Causal Inference for Recommendation

Dawen Liang
Columbia University
dliang@ee.columbia.edu

Laurent Charlin
HEC Montréal
laurent.charlin@hec.ca

David M. Blei
Columbia University
david.blei@columbia.edu

Abstract

We develop a causal inference approach to recommender systems. Observational recommendation data contains two sources of information: which items each user decided to look at and which of those items each user liked. We assume these two types of information come from different models—the *exposure* data comes from a model by which users discover items to consider; the *click* data comes from a model by which users decide which items they like. Traditionally, recommender systems use the click data alone (or ratings data) to infer the user preferences. But this inference is biased by the exposure data, i.e., that users do not consider each item independently at random. We use causal inference to correct for this bias. On real-world data, we demonstrate that causal inference for recommender systems leads to improved generalization to new data.

1 Introduction

The goal of recommender systems is to infer users' preferences for items and then to predict items that users will like. We develop a causal inference approach to this problem.

Here is the idea. Observational recommendation data contains two sources of information: which items each user decided to look at and which of those items each user liked. For example, one of the data sets we analyze contains which movies each user watched and which of them each liked; another contains which scientific abstracts each user saw and which PDFs each decided to download.

We assume these two types of information come from different models—the *exposure* data comes from a model by which users discover items to consider; the *click* data comes from a model by which users decide which items they like. Traditionally, recommender systems use the click data alone (or ratings data) to infer the user preferences. But this inference is biased by the exposure data, i.e., that users do not consider each item independently at random.

We use causal inference to correct for this bias. First, we estimate the exposure model from the exposure data, a model of which items each user is likely to consider. Then we fit the preferences with weighted click data, where each click (or skip) is weighted by the inverse probability of exposure (from the exposure model). This is a propensity weighting approach to causal inference [5], i.e., we warp the observational click data as though it came from an "experiment" where users are randomly shown items. We study several variants of this strategy.

Why might this work? Consider the film enthusiast (from our data) who mostly watches popular drama but has also enjoyed a couple of documentaries ("Crumb" and "The Cruise"). A classical recommender system will infer film preferences that center around drama. Our causal method detects a preference for drama too, but further up-weights the preference for documentaries. The reason is that the history of the user indicates that she is unlikely to have been exposed to many documentaries; the method values its signal from the two she did like. Consequently, when we recommend from among the unwatched films, our method promotes documentaries ("Fast, Cheap & Out of Control" and "Paris Is Burning") that the user (in held-out data) also liked. Across users, on real-world data, we demonstrate that causal inference for recommender systems leads to improved generalization to new data.

Related work. Marlin and Zemel [11] first formalized statistical models for correcting bias in observational recommendation data. They posit that a user's decision to rate an item depends on the user's opinion of the item.

They propose a mechanism to correct for this rating-selection bias, based first on generating a rating and then conditionally on whether the rating is observed. Others have proposed different rating models using this same mechanism [10, 3]. In contrast, our model (similar to Liang et al. [9]) first generates each user's exposure to an item and then her rating. Unlike [9], we work with explicit click data. Thus we can use causal inference to debias the resulting inference of user preferences.

Solving recommendations using causality has been explored in the multi-arm bandit literature (e.g., [8, 16, 17, 7]). They focus on unbiased evaluation of a recommendation policy using biased data (e.g., data collected in web log). This work typically uses importance sampling, weighting the probability of each observed click under the logging policy and under the (new) recommendation policy. We use the same tools for data re-weighting—propensity score weighting is equivalent to importance sampling—but we reason about learning preferences rather than evaluating recommendation policies. Further we work in a batch learning setting (as opposed to online learning).

The recent work of Schnabel et al. [15] is closest to ours. The authors propose a causal inference approach to learning unbiased estimators from biased rating data. One important difference with our work is that their propensity weights depend on user preferences (either directly through ratings or indirectly through user and item covariates)—a process known as self-selection—rather than exposure, as in our work. Their formalization of the problem also differs: they appeal to empirical risk minimization while we take a Bayesian perspective.

2 A causal model for recommendation

In this section we develop our method. We describe explicit recommendation data, a joint model of exposure and clicks, how we do prediction, and how we do causal inference.

Data. Our data are *explicit data*: we know which items each user saw and which of those items each clicked (liked) or skipped (disliked). For example, in Section 4.4 we analyze a large collection of click data from arXiv.org.[1] We know which arXiv abstracts a user has viewed and, among those, which PDFs she has downloaded. Our goal is to infer each user's latent preferences for items and then to use those preferences in a recommender system.

[1] http://arxiv.org is a pre-print repository for scientific papers.

We begin with notation for the data. A user is indexed by u and an item is indexed by i. There are two types of observations. The *exposure data* is a_{ui}, whether user u had the opportunity to click on item i. (We use the verb "click" for concreteness; this can be any type of interaction, including "download," "purchase," "listen," or "watch.") The *click data* is y_{ui}, an indicator of whether user u clicked on (liked) item i or decided to skip (disliked) the item.

These data capture the users' clicks. There are some items which a user was exposed to ($a_{ui} = 1$) but did not click on ($y_{ui} = 0$); there are other items that a user was exposed to ($a_{ui} = 1$) and did click on ($y_{ui} = 1$); finally, there are items that a user was not exposed to ($a_{ui} = 0$) and, by definition, did not click on ($y_{ui} = 0$). A user cannot click on an item she is not exposed to.

Joint models of exposure and clicks. We build a joint model of this data: an exposure model of what the user sees and a click model of what the user clicks on, conditional on her seeing it. The key idea behind our approach is this. Given observational data, i.e., data collected by users exploring information and clicking on items, classical inference of the click model—of the user's preferences for clicking on items that she is exposed to—is incorrect because of the biases induced by the exposure model. We take a causal inference approach to this problem: we infer the user's preferences from an imagined experiment where each item is exposed with equal probability.

We first describe the *observation joint*, from which we observe our data set.

$$\begin{aligned} a_{ui} &\sim f(\cdot \mid \pi_{ui}) \\ y_{ui} \mid a_{ui} = 0 &\sim \delta_0(\cdot) \\ y_{ui} \mid a_{ui} = 1 &\sim g(\cdot \mid \mu_{ui}). \end{aligned}$$

Here the exposure and click models are generic. Each is governed by the exposure parameter π_{ui} and click parameter μ_{ui}, respectively.

For example, one exposure model we study is a Bernoulli with an item-specific parameter. We call this the popularity exposure because it allows some items to be more likely to be exposed (across users) than others,

$$a_{ui} \sim \text{Bernoulli}(\rho_i). \tag{1}$$

Alternatively, the exposure model can capture a user's preferences for seeking out items. We will also study Poisson factorization,

$$a_{ui} \sim \text{Poisson}(\pi_u^\top \gamma_i), \tag{2}$$

which finds non-negative embeddings for users and items [2].[2]

For the click model, we use classical probabilistic matrix factorization [14]. Conditional on being exposed, the click comes from a normal distribution, $y_{ui} \mid a_{ui} = 1 \sim \mathcal{N}(\theta_u^\top \beta_i, \lambda_y^{-1})$. Here θ_u is a latent K-vector of user preferences and β_i is a latent K-vector of item attributes. In all models, the conditional distribution of a click y_{ui} given that a user is not exposed to the item ($a_{ui} = 0$) is a point mass at zero.

Forming predictions. Our goal is to use this model to form future predictions about the users. We are given observed data $\mathcal{D} = \{(a_{ui}, y_{ui})\}$ of what each user was exposed to and what each user clicked on. We want to predict what we should expose them to in the future, i.e., what they would like to see.

We will study two ways of predicting. One is to form conditional predictions as the probability that a user clicks on an item given that she is exposed to it,

$$\mathbb{E}\left[y_{ui} \mid a_{ui} = 1, \mathcal{D}\right]. \tag{3}$$

Alternatively, we use marginal predictions, where we marginalize out the exposure variable

$$\mathbb{E}\left[y_{ui} \mid \mathcal{D}\right] = \mathbb{P}(a_{ui} = 1 \mid \pi_{ui}, \mathcal{D})\mathbb{E}\left[y_{ui} \mid a_{ui} = 1, \mathcal{D}\right]. \tag{4}$$

The marginal prediction uses that $y_{ui} = 0$ when $a_{ui} = 0$. It is apt when the exposure model also contains information about the user, i.e., information about what the user is likely to seek out.

Note that these methods require approximating the posterior predictive distribution of y_{ui} and a_{ui} given the data. We now turn to this inference problem.

Causal inference for recommendation. One way to solve the inference problem is with classical Bayesian inference, where we condition on the observed data and then use posterior prediction to recommend items. But there is an issue with using classical Bayesian inference to form recommendations: the data we observe $\mathcal{D}^{\text{obs}}$ is not the data from which we would like to infer the user's preferences and item attributes, i.e., the click model. The reason is that the exposure model—the distribution that governs what each user sees—biases our inference about the click model. Items that users are likely to be exposed exert too much of an influence; items that users are rarely exposed to have too little influence.

Ideally, we would infer preferences from an experiment, a model that randomly exposed each user to items and then recorded which items each one click on. We call this the *intervention joint*,

$$\begin{aligned} a_{ui} &\sim \text{Bernoulli}(\pi) \\ y_{ui} \mid a_{ui} = 0 &\sim \delta_0(\cdot) \\ y_{ui} \mid a_{ui} = 1 &\sim g(\cdot \mid \mu_{ui}). \end{aligned}$$

In this model, we have intervened on the mechanism from which users are exposed to items. (This is the "mutilated model" [13].) Data from this model leads to better estimates of the click model (i.e., their preferences) and better generalization to the items that they will want to click on.

This is a causal approach to the problem. The observation joint is the model of how we collected the data; the intervention joint is a model of a randomized experiment that would (in theory) help us make the inferences that we need. The challenge is to use data from the observation joint to perform inference in the intervention joint.

How do we solve this problem? Assume for now that the exposure model is known and is the popularity model, i.e., $a_{ui} \sim \text{Bernoulli}(\rho_i)$. We will use *inverse propensity weighting* [5], which takes samples from the observation joint and weights them to look like samples from the intervention joint; this is essentially an importance sampling technique. Specifically, we weight each observation (a_{ui}, y_{ui}) by $1/\rho_i$ to estimate θ_u. (Because of the click model, this estimate only relies on those data where $a_{ui} = 1$.) When inferring a user's preferences, this down-weights the influence of popular items and up-weights the influence of unpopular items.

More formally, our goal is to obtain a data set $\mathcal{D} = \{(a_{ui}, y_{ui})\}$ from the intervention joint and then estimate $p(\theta_u \mid \mathcal{D})$. We define the "do dataset" to be the observed data embellished with weights, $\mathcal{D}^{\text{do}} = \{(a_{ui}, y_{ui}, w_{ui})\}$, where w_{ui} is the inverse probability of exposure. The posterior is

$$p(\theta_u \mid \mathcal{D}^{\text{do}}) \propto p(\theta_u) \prod_i p(y_{ui} \mid a_{ui}, \theta_u, \beta_i)^{w_{ui}} \tag{5}$$

Intuitively, this assumes that we see each data point "w_{ui} times", and that the clicks are conditionally independent given the preferences.

How is this different from standard causality? One way is that, in typical causal settings, we have a single causal question [5]. Here we have many causal questions (one per user-item pair). What is crucial is that the causal outcomes are related, each governed by the same set of parameters.

[2] Though Poisson models capture count data, they are effective for binary data with many items [2].

3 The algorithm

We first estimate the exposure model from the observed data. This can be the popularity model or Poisson factorization. Then, we use the fitted exposure model to weight the data (by the inverse probability) and fit the click model. Finally we use the posterior distribution of the exposure model and (causal) posterior distribution of the click model to form predictions. This procedure generalizes better than Bayesian inference, especially under intervention, i.e., when we change the distribution of which items a user is exposed to.

3.1 Fitting the exposure model

Popularity model. For popularity exposure model $a_{ui} \sim \text{Bernoulli}(\rho_i)$ (Eq. 1), we obtain the maximum likelihood estimate $\hat{\rho}_i$ by counting the portion of the users who have been exposed to item i. The propensity score p_{ui} in this case is fixed across users:

$$p_{ui} = \hat{\rho}_i, \forall u \in \{1, \ldots, U\}. \tag{6}$$

Poisson factorization model. For Poisson factorization exposure model $a_{ui} \sim \text{Pois}(\pi_u^\top \gamma_i)$ (Eq. 2) with conjugae gamma prior on the latent embeddings π_u and γ_i, we perform standard variational inference [2] on the exposure data a_{ui}. After obtaining the optimal approximating variational distribution q on π_u and γ_i at convergence, we compute the propensity score,

$$p_{ui} = 1 - \mathbb{P}(a_{ui} = 0 \mid \pi_{ui}, \mathcal{D}) \approx 1 - \exp\{-\mathbb{E}_q\left[\pi_u^\top \gamma_i\right]\}. \tag{7}$$

3.2 Fitting the click model

The click model (conditional on exposure) is a matrix factorization $y_{ui} \mid a_{ui} = 1 \sim \mathcal{N}(\theta_u^\top \beta_i, \lambda_y^{-1})$. We place a diagonal normal prior on both user preference $\theta_u \sim \mathcal{N}(0, \lambda_\theta^{-1} \mathrm{I}_K)$ and item attributes $\beta_i \sim \mathcal{N}(0, \lambda_\beta^{-1} \mathrm{I}_K)$.

To fit the model, we compute the maximum *a posteriori* estimates of the parameters θ_u and β_i. This leads to efficient closed-form coordinates updates, which scale to large datasets. Concretely, the objective for the inverse propensity weighted matrix factorization model (Eq. 5) is (without loss of generality, we set $\lambda_y = 1$, as we can always scale λ_θ and λ_β by λ_y to obtain the same solution):

$$\mathcal{L} = \sum_{(u,i)\in\mathcal{O}} \frac{1}{p_{ui}} (y_{ui} - \theta_u^\top \beta_i)^2 + \lambda_\theta \sum_u \|\theta_u\|_2^2 + \lambda_\beta \sum_i \|\beta_i\|_2^2,$$

where the propensity score p_{ui} can be obtained by either Eq. 6 or Eq. 7. The observed set $\mathcal{O}$ contains all the entries with $a_{ui} = 1$. We can obtain the following coordinate updates by taking the gradients with respect to θ_u and β_i and setting them to 0:

$$\theta_u^{\text{new}} \leftarrow \Big(\sum_{i:(u,i)\in\mathcal{O}} \frac{1}{p_{ui}} \beta_i \beta_i^\top + \lambda_\theta \mathrm{I}_K \Big)^{-1} \Big(\sum_{i:(u,i)\in\mathcal{O}} \frac{1}{p_{ui}} y_{ui} \beta_i \Big) \tag{8}$$

$$\beta_i^{\text{new}} \leftarrow \Big(\sum_{u:(u,i)\in\mathcal{O}} \frac{1}{p_{ui}} \theta_u \theta_u^\top + \lambda_\beta \mathrm{I}_K \Big)^{-1} \Big(\sum_{u:(u,i)\in\mathcal{O}} \frac{1}{p_{ui}} y_{ui} \theta_u \Big) \tag{9}$$

The full algorithm for the inverse propensity weighted matrix factorization is summarized in Algorithm 1. Note that this algorithm only includes options for fitting the model causally (Eq. 5). In Section 4, we empirically explore different combinations of the exposure model, prediction method, and fitting procedure.

Algorithm 1: IPW-MF Coordinate updates for inverse propensity weighted matrix factorization

Input: Exposed entires in the click matrix $\{y_{ui} : (u,i) \in \mathcal{O}\}$, regularization parameters λ_θ and λ_β
Output: User latent factors $\theta_{1:U}$ and item latent factors $\beta_{1:I}$

```
Fit the exposure model to compute the propensity score (Eq. 6 or Eq. 7)
Randomly initialize θ_{1:U}, β_{1:I}
while not converged do
    for u ← 1 to U do
        Update user factor θ_u (Eq. 8)
    end
    for i ← 1 to I do
        Update item factor β_i (Eq. 9)
    end
end
return θ_{1:U}, β_{1:I}
```

4 Empirical study

We studied causal recommender systems on several data sets. We compared models trained observationally with models trained causally; we compared predictions made marginally and those made conditional on exposure; we studied and evaluated different exposure models, both those based on popularity and based on personalized preferences; and we studied typical test sets and test sets that focus on rare items.

We highlight the following results:

	ML-1M	ML-10M	Yahoo-R3	ArXiv
# of users	6,040	69,878	15,400	26,541
# of items	3,706	10,677	1,000	80,082
# of exposures	1.0M	10.0M	0.3M	1.9M
% of exposures	4.47%	1.34%	2.02%	0.09%

Table 1: Attributes of the data. # of exposures is the number of entries with $a_{ui} = 1$ (rated an item, viewed an abstract). % of exposure refers to the density of the user-item exposure matrix.

- Poisson factorization (Eq. 2) is a better exposure model than the one based on item popularity (Eq. 1). We evaluate the exposure model both as a standalone model to predict held-out exposure and as a component in the whole recommender system.
- When the test set focuses on rare items, fitting causally (Eq. 5) gives better generalization than classical inference. Causal inference is important for generalizing to situations that we do not see in training.
- Accounting for exposure is important when making prediction—recommendation with marginal prediction (Eq. 4) significantly boosts the ranking-based recommendation performance.

We give details below. We describe the data, methods, metrics, and results.

4.1 Data

We study three types of data (and four data sets):

- *MovieLens (ML-1M and ML-10M).* User-movie ratings collected from a movie recommendation service.[3] The ratings are on a 1–5 scale.
- *Yahoo-R3.* Music ratings collected from Yahoo! Music services [11]. The ratings are 1–5.
- *ArXiv.* User-paper clicks from the 2012 log-data of the arXiv pre-print server. The data are binarized: multiple clicks by the same user on the same paper are considered to be a single click. This data contains which papers a user downloaded and which she only read the abstract.

For ML-1M, ML-10M, and Yahoo-R3, we denote exposure $a_{ui} = 1$ as user u having rated item i. These three datasets are typically used for rating prediction. Because our end goal is recommendation, we binarize the ratings and encode preferences as being either positive or negative ($y_{ui} = 1$ if rating is greater than or equal to 3 and $y_{ui} = 0$ otherwise). This type of binarization gives better recommendation performance than directly using predicted ratings [4].[4]

In ArXiv we denote exposure $a_{ui} = 1$ as user u having viewed the abstract of paper i. Among papers that a user is exposed to, we set $y_{ui} = 1$ if she downloaded the paper and $y_{ui} = 0$ otherwise.

Table 1 summarizes the important attributes of our four datasets.

Data pre-processing. For each dataset, we create two training/validation/test splits: regular (REG) and skewed (SKEW). We create a regular split by randomly selecting the exposed items for each user into training/validation/test sets, following 70/10/20 proportions. In the regular split, the test set has the same exposure distribution as the training and validation sets. This is how researchers typically evaluate recommendation models (with observational data).

The skewed split rebalances the splits to better approximate an intervention. We create it by first sampling a test set with roughly 20% of the total exposures, such that each item has uniform probability. Training and validation sets are then created from the remaining data (as in a regular split) with 70/10 proportions. For a skewed split, the test set will have a completely different exposure distribution from the training and validation sets. We use this split to demonstrate that causal inference for recommendation leads to improved generalization performance.

Figure 1 shows the scatter plots of the training exposure distribution (reflected by the empirical item popularity) against the test exposure distribution on regular and skewed splits of the ML-1M dataset. The empirical item popularity is computed by counting the number of users who have been exposed to each item. The skewed split has a roughly uniform exposure distribution across items, while in the regular split, both training and test sets follow similar exposure patterns.

[3] http://grouplens.org/datasets/movielens/

[4] We note that the Yahoo! data set also contains a random test set, where a subset of the users are given 10 randomly selected songs to rate. But most of the ratings for this random test set are below 3. Rather, we created a skewed test set.

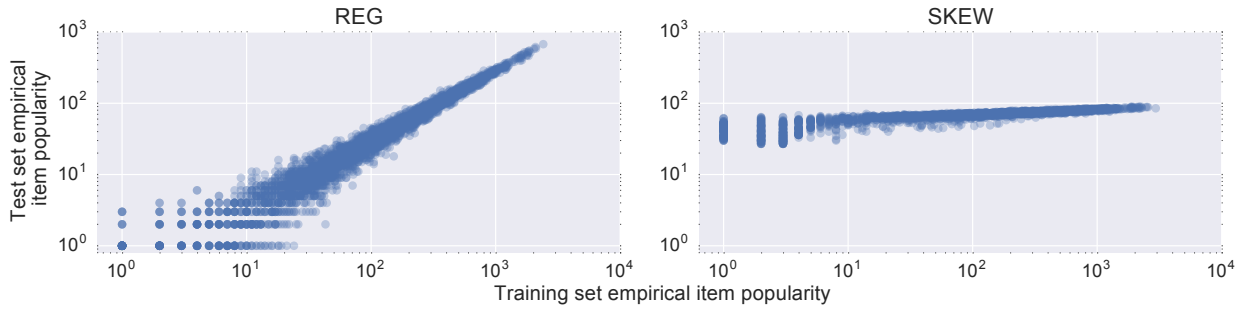

Figure 1: Scatter plots of the training exposure distribution (reflected by the empirical item popularity) against the test exposure distribution on REG (left) and SKEW (right) splits for ML-1M dataset. SKEW split has a roughly uniform exposure distribution across items, while in REG split both training and test sets follow similar exposure patterns.

4.2 Methods

There are several choices in the proposed method. We explored combinations of the exposure model, prediction method, and fitting procedure. The different choices are summarized below:

- *Exposure model.* Popularity (Pop, Eq. 1) or Poisson factorization (PF, Eq. 2).
- *Prediction.* Conditional prediction (Cond, Eq. 3) or marginal prediction (Mar, Eq. 4).
- *Model fitting.* Train the click model causally (CAU, Eq. 5), with inverse propensity weighting, or observationally (OBS), with classical inference.

Among these methods are two baselines. The models that are trained observationally (OBS) with conditional prediction (Cond) correspond to classical matrix factorization [14]. The models that are trained causally (CAU) with conditional prediction (Cond) correspond to inverse propensity weighted matrix factorization proposed in Schnabel et al. [15].[5] We note that this approach significantly outperformed the previous state-of-the-art model proposed in Hernández-Lobato et al. [3] for the task of rating prediction (though the main focus of this paper is on recommendation).

Hyperparameters. We perform grid search using $\lambda_\theta \in \{10^{-4}, 10^{-3}, \ldots, 10^4\}$ and $\lambda_\beta \in \{10^{-4}, 10^{-3}, \ldots, 10^4\}$ to select hyperparameters based on the normalized discounted cumulative gain (NDCG) [6] of the validation set.

We set the dimension of the latent space K to 30 and use the same random initialization of θ_u and β_i in all settings. For the scalable coordinate updates in Algorithm 1, we declare convergence when the mean squared error on the validation set increases.

4.3 Metrics

We separately evaluate the exposure model, how well we predict which items a user will see, and the click model, which items a user will like. Note that causal inference of the click model uses the exposure model to compute the propensity score. Further, marginal prediction of clicks also uses the exposure model.

We evaluate the exposure model using model fitness to the data (predictive log-likelihood). We evaluate the click model with recommendation metrics, both a likelihood-based metric (a tail probability) and a ranking-based metric (mean average rank [1]).[6] We describe the recommendation metrics in turn.

Predictive log tail probability (PLP). For y_{ui} in the heldout test set, we compute the predictive log-probability based on its value and whether we predict conditionally or marginally (see Eq. 4).

Conditional prediction uses $\mathbb{E}[y_{ui} \mid a_{ui} = 1, \mathcal{D}]$. If $y_{ui} = 1$, we compute right-tail conditional predictive log-probability for positively preferred items,

$$\log \mathbb{P}(y_{ui}^{\text{pred}} > 1 \mid a_{ui} = 1, \mathcal{D}).$$

Otherwise we compute left-tail conditional predictive log-probability

$$\log \mathbb{P}(y_{ui}^{\text{pred}} \leq 0 \mid a_{ui} = 1, \mathcal{D}).$$

Both correspond to Gaussian tail probability for matrix factorization.

[5] Even though Schnabel et al. [15] derive the model from empirical risk minimization framework, the model objective closely resembles the joint log-likelihood of the causally trained model (CAU) with conditional prediction (Cond).

[6] NDCG [6] is another commonly used ranking-based metric. It emphasizes the importance of the top ranks by logarithmically discounting ranks. MAR, on the other hand, makes no such discounting.

Marginal prediction uses $\mathbb{E}[y_{ui} \mid \mathcal{D}]$. Recall p_{ui} is the probability that user u is exposed to item i from the exposure model. If $y_{ui} = 1$, we compute right-tail marginal predictive log-probability,

$$\begin{aligned}&\log \mathbb{P}(y_{ui}^{\text{pred}} > 1 \mid \mathcal{D})\\ &= \log p_{ui} + \log \mathbb{P}(y^{\text{pred}} > 1 | a_{ui} = 1, \mathcal{D}).\end{aligned}$$

Otherwise we compute left-tail marginal predictive log-probability

$$\begin{aligned}&\log \mathbb{P}(y_{ui}^{\text{pred}} \leq 0 \mid \mathcal{D})\\ &= \log \left(p_{ui}\, \mathbb{P}(y_{ui}^{\text{pred}} \leq 0 \mid a_{ui} = 1, \mathcal{D}) + (1 - p_{ui})\right).\end{aligned}$$

The intuition behind PLP is that we would like to have 0's and 1's in the heldout set well-separated. This is different from the commonly used metrics for rating prediction, e.g., mean squared error or mean absolute error, both of which penalize the model unless it predicts with a perfect 0 and 1. We report average PLP over all the heldout y_{ui} in the test set.

Mean Average Rank. We compute MAR as follows. For user u we calculate the ranking of all the items $i \in \{1, 2, \ldots, I\}$ by sorting the predictions and excluding the items from the training and validation sets. Define $\text{rank}(u, i)$ as the predicted rank of item i for user u: $\text{rank}(u, i) = 0$ if item i is ranked first for user u and $\text{rank}(u, i) = I - 1$ if ranked last. For items within a set I_u,

$$\text{MAR}_u = \frac{1}{|I_u|} \sum_{i \in I_u} \text{rank}(u, i).$$

In our studies, I_u is the item set in the heldout set with $y_{ui} = 1$, i.e., the items that user u rated positively or the papers that user u downloaded after looking at the abstract. Since the value of MAR depends on the size of the item set I, we report the normalized MAR percentile instead as MAR_u / I. This also corresponds to the expected percentile ranking proposed in Hu et al. [4] with binary feedback data. The interpretation of MAR is on average at what percentile a heldout item will be ranked (smaller is better). The reported MAR averages over all users.

