The Deconfounded Recommender: A Causal Inference Approach to Recommendation

Yixin Wang Columbia University
Dawen Liang Netflix Inc.
Laurent Charlin HEC Montréal
David M. Blei Columbia University

Abstract

The goal of a recommender system is to show its users items that they will like. In forming its prediction, the recommender system tries to answer: "what would the rating be if we 'forced' the user to watch the movie?" This is a question about an intervention in the world, a causal question, and so traditional recommender systems are doing causal inference from observational data. This paper develops a causal inference approach to recommendation. Traditional recommenders are likely biased by unobserved confounders, variables that affect both the "treatment assignments" (which movies the users watch) and the "outcomes" (how they rate them). We develop the *deconfounded recommender*, a strategy to leverage classical recommendation models for causal predictions. The deconfounded recommender uses Poisson factorization on which movies users watched to infer latent confounders in the data; it then augments common recommendation models to correct for potential confounding bias. The deconfounded recommender improves recommendation and it enjoys stable performance against interventions on test sets.

1 Introduction

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

Why is recommendation a causal inference problem? For concreteness, suppose the items are movies and the users are rating movies they have seen. In prediction, the recommender system is trying to answer the question "How would the user rate this movie if he or she saw it?" In the language of causality, this is a question about an intervention: what would the rating be if we "forced" the user to watch the movie? The item plays the role of the treatment; the rating plays the role of the response.

How does this framing of recommendation, as a causal inference problem, differ 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. This strategy would provide valid causal inferences—in the sense described above—if users randomly watched and rated movies. (This would be akin to a randomized clinical trial, where pills are replaced by movies.)

But 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, a user's social network might affect both the movies she is exposed to and how much she likes those movies.

Compounding this issue, the confounders might be difficult (or impossible) to measure and observe. Further, the theory around causal inferences says that these inferences are valid only if we have

Preprint. Work in progress.

accounted for all confounders [17]. And, alas, whether we have indeed measured all confounders is an uncheckable assumption [5].

How can we overcome these obstacles? In this paper, we build on the recent ideas of [22] to develop the *deconfounded recommender*. Here is the idea. The recommendation data contains two sources of information: which movies each user decided to watch and the user's rating for each of those movies. We assume these 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. This separation clarifies how the classical inferences from a matrix factorization are biased—they are biased by the exposure model, i.e., that users do not randomly choose movies.

The deconfounded recommender tries to correct this bias. First, it uses the exposure data to estimate a model of which movies each user is likely to consider. (In the language of recommender systems, the exposure data is a form of "implicit" data.) Then, it uses this exposure model to estimate a substitute for the unobserved confounders. That this is possible is the result of Wang and Blei [22]; correlations among considered movies provide indirect evidence for confounders. Finally, it fits a ratings model (e.g., matrix factorization) that corrects for the substitute confounders.

Why might this strategy work? Consider the film enthusiast (from our dataset) who mostly watches popular drama ("Enchanted April") but has also enjoyed a couple documentaries ("Crumb" and "The Cruise"). A traditional recommender system will infer preferences that center around drama. Deconfounded recommendation will also detect a preference for drama, but will further up-weight 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 it recommends from among the unwatched films, the deconfounded recommendation 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, the deconfounded recommender provides better recommendations.

In the next sections, we cast recommendation as causal inference, develop the deconfounded recommender, and demonstrate its improvements on recommendation and its robustness to interventions.

Related work. The first body of related work is on evaluating recommendation policy via biased data. It is mostly explored in the multi-armed bandit literature [9, 23, 21, 10]. 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 policies, and we use offline learning and parametric models.

The second body of related work is around the missing-not-completely-at-random assumption in recommendation methods. Marlin and Zemel [14] 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 [11, 3, 12, 20]. In contrast to these works, we take an explicitly causal view of the problem. While missingness and causality are deeply connected through the concept of ignorability [1], our causal view opens up the door to new debiasing tools (e.g. [22]).

Finally, the recent work of Schnabel et al. [20] adapted causal inference—inverse propensity weighting, in particular—to address missingness. Their propensity models rely on either observed ratings of a missing-completely-at-random sample 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 and nor do we use external covariates.

2 Recommender system as causal inference

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

2.1 A potential outcomes perspective

A recommender system observes users rating items. We can think of this system as generating two datasets. One dataset contains exposures a_{ui}, indicators of whether user u rated item i. The other

dataset contains outcomes $y_{ui}(a_{ui})$, which are the observed ratings when $a_{ui} = 1$ (i.e., if the user saw the movie) and 0 if $a_{ui} = 0$ (i.e., if the user did not see the movie). This is *potential outcomes* notation [7, 18, 19]. For every user/movie pair, there is a variable $y_{ui}(0)$ and the variable $y_{ui}(1)$.

As we said, the goal of a recommender system is to recommend movies its users will like. Given the dataset $\{a_{ui}, y_{ui}(a_{ui})\}$, this goal amounts to estimating the ratings had all movies been seen by all users; we want to estimate $y_{ui}(1)$ for all u and i. Again note the estimate of $y_{ui}(1)$ is a prediction under intervention. It answers the question "What would the rating be if user u was forced to see movie i?" The challenge to this estimation is that we only observe $y_{ui}(1)$ when $a_{ui} = 1$. The process by which users select movies to rate can bias estimates of $y_{ui}(1)$ for the unexposed items [17].

Traditional recommender systems fