important that performance estimates on intervened test sets, not only get close to the ground truth performance, but also rank different recommenders in the same way they would be ranked by performance estimates on the unbiased test set. We use Kendall's concordance coefficient (τ) to compare the ground truth recommender ranking obtained on the unbiased test set with the ones produced by the different interventions.

Table 2: Recall@10 results for CoatShopping and Webscope R3. We report ground truth performances on test set Y^{gt} in terms of Recall@10. We show the percentage difference of performances on the baselines and the intervened test sets with respect to Y^{gt}.

	CoatShopping						Webscope R3					
Recommender	Y^{gt}	FULL	REG	SKEW	WTD	WTD_H	Y^{gt}	FULL	REG	SKEW	WTD	WTD_H
PosPop	0.066	+133%	+124%	+13%	**+1%**	-43%	0.056	+280%	+272%	**-14%**	-68%	-79%
AvgRating	0.068	+61%	+53%	+31%	**+6%**	+24%	0.016	-77%	-77%	-50%	-51%	**-49%**
UB_KNN	0.067	+229%	+225%	+112%	+90%	**+34%**	0.073	+273%	+265%	+35%	**-9%**	-24%
IB_KNN	0.073	+236%	+227%	+105%	+82%	**+26%**	0.071	+313%	+306%	+45%	**-2%**	-17%
MF	0.063	+180%	+176%	+179%	+123%	**+102%**	0.077	+258%	+252%	+90%	+66%	**+52%**

Table 3: Kendall's concordance coefficient τ for CoatShopping and Webscope R3. The closer to 1, the better the ranking of the recommenders on an intervened test set approximates the ground truth ranking.

	FULL	REG	SKEW	WTD	WTD_H
CoatShopping	**0.2**	**0.2**	0	0	0
Webscope	0.6	0.6	**0.8**	**0.8**	**0.8**

Table 4: Kendall's concordance coefficient τ for CoatShopping and Webscope R3. Differently from Table 3, here we exclude MF when calculating the coefficient.

	FULL	REG	SKEW	WTD	WTD_H
CoatShopping	0.3	0.3	**0.7**	**0.7**	**0.7**
Webscope	**0.68**	**0.68**	**0.68**	**0.68**	**0.68**

The τ values on CoatShopping are far from the maximum possible value (i.e. $\tau = 1$). Also, in this case, 'intelligent' intervention seems to harm the concordance coefficient: SKEW, WTD and WTD_H have lower values ($\tau = 0$) than FULL and REG ($\tau = 0.2$). For Webscope, the τ values are much closer to 1. Also, the 'intelligent' intervention strategies improve the τ values ($\tau = 0.8$) over the baseline ones ($\tau = 0.6$). The concordance coefficients for CoatShopping seem to advise against using 'intelligent' intervention approaches such as SKEW, WTD or WTD_H. However, we note that τ values are subject to great variability, depending on the set of recommenders being compared. In fact, simply dropping the MF model from the comparison, we get very different τ values; see Table 4. Now τ values for Webscope are all the same ($\tau = 0.68$). But we have a completely different scenario for CoatShopping: SKEW, WTD and WTD_H improve concordance (from $\tau = 0$ to $\tau = 0.7$) and they outperform FULL and REG (which slightly improve from $\tau = 0.2$ to $\tau = 0.3$). Low τ values for CoatShopping in Table 3 are a consequence of the fact that all test sets incorrectly rank MF to be one of the best-performing models, while it is the worst according to the ground truth.

7 CONCLUSIONS

In this paper, we presented new sampling strategies that generate intervened test sets with MAR-like properties from MNAR data. These intervened test sets are therefore more suitable for approximating the performance of a recommender on unbiased test data. One of the sampling strategies, WTD, requires that some MAR-like data be available since it approximates posterior probabilities calculated from that data. The other strategy, WTD_H, approximates the probabilities that we expect MAR data to exhibit.

The paper assesses the effectiveness of these two strategies and it assesses, for the first time, the effectiveness of an existing intervention strategy from the literature, namely SKEW, which samples in inverse proportion to item popularity. With the use of an essentially unbiased test set as a ground-truth, we showed these three sampling approaches to be successful in mitigating the biases found in a classical random test set. We found SKEW to be particularly good at reducing the bias for a popularity-based recommender (which is related to the popularity bias of the items for which SKEW was designed). But our new strategies are the most robust across various recommenders since they most closely approximate the unbiased ground-truth performances. The WTD strategy requires MAR data, which is rarely available, but we found that WTD_H, which uses a hypothesized MAR distribution, does work well, so MAR data is not necessary.

Our approach brings several intrinsic benefits. First of all, it enjoys low overheads.

- Its design is simple and easy to implement and it does not require any learning phase for the weights, contrary to some unbiased estimators which might require expensive learning (e.g. [22], where propensities are found via logistic regression).
- Moreover, intervention reduces the computational costs of testing a recommender because it generates smaller test sets.

Another advantage of our approach is that it has high generality.

- It works for both implicit and explicit datasets because it is independent of the interaction values (e.g. ratings) in the dataset.
- Despite the fact that WTD and WTD_H are very close to SKEW, our way of calculating weights is less heuristic than the one of SKEW and, unlike SKEW, it is not tailored to item popularity bias.
- It can be extended to training a recommender, without any modification. Training a recommender on an intervened training set instead of on a classical biased training set, might improve the recommender's model and therefore boost prediction or ranking performances. For this reason, at the time of writing we are investigating using our approach to debias training sets to complement this work on debiasing test sets.

