Causal Inference for Recommender Systems

Yixin Wang
Columbia University
yixin.wang@columbia.edu

Dawen Liang
Netflix Inc.
dliang@netflix.com

Laurent Charlin
Mila, HEC Montréal
laurent.charlin@hec.ca

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

ABSTRACT

The task of recommender systems is classically framed as a prediction of users' preferences and users' ratings. However, its spirit is to answer a counterfactual question: "What would the rating be if we 'forced' the user to watch the movie?" This is a question about an intervention, that is a causal inference question. The key challenge of this causal inference is unobserved confounders, variables that affect both which items the users decide to interact with and how they rate them. To this end, we develop an algorithm that leverages classical recommendation models for causal recommendation. Across simulated and real datasets, we demonstrate that the proposed algorithm is more robust to unobserved confounders and improves recommendation.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; *Latent variable models.*

KEYWORDS

recommender systems, causal inference, unobserved confounding

ACM Reference Format:
Yixin Wang, Dawen Liang, Laurent Charlin, and David M. Blei. 2020. Causal Inference for Recommender Systems. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20), September 22–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3383313.3412225

1 INTRODUCTION

The goal of a recommender is to show its users items that they will like. Given a dataset of users' ratings, a recommender system learns the preferences of the users, predicts the users' ratings on those items they did not rate, and finally makes suggestions based on those predictions. In this paper we develop a causal inference approach to recommendation.

Why is recommendation a causal inference? Concretely, suppose the items are movies and the users rate movies they have seen. In prediction, the recommender system is trying to answer "How would the user rate this movie if she saw it?" However, recommending all the movies that users will like may not be the most cost-efficient strategy. Many recommendations, though costing money, will not make a difference in user behaviors. For example, users like certain movies so much that they will go see them no matter whether there is a recommendation. Therefore, we only want to recommend the movies that (1) if *made exposed*, the user will go see them and (2) if not, the user will not go see them. But this is a question about an *intervention*: what would the rating be if we *make* the user exposed (or not exposed) to the movie? One tenet of causal inference is that predictions under intervention are different from usual predictions.

Framing recommendation as a causal problem differs from the traditional approach. The traditional approach builds a model from observed ratings data, often a matrix factorization, and then uses that model to predict unseen ratings. But this strategy only provides valid causal inferences—in the intervention sense above—if users randomly watched movies. (This is akin to a randomized clinical trial, where the treatment is exposure to a movie and the response is a rating.)

Users do not (usually) watch movies at random and, consequently, answering the causal question from observed ratings data is challenging. The issue is that there may be *confounders*, variables that affect both the treatment assignments (which movies the users watch) and the outcomes (how they rate them). For example, because a user watches many movies by a particular director, they may often be recommended movies by this director and also tend to watch and like those movies. The director is a confounder that biases our inferences; it affects both which movies the user were recommended and whether they watch and like them. Compounding this issue, the confounders might be difficult (or impossible) to measure. Further, the theory around causal inferences says that these inferences are valid only if we have accounted for all confounders [19]. And, alas, whether we have indeed measured all confounders is uncheckable [8].

How can we overcome these obstacles? In this paper, we develop the *deconfounded recommender*, a method that tries to correct classical matrix factorization for unobserved confounding. The deconfounded recommender builds on the two sources of information in recommendation data: which movies each user decided to watch and the user's rating for each of those movies. It posits that the two types of information come from different models—the *exposure* data comes from a model by which users discover movies to watch; the *ratings* data comes from a model by which users decide which movies they like. The ratings data entangles both types of

RecSys '20, September 22–26, 2020, Virtual Event, Brazil

ACM ISBN 978-1-4503-7583-2/20/09…$15.00
https://doi.org/10.1145/3383313.3412225

information—users only rate movies that they see—and so classical matrix factorization is biased by the exposure model, i.e., that users are not randomly exposed to movies.

The deconfounded recommender tries to correct this bias. It first uses the exposure data to estimate a model of which movies each user is likely to consider. (In recommender systems, the exposure data is a form of "implicit" data.) It then uses this exposure model to estimate a substitute for the unobserved confounders. Second, it fits a ratings model (e.g., matrix factorization) that accounts for the substitute confounders. The justification for this approach comes from Wang and Blei [26]; correlations among the considered movies provide indirect evidence for confounders.[1]

Consider a film enthusiast who mostly watches western action movies but who has also enjoyed two Korean dramas, even though non-English movies are not easily accessible in her area. A traditional recommender will infer preferences that center around westerns; the dramas carry comparatively little weight. The deconfounded recommender will also detect the preference for westerns, but it will further up-weight the preference for Korean dramas. The reason is that the history of the user indicates that she is unlikely to have been exposed to many non-English movies, and she liked the two Korean dramas that she did see. Compared to westerns, Korean dramas are likely movies that if recommended she might like and if not recommended she might see. Consequently, when recommending from among the unwatched movies, the deconfounded recommender promotes other Korean dramas along with westerns.

Below we develop the deconfounded recommender. We empirically study it on both simulated data, where we control the amount of confounding, and real data, about shopping and movies. Compared to existing approaches, it predicts the ratings better and consistently improves recommendation.

Related work. This work draws on several threads of previous research in recommendation algorithms.

The first is on evaluating recommendation algorithms via biased data. It is mostly explored in the multi-armed bandit literature [11, 12, 25, 27]. These works focus on online learning and rely on importance sampling. Here we consider an orthogonal problem. We reason about user preferences, rather than recommendation algorithms, and we use offline learning and parametric models.

The second thread is around the missing-not-completely-at-random assumption in recommendation algorithms. Marlin and Zemel [16] studied the effect of violating this assumption in ratings. Similar to our exposure model, they posit an explicit missingness model that leads to improvements in predicting ratings. Later, other researchers proposed different rating models to accommodate this violated assumption [1, 7, 13, 14, 23, 24]. In this work, we take an explicitly causal view of the problem. While violating the missing-not-completely-at-random assumption is one form of confounding bias [4], the explicit causal view opens up the door to other recent debiasing tools, such as the deconfounder [26]. It also articulates the rationale of such adjustments: By modeling which movies users tend to watch, we avoid recommending the movies that user will watch anyway even without a recommendation. Rather, we only want to recommend movies that if recommender the user will watch and otherwise not.

Finally, the recent work of Schnabel et al. [22] also adapted causal inference—inverse propensity weighting (IPW), in particular—to address missingness. Their propensity models rely on either observed ratings of a randomized trial or externally observed user and item covariates. In contrast, our work relies solely on the observed ratings: we do not require ratings from a gold-standard randomized exposure nor do we use external covariates. In § 3, we show that the deconfounded recommender provides better recommendations than Schnabel et al. [22].

2 THE DECONFOUNDED RECOMMENDER

We frame recommendation as a causal inference and develop the deconfounded recommender.

Matrix factorization as potential outcomes. We first set up notation. Denote a_{ui} as the indicator of whether user u rated movie i. Let $y_{ui}(1)$ be the rating that user u would give movie i if she watches it. This rating is only observed if the user u watched and rated the movie i; otherwise it is unobserved. Similarly define $y_{ui}(0)$ to be the rating of user u on movie i if she does not see the movie. (We often "observe" $y_{ui}(0) = 0$ in recommendation data; unrated movie entries are filled with zeros.) The pair $(y_{ui}(0), y_{ui}(1))$ is the *potential outcomes* notation in the Rubin causal model [10, 20, 21], where watching a movie is a "treatment" and a user's rating of the movie is an "outcome."

A recommender system observes users' ratings of movies. We can think of these observations as two datasets. One dataset contains (binary) exposures, $\{a_{ui}, u = 1, \ldots, U, i = 1, \ldots, I\}$. It indicates who watched what. The other dataset contains the ratings for movies that users watched, $\{y_{ui}(a_{ui})$ for (u, i) such that $a_{ui} = 1\}$.