4.4 Results

We report our studies on all data. We evaluated both the exposure model alone and the click model, which uses the exposure model to improve its recommendations.

Evaluating the exposure model. We first compare two different exposure models used in this paper: Poisson factorization (PF) and the popularity model (Pop). We use the training set created in Section 4.1 to train the model (for PF, we use the validation set to monitor convergence). We randomly sample the same number of entries with $a_{ui} = 0$ as those with $a_{ui} = 1$ and report the average heldout predictive log-likelihood in Table 2.

PF always outperforms Pop. Further, the predictive log-likelihood is always lower on skewed test sets than on regular test sets. This is expected because skewed test sets follow a different exposure distribution from the training and validation sets. This makes it harder for the exposure model to correctly predict its values.

Evaluating the click model. We summarize the log probability (PLP) and mean average rank (MAR) (described in Section 4.3) in Table 3a and Table 3b, respectively. The table reports eight different model configurations based on which exposure model is used, how the model is fit, and how predictions are formed.[7]

From Table 3, we make the following observations.

1. Poisson factorization (PF) gives better performance in terms of both PLP and MAR than the popularity exposure model (Pop). (Pop configurations are on the top half of each table; PF configurations are on the bottom half.)

2. If the test set exposure comes from the same distribution as the training set (regular split), training the model observationally or causally does not make a difference in terms of PLP. As for MAR, we can make the same observation with marginal prediction, but ArXiv is an exception.

   On the other hand, if the test set exposure distribution is different (the skewed split), training the model causally gives more robust generalization performance. Even on ArXiv, we can see that moving from regular to skewed severely degrades the performance of observationally-trained models, as opposed to causally-trained models, where the degradation is comparably weaker.

3. Overall, marginal prediction boosts MAR comparing to the conditional counterparts. (PLP is not comparable between conditional and marginal.) When we predict whether a user will like an item, we should consider her preference as well as how likely she is to seek out the item.

4. In Schnabel et al. [15], the authors use a naive Bayes propensity score estimator. Our results show that a

[7] There are seven distinct configurations, as the ones that are trained observationally (OBS) with conditional prediction (Cond) will not depend on the exposure model. We keep all eight configurations in Table 3 for easy comparison.

	ML-1M		ML-10M		Yahoo-R3		ArXiv	
	REG	SKEW	REG	SKEW	REG	SKEW	REG	SKEW
Pop	-1.39	-2.07	-1.64	-2.76	-1.81	-2.74	-3.83	-3.95
PF	-0.97	-1.51	-1.08	-2.06	-1.58	-2.35	-2.71	-2.80

Table 2: Heldout predictive log-likelihood for Poisson factorization (PF) exposure model and popularity exposure model (Pop). PF outperforms Pop across datasets. The predictive log-likelihood is generally lower on SKEW than REG.

			ML-1M		ML-10M		Yahoo-R3		ArXiv	
			REG	SKEW	REG	SKEW	REG	SKEW	REG	SKEW
Pop	Cond	OBS	-1.50	-2.07	-1.62	-2.59	-1.58	-1.75	-1.61	-1.65
		CAU	-1.61	-1.95	-1.67	-1.89	-1.51	-1.56	-1.74	-1.76
	Mar	OBS	-3.17	-4.29	-3.56	-5.63	-2.98	-3.53	-3.93	-4.21
		CAU	-3.21	-4.25	-3.60	-5.24	-2.84	-3.53	-3.94	-4.15
PF	Cond	OBS	-1.50	-2.07	-1.62	-2.59	-1.58	-1.75	-1.61	-1.65
		CAU	-1.48	-1.84	-1.51	-1.96	-1.49	-1.55	-1.60	-1.62
	Mar	OBS	-2.62	-3.87	-2.69	-4.61	-2.71	-3.40	-3.05	-3.32
		CAU	-2.60	-3.57	-2.69	-4.42	-2.59	-3.14	-3.04	-3.33

(a) Predictive log tail probability (bigger is better)