- Intervened data can be used to train *existing* recommender systems and to test recommender systems using *existing* metrics. Debiased training and testing hence become widely applicable without designing special models and special metrics.

Apart from the use of our approach for training a recommender, our aim for the future is to investigate other ways of calculating the weights for the sampling. An alternative might be using techniques developed for causal inference, e.g. [1, 6, 7].

ACKNOWLEDGMENTS

This paper emanates from research supported by a grant from Science Foundation Ireland (SFI) under Grant Number 12/RC/2289-P2, which is co-funded under the European Regional Development Fund.

REFERENCES

[1] Peter C. Austin and Elizabeth A. Stuart. 2015. Moving towards best practice when using inverse probability of treatment weighting (IPTW) using the propensity score to estimate causal treatment effects in observational studies. *Statistics in Medicine* 34, 28 (2015), 3661–3679.

[2] Alejandro Bellogín, Pablo Castells, and Iván Cantador. 2017. Statistical biases in Information Retrieval metrics for recommender systems. *Information Retrieval Journal* 20, 6 (01 Dec 2017), 606–634.

[3] Stephen Bonner and Flavian Vasile. 2018. Causal Embeddings for Recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, New York, NY, USA, 104–112.

[4] Rocío Cañamares and Pablo Castells. 2018. Should I Follow the Crowd?: A Probabilistic Analysis of the Effectiveness of Popularity in Recommender Systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 415–424.

[5] Allison J. B. Chaney, Brandon M. Stewart, and Barbara E. Engelhardt. 2018. How Algorithmic Confounding in Recommendation Systems Increases Homogeneity and Decreases Utility. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, New York, NY, USA, 224–232.

[6] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. 2010. Learning Bounds for Importance Weighting. In *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta (Eds.). Curran Associates, Inc., 442–450.

[7] Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. 2008. Sample Selection Bias Correction Theory. In *Algorithmic Learning Theory*, Yoav Freund, László Györfi, György Turán, and Thomas Zeugmann (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 38–53.

[8] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 39–46.

[9] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic Matrix Factorization with Non-random Missing Data. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32 (ICML'14)*. JMLR.org, II–1512–II–1520.

[10] Guido W. Imbens and Donald B. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction.* Cambridge University Press.

[11] Thorsten Joachims and Adith Swaminathan. 2016. SIGIR Tutorial on Counterfactual Evaluation and Learning for Search, Recommendation and Ad Placement. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (proceedings of the 39th international acm sigir conference on research and development in information retrieval ed.). ACM, 1199–1201.

[12] Yong-Deok Kim and Seungjin Choi. 2014. Bayesian Binomial Mixture Model for Collaborative Prediction with Non-random Missing Data. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys '14)*. ACM, New York, NY, USA, 201–208.

[13] Dawen Liang, Laurent Charlin, and David M. Blei. 2016. Causal Inference for Recommendation. In *UAI Workshop on Causation*.

[14] Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. 2016. Modeling User Exposure in Recommendation. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 951–961.

[15] Daryl Lim, Julian McAuley, and Gert Lanckriet. 2015. Top-N Recommendation with Missing Implicit Feedback. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. ACM, New York, NY, USA, 309–312.

[16] Roderick J A Little and Donald B Rubin. 1986. *Statistical Analysis with Missing Data*. John Wiley & Sons, Inc., New York, NY, USA.

[17] Benjamin M. Marlin and Richard S. Zemel. 2009. Collaborative Prediction and Ranking with Non-random Missing Data. In *Third ACM Conference on Recommender Systems*. 5–12.

[18] Benjamin M. Marlin, Richard S. Zemel, Sam Roweis, and Malcolm Slaney. 2007. Collaborative Filtering and the Missing at Random Assumption. In *Twenty-Third Conference on Uncertainty in Artificial Intelligence*. 267–275.

[19] Arnaud De Myttenaere, Bénédicte Le Grand, Boris Golden, and Fabrice Rossi. 2014. Reducing Offline Evaluation Bias in Recommendation Systems. *CoRR* abs/1407.0822 (2014).

[20] István Pilászy, Dávid Zibriczky, and Domonkos Tikk. 2010. Fast Als-based Matrix Factorization for Explicit and Implicit Feedback Datasets. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 71–78.

[21] Bruno Pradel, Nicolas Usunier, and Patrick Gallinari. 2012. Ranking with Non-random Missing Ratings: Influence of Popularity and Positivity on Evaluation Metrics. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12)*. ACM, New York, NY, USA, 147–154.

[22] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. *CoRR* abs/1602.05352 (2016).

[23] Harald Steck. 2010. Training and Testing of Recommender Systems on Data Missing Not at Random. In *16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 713–722.

[24] Harald Steck. 2011. Item Popularity and Recommendation Accuracy. In *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11)*. ACM, New York, NY, USA, 125–132.

[25] Harald Steck. 2013. Evaluation of Recommendations: Rating-prediction and Ranking. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. ACM, New York, NY, USA, 213–220.

[26] Adith Swaminathan and Thorsten Joachims. 2015. Counterfactual Risk Minimization: Learning from Logged Bandit Feedback. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, Lille, France, 814–823.

[27] Yixin Wang, Dawen Liang, Laurent Charlin, and David M. Blei. 2018. The Deconfounded Recommender: A Causal Inference Approach to Recommendation. *CoRR* abs/1808.06581 (2018).

[28] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased Offline Recommender Evaluation for Missing-not-at-random Implicit Feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, New York, NY, USA, 279–287.