The goal of the recommender is to suggest movies its users will like. It first estimates $y_{ui}(1)$ for user-movie pairs with $a_{ui} = 0$; that is, it predicts each user's ratings for their unseen movies. It then uses these estimates to suggest movies to users. Note $y_{ui}(1)$ is a prediction under intervention: "What would the rating be if user u was made to watch movie i?"

To form the prediction of $y_{ui}(1)$, we recast matrix factorization in potential outcomes. First set up an *outcome model*,

$$y_{ui}(a) = \theta_u^\top \beta_i \cdot a + \epsilon_{ui}, \quad \epsilon_{ui} \sim \mathcal{N}(0, \sigma^2). \tag{1}$$

When $a = 1$ (i.e., user u watches movie i), this model says that the rating comes from a Gaussian distribution whose mean combines user preferences θ_u and item attributes β_i. When $a = 0$, the "rating" is a zero-mean Gaussian.

Fitting Eq. 1 to the observed data recovers classical probabilistic matrix factorization [17]. Its log likelihood only involves observed ratings; it ignores the unexposed items. The fitted model can then predict $\mathbb{E}[y_{ui}(1)] = \theta_u^\top \beta_i$ for every (unwatched) user-movie pair. These predictions suggest movies that users would like.

Classical causal inference and adjusting for confounders in recommendation. But matrix factorization does not provide an unbiased causal inference of $y_{ui}(1)$. The theory around potential outcomes says we can only estimate $y_{ui}(1)$ if we assume *ignorability*. For all users u, ignorability requires $\{\boldsymbol{y}_u(0), \boldsymbol{y}_u(1)\} \perp\!\!\!\perp \boldsymbol{a}_u$, where $\boldsymbol{y}_u(a) = (y_{u1}(a), \ldots, y_{uI}(a))$ and $\boldsymbol{a}_u = (a_{u1}, \ldots, a_{uI})$. In words,

[1] The deconfounded recommender focuses on how the exposure of each individual movie (i.e. one of the many causes) causally affects its observed rating (Eq. 6); we rely on Theorem 7 of Wang and Blei [26] for the identification of causal parameters. Note this result does not contradict the causal non-identification examples given in D'Amour [2], which operate under different assumptions.

the vector of movies a user watches $\boldsymbol{a}_u$ is independent of how she would rate them if she watched them all $\boldsymbol{y}_u(1)$ (and if she watched none $\boldsymbol{y}_u(0)$).

Ignorability clearly does not hold for $\boldsymbol{y}_u(1)$—the process by which users find movies is not independent of how they rate them. Practically, this violation biases the estimates of user preferences θ_u: movies that u is not likely to see are downweighted and vice versa. Again consider the American user who enjoyed two Korean dramas and rated them highly. Because she has only two high ratings of Korean dramas in the data, her preference for Korean dramas carries less weight than her other ratings; it is biased downward. Biased estimates of preferences lead to biased predictions of ratings.

When ignorability does not hold, classical causal inference asks us to measure and control for confounders [18, 21]. These are variables that affect both the exposure and the ratings. Consider the location of a user as an example. It affects both which movies they are exposed to and (perhaps) what kinds of movies they like.

Suppose we measured these per-user confounders w_u; they satisfy $\{\boldsymbol{y}_u(0), \boldsymbol{y}_u(1)\} \perp\!\!\!\perp \boldsymbol{a}_u \mid w_u$. Classical causal inference controls for them in the outcome model, $y_{ui}(a) = \theta_u^\top \beta_i \cdot a + \eta^\top w_u + \epsilon_{ui}, \epsilon_{ui} \sim \mathcal{N}(0, \sigma^2)$. However, this solution requires we measure *all* confounders. This assumption is known as *strong ignorability*.[2] Unfortunately, it is untestable [8].

The deconfounded recommender. We now develop the deconfounded recommender. It leverages the dependencies among the exposure ("which movies the users watch") as indirect evidence for unobserved confounders. It uses a model of the exposure to construct a substitute confounder; it then conditions on the substitute when modeling the ratings.

The key idea is that causal inference for recommendation systems is a *multiple causal inference* problem: there are multiple treatments. Each user's binary exposure to each movie a_{ui} is a treatment; thus there are I treatments for each user. The vector of ratings $\boldsymbol{y}_u(1)$ is the outcome; this is an I-vector, which is partially observed. The multiplicity of treatments enables causal inference with unobserved confounders [26].

The first step is to fit a model to the exposure data. We use Poisson factorization (PF) model [5]. PF assumes the data come from the following process,

$$a_{ui} \mid \pi_u, \lambda_i \sim \text{Poisson}(\pi_u^\top \lambda_i), \qquad \forall u, i, \tag{2}$$

where both $\pi_u \overset{iid}{\sim} \text{Gamma}(c_1, c_2)$ and $\lambda_i \overset{iid}{\sim} \text{Gamma}(c_3, c_4)$ are nonnegative K-vectors. The user factor π_u captures user preferences (in picking what movies to watch) and the item vector λ_i captures item attributes. PF is a scalable variant of nonnegative factorization and is especially suited to binary data [5]. It is fit with coordinate ascent variational inference.[3]

With a fitted PF model, the deconfounded recommender computes a substitute for unobserved confounders. It reconstructs the exposure matrix $\hat{a}$ from the PF fit,

$$\hat{a}_{ui} = \mathbb{E}_{\text{PF}}[\pi_u^\top \lambda_i \mid \boldsymbol{a}], \tag{3}$$

where $\boldsymbol{a}$ is the observed exposure for all users, and the expectation is taken over the posteriors computed from the PF model. This is the posterior predictive mean of $\pi_u^\top \lambda_i$, which serves as a substitute confounder [26].

Finally, the deconfounded recommender posits an outcome model conditional on the substitute confounders $\hat{a}$,

$$y_{ui}(a) = \theta_u^\top \beta_i \cdot a + \gamma_u \cdot \hat{a}_{ui} + \epsilon_{ui}, \quad \epsilon_{ui} \sim \mathcal{N}(0, \sigma^2), \tag{4}$$

where γ_u is a user-specific coefficient that describes how much the substitute confounder $\hat{a}$ contributes to the ratings. The deconfounded recommender fits this outcome model to the observed data; it infers $\theta_u, \beta_i, \gamma_u$, via MAP estimation. The coefficients θ_u, β_i in Eq. 4 are fit only with the observed user ratings (i.e., $a_{ui} = 1$) because $a_{ui} = 0$ zeroes out the term that involves them; in contrast, the coefficient γ_u is fit to all movies ($a_{ui} = 0$ and $a_{ui} = 1$) since $\hat{a}_{ui}$ is always non-zero.

To form recommendations, the deconfounded recommender calculates all the potential ratings $y_{ui}(1)$ with the fitted $\hat{\theta}_u, \hat{\beta}_i, \hat{\gamma}_u$. It then orders the potential ratings of the unseen movies. These are causal recommendations. Algorithm 1 provides the algorithm for forming recommendations with the deconfounded recommender.

Why does it work? Poisson factorization (PF) learns a per-user latent variable π_u from the exposure matrix a_{ui}, and we take π_u as a substitute confounder. What justifies this approach is that PF admits a special conditional independence structure: conditional on π_u, the treatments a_{ui} are independent (Eq. 2). If the exposure model PF fits the data well, then the per-user latent variable π_u (or functions of it, like $\hat{a}_{ui}$) captures multi-treatment confounders, i.e., variables that correlate with multiple exposures and the ratings vector (Lemma 3 of [26]). We note that the true confounding mechanism does not need to coincide with PF and nor does the real confounder need to coincide with π_u. Rather, PF produces a substitute confounder that is sufficient to debias confounding.

Beyond probabilistic matrix factorization. The deconfounder involves two models, one for exposure and one for outcome. We have introduced PF as the exposure model and probabilistic matrix factorization [17] as the outcome model. Focusing on PF as the exposure model, we extend the deconfounded recommender to general outcome models.

We start with a general form of matrix factorization,

$$y_{ui}(a) \sim p(\cdot \mid m(\theta_u^\top \beta_i, a), v(\theta_u^\top \beta_i, a)), \tag{5}$$

where $m(\theta_u^\top \beta_i, a)$ characterizes the mean and $v(\theta_u^\top \beta_i, a)$ the variance of the ratings $y_{ui}(a)$. This form encompasses many factorization models, including probabilistic [17], weighted [9], and Poisson matrix factorization [5]. The deconfounded recommender then fits an augmented outcome model M_Y. This outcome model M_Y includes the substitute confounder,

$$y_{ui}(a) \sim p(\cdot \mid m(\theta_u^\top \beta_i, a) + \gamma_u \hat{a}_{ui} + \beta_0, v(\theta_u^\top \beta_i, a)). \tag{6}$$

Notice the parameter γ_u is a user-specific coefficient; for each user, it characterizes how much the substitute confounder $\hat{a}$ contributes to the ratings. Note the deconfounded recommender also includes an intercept β_0. These deconfounded outcome model can be fit by *maximum a posteriori* estimation.

[2] In causal graphical models, this requirement is equivalent to "no open backdoor paths" [18].

[3] The Bernoulli distribution is more natural to model binary exposure, but PF is more computationally efficient and several precedents use a Poisson to model binary data [5, 6]. PF scales linearly with the number of *nonzero* entries in the exposure matrix $\{a_{ui}\}_{U \times I}$ while Bernoulli scales with the number of *all* entries. Further, the Poisson distribution closely approximates the Bernoulli when the exposure matrix $\{a_{ui}\}_{U \times I}$ is sparse [3]. Finally, PF can also model non-binary *count* exposures: e.g., PF can model exposures that count how many times a user has been exposed to an item.

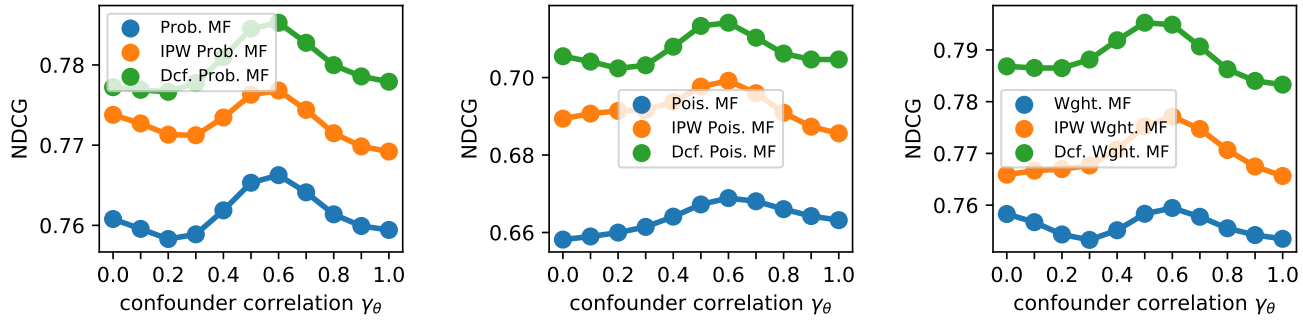

Figure 1: (Left) Probabilistic MF (Center) Poisson MF (Right) Weighted MF. Varying confounder correlation γ_θ from 0.0 to 1.0 ($\gamma_y = 3.0$). The recommendation performance of the deconfounded recommender (green) is better than its classical counterpart (blue) and the existing causal approach, IPW MF [22] (orange). (Higher is better.)

Algorithm 1 The Deconfounded Recommender

Input: a dataset of exposures and ratings $\{(a_{ui}, y_{ui}(a_{ui}))\}_{u,i},\ i = 1, \ldots, I, u = 1, \ldots, U$

Output: the potential outcome given treatment $\hat{y}_{ui}(1)$

1. Fit PF to the exposures $\{a_{ui}\}_{u,i}$ from Eq. 2
2. Compute substitute confounders $\{\hat{a}_{ui}\}_{u,i}$ from Eq. 3
3. Fit the outcome model $\{(a_{ui}, y_{ui}(a_{ui}))\}_{u,i}$ from Eq. 6
4. Estimate all potential ratings $y_{ui}(1)$ with the fitted outcome model (Eq. 6)

3 EMPIRICAL STUDIES

We study the deconfounded recommender on simulated and real datasets. We examine its recommendation performance and compare to existing recommendation algorithms. We find that the deconfounded recommender predicts the ratings better and consistently improves recommendation. (The supplement contains software that reproduces these studies.)

3.1 Evaluation of causal recommendation models

We first describe how we evaluate the recommender. Traditionally, we evaluate the accuracy (e.g. mean squared error or ranking metrics) of the predicted ratings. However, causal recommendation models pose unique challenges for evaluation. In causal inference, we need to evaluate how a model performs across *all* potential outcomes, $\text{err}_{\text{cau}} = \frac{1}{U}\sum_{u=1}^{U} \ell(\{\hat{y}_{ui}\}_{i\in\{1,\ldots,I\}}, \{y_{ui}(1)\}_{i\in\{1,\ldots,I\}})$, where ℓ is a loss function, such as mean squared error (MSE) or normalized discounted cumulative gain (NDCG). The challenge is that we don't observe all potential outcomes $y_{ui}(1)$. If we use a "regular test set" by randomly splitting the data, it gives a biased estimate of err_{cau}; it emphasizes popular items and active users. The (expensive) solution is to measure a randomized test set. Randomly select a subset $\mathcal{I}_u$ from all items and ask the users to interact and rate all of them. Then compute the average loss across users, $\text{err}_{\text{rand}} = \frac{1}{U}\sum_{u=1}^{U} \ell(\{\hat{y}_{ui}\}_{i\in\mathcal{I}_u}, \{y_{ui}(1))\}_{i\in\mathcal{I}_u})$, which is an unbiased estimate of the average across all items in err_{cau}; it tests the recommender's ability to answer the causal question. Two available datasets that include such random test sets are the Yahoo! R3 dataset [16] and the coat shopping dataset [22]. We also create random test sets in simulation studies.

3.2 Simulation studies

We study the deconfounded recommender on simulated datasets. We simulate movie ratings for $U = 1,000$ users and $I = 1,000$ items, where effect of preferences on rating is confounded.

Simulation setup. We simulate a K-vector confounder for each user $c_u \sim \text{Gamma}_K(0.3, 0.5)$ and a K-vector of attributes for each item $\beta_i \sim \text{Gamma}_K(0.3, 0.5)$. We then simulate the user preference K-vectors θ_u conditional on the confounders, $\theta_u \sim \gamma_\theta \cdot c_u + (1-\gamma_\theta)\cdot \text{Gamma}_K(0.3, 0.5)$. The constant $\gamma_\theta \in [0, 1]$ controls the exposure-confounder correlation; higher values imply stronger confounding.

We next simulate the binary exposures $a_{ui} \in \{0, 1\}$, the ratings for all users watching all movies $y_{ui}(1) \in \{1, 2, 3, 4, 5\}$, and calculate the observed ratings y_{ui}. The exposures and ratings are both simulated from truncated Poisson distributions; the exposures are from $a_{ui} \sim \min(\text{Poisson}(c_u^\top \beta_i), 1)$ and the ratings are from $y_{ui}(1) \sim \min(1 + \text{Poisson}((\theta_u + \gamma_y \cdot c_u)^\top \beta_i), 5)$. Finally, the observed ratings mask the ratings by the exposure, $y_{ui} = a_{ui} \cdot y_{ui}(1)$. The constant $\gamma_y \geq 0$ controls how much the confounder c_u affects the outcome; higher values imply stronger confounding.

Competing methods. We compare the deconfounded recommender to baseline methods. One set of baselines are the classical counterparts of the deconfounded recommender. We explore probabilistic matrix factorization [17], Poisson matrix factorization [5], and weighted matrix factorization [9]. We additionally compare to inverse propensity weighting (IPW) matrix factorization [22], which also handles selection bias in observational recommendation data.

Table 1: Recommendation on random test sets for existing users (weak generalization) and new users (strong generalization). The deconfounded recommender improves recommendation over classical approaches and the existing causal approach [22]. (Higher is better.)

	Existing users (Weak generalization)				New users (Strong generalization)			
	Yahoo! R3		Coat		Yahoo! R3		Coat	
	NDCG	Recall@5	NDCG	Recall@5	NDCG	Recall@5	NDCG	Recall@5
Prob. MF [17]	0.811	0.671	0.719	0.451	0.811	0.831	0.779	0.650
IPW Prob. MF [22]	0.813	0.676	0.714	0.480	0.814	**0.834**	0.781	0.548
Dcf. Prob. MF (ours)	**0.815**	**0.680**	**0.721**	**0.487**	**0.816**	0.793	**0.805**	**0.760**
Pois. MF [5]	0.773	0.510	0.680	0.358	0.767	**0.771**	0.725	0.608
IPW Pois. MF [22]	0.783	0.550	0.692	0.362	0.772	0.735	0.717	0.601
Dcf. Pois. MF (ours)	**0.788**	**0.565**	**0.696**	**0.387**	**0.782**	0.741	**0.739**	**0.618**
Wght. MF [9]	0.821	0.641	0.718	0.554	0.802	0.776	**0.799**	**0.617**
IPW Wght. MF [22]	0.822	0.644	0.717	0.449	0.800	0.772	0.770	0.600
Dcf. Wght. MF (ours)	**0.823**	**0.648**	**0.720**	**0.566**	**0.810**	**0.782**	0.766	0.594

Results. Figure 1 shows the recommendation performance of different algorithms. The deconfounded recommender leads to higher NDCGs in recommendation than its classical counterparts (Probabilistic, Poisson, or Weighted Matrix Factorization) and the existing causal approach (Inverse Propensity Weighted (IPW) Probabilistic, Poisson, or Weighted Matrix Factorization [22]).

3.3 Case studies: The deconfounded recommender on random test sets

We next study the deconfounded recommender on two real datasets: Yahoo! R3 [16] and coat shopping [22].[4] Both datasets are comprised of an observational training set and a random test set. The training set comes from users rating user-selected items; the random test set comes from the recommender system asking its users to rate randomly selected items. The latter enables us to evaluate how different recommendation models predict *potential outcomes*: what would the rating be if we *make* a user watch and rate a movie?

Evaluation metrics. We use the recommenders for two types of prediction: weak generalization and strong generalization [15]. Weak generalization predicts preferences of existing users in the training set on their unseen movies. Strong generalization predicts preferences of new users—users not in the training set—on their unseen movies. Based on the predictions, we rank the items with nonzero ratings. For evaluation, we report NDCG and recall.

Results. Table 1 show the performance of the deconfounded recommender and its competitors. Across the three metrics and the two datasets, the deconfounded recommender outperforms its classical counterpart for both weak and strong generalization: it produces better item rankings and improves retrieval quality; its predicted ratings are also more accurate. The deconfounded recommender also outperforms the IPW matrix factorization [22], which is the main existing approach that targets selection bias in recommendation systems. These results show that the deconfounded recommender produces more accurate predictions of user preferences.

[4]Yahoo! R3 [16] contains user-song ratings. The training set contains over 300K user-selected ratings from 15400 users on 1000 items. Its random test set contains 5400 users who were asked to rate 10 randomly chosen songs. The coat shopping dataset [22] contains user-coat ratings. The training set contains 290 users. Each user supplies 24 user-selected ratings among 300 items. Its random test contains ratings for 16 randomly selected coat per user.

Acknowledgments. This work is supported by ONR N00014-17-1-2131, ONR N00014-15-1-2209, NIH 1U01MH115727-01, NSF CCF-1740833, DARPA SD2 FA8750-18-C-0130, Amazon, NVIDIA, Simons Foundation, CIFAR AI chair program, NSERC, FRQNT, IVADO, Samsung, and Google.

REFERENCES

[1] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 104–112.
[2] Alexander D'Amour. 2019. On multi-cause approaches to causal inference with unobserved counfounding: Two cautionary failure cases and a promising alternative. In *The 22nd International Conference on Artificial Intelligence and Statistics*. 3478–3486.
[3] M.H. DeGroot and M.J. Schervish. 2012. *Probability and Statistics*. Addison-Wesley. https://books.google.com/books?id=4TlEPgAACAAJ
[4] Peng Ding, Fan Li, et al. 2018. Causal inference: A missing data perspective. *Statist. Sci.* 33, 2 (2018), 214–237.
[5] Prem Gopalan, Jake M Hofman, and David M Blei. 2015. Scalable Recommendation with Hierarchical Poisson Factorization.. In *UAI*. 326–335.
[6] Prem K Gopalan, Laurent Charlin, and David Blei. 2014. Content-based recommendations with Poisson factorization. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger (Eds.). Curran Associates, Inc., 3176–3184. http://papers.nips.cc/paper/5360-content-based-recommendations-with-poisson-factorization.pdf
[7] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic matrix factorization with non-random missing data. In *International Conference on Machine Learning*. 1512–1520.
[8] Paul W Holland, Clark Glymour, and Clive Granger. 1985. Statistics and causal inference. *ETS Research Report Series* 1985, 2 (1985).
[9] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 263–272.
[10] Guido W Imbens and Donald B Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.
[11] Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. 2015. Counterfactual estimation and optimization of click metrics in search engines: A case study. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 929–934.
[12] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 661–670.
[13] 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*. International World Wide Web Conferences Steering Committee, 951–961.
[14] Guang Ling, Haiqin Yang, Michael R Lyu, and Irwin King. 2012. Response aware model-based collaborative filtering. *arXiv preprint arXiv:1210.4869* (2012).
[15] Benjamin M Marlin. 2004. Modeling user rating profiles for collaborative filtering. In *Advances in neural information processing systems*. 627–634.
[16] Benjamin M Marlin and Richard S Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems*. ACM, 5–12.

[17] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.
[18] Judea Pearl. 2009. *Causality*. Cambridge University Press.
[19] Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70, 1 (1983), 41–55.
[20] Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology* 66, 5 (1974), 688.
[21] Donald B Rubin. 2005. Causal inference using potential outcomes: Design, modeling, decisions. *J. Amer. Statist. Assoc.* 100, 469 (2005), 322–331.
[22] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. *arXiv preprint arXiv:1602.05352* (2016).
[23] Andrew Smith and Charles Elkan. 2004. A Bayesian network framework for reject inference. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 286–295.
[24] Nathan Srebro and Ruslan R Salakhutdinov. 2010. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *Advances in Neural Information Processing Systems*. 2056–2064.
[25] Hastagiri P Vanchinathan, Isidor Nikolic, Fabio De Bona, and Andreas Krause. 2014. Explore-exploit in top-n recommender systems via gaussian processes. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 225–232.
[26] Yixin Wang and David M Blei. 2018. The blessings of multiple causes. *arXiv preprint arXiv:1805.06826* (2018).
[27] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive collaborative filtering. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 1411–1420.