			ML-1M [%]		ML-10M [%]		Yahoo-R3 [%]		ArXiv [%]	
			REG	SKEW	REG	SKEW	REG	SKEW	REG	SKEW
Pop	Cond	OBS	13.0	25.6	5.4	18.3	15.1	36.1	18.4	23.6
		CAU	17.3	27.1	8.0	18.4	21.5	31.7	32.1	35.7
	Mar	OBS	11.8	26.6	5.1	18.9	15.6	36.9	22.5	33.8
		CAU	12.3	26.9	5.3	18.7	15.8	36.6	30.0	42.9
PF	Cond	OBS	13.0	25.6	5.4	18.3	15.1	36.1	18.4	23.6
		CAU	16.9	26.6	7.8	17.1	16.6	29.2	30.7	33.9
	Mar	OBS	6.9	19.1	2.9	14.2	10.2	28.9	7.5	13.0
		CAU	6.9	18.4	3.1	14.2	9.9	25.9	11.2	13.1

(b) Mean average rank (smaller is better)

Table 3: Predictive log tail probability (PLP) and mean average rank (MAR) for the click model on different datasets. The results are organized by the exposure model (Pop or PF), how to fit the model (OBS or CAU), and how to make prediction (Cond or Mar). The Cond-OBS models correspond to the classical matrix factorization [14]. The Cond-CAU models correspond to Schnabel et al. [15]. See main text for detailed analysis.

more flexible propensity model (e.g., Poisson factorization) tends to give better recommendation performance.

5. We notice that the results with causally-trained models (CAU) on ArXiv are less stable than those from the other three datasets. ArXiv is more than one order of magnitude sparser than the other datasets and less popularity-biased—even considering abstract views, most of the papers are only viewed and downloaded by a small number of users. Therefore, the estimated propensity score could contain extreme values, a common problem for methods involving propensity score [12]. As part of the future work, we will investigate different propensity score smoothing techniques.

5 Conclusion

In this paper, we develop a causal inference approach to recommendation with explicit data. We separately model two sources of information: the *exposure* data (which items each user decided to look at) and *click* data (which of those items each user liked). Exposure data introduces bias when we estimate parameters of a recommendation model from the click data, as rare items do not get as much exposure as popular ones. We use inverse propensity weighting to correct for this bias. Through extensive empirical study, we demonstrate that this causal approach to recommender systems leads to improved generalization to new data.

As future work, we can develop similar methodology for implicit data [4]. The main difficulty in implicit data is that we do not know which items a user has been exposed to. The exposure matrix factorization model and its variations developed in Liang et al. [9] which introduce the user exposure as latent variables could be potentially helpful with that.

References

[1] Charlin, L., Ranganath, R., McInerney, J., and Blei, D. M. (2015). Dynamic Poisson factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 155–162.

[2] Gopalan, P., Hofman, J., and Blei, D. (2015). Scalable recommendation with hierarchical Poisson factorization. In *Uncertainty in Artificial Intelligence*.

[3] Hernández-Lobato, J. M., Houlsby, N., and Ghahramani, Z. (2014). Probabilistic matrix factorization with non-random missing data. In *Proceedings of the 31th International Conference on Machine Learning, ICML*, pages 1512–1520.

[4] Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Data Mining, The IEEE International Conference on*, pages 263–272.

[5] Imbens, G. W. and Rubin, D. B. (2015). *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge University Press.

[6] Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

[7] Li, L., Chen, S., Kleban, J., and Gupta, A. (2015). Counterfactual estimation and optimization of click metrics in search engines: A case study. In *Proceedings of the 24th International World Wide Web Conference (WWW'14), Companion Volume*.

[8] Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 661–670, New York, NY, USA.

[9] Liang, D., Charlin, L., McInerney, J., and Blei, D. M. (2016). Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*, pages 951–961.

[10] Ling, G., Yang, H., Lyu, M. R., and King, I. (2012). Response aware model-based collaborative filtering. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 501–510.

[11] Marlin, B. M. and Zemel, R. S. (2009). Collaborative prediction and ranking with non-random missing data. In *Proceedings of the Third ACM Conference on Recommender Systems*, pages 5–12.

[12] Morgan, S. L. and Winship, C. (2014). *Counterfactuals and Causal Inference*. Cambridge University Press.

[13] Pearl, J. (2009). *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition.

[14] Salakhutdinov, R. and Mnih, A. (2008). Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1257–1264.

[15] Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., and Joachims, T. (2016). Recommendations as treatments: Debiasing learning and evaluation. In *International Conference on Machine Learning (ICML)*.

[16] Vanchinathan, H., Nikolic, I., De Bona, F., and Krause, A. (2014). Explore-exploit in top-n recom-

mender systems via Gaussian processes. In *Proc. ACM Recommender Systems Conference (RecSys)*.

[17] Zhao, X., Zhang, W., and Wang, J. (2013). Interactive collaborative filtering. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 1411–1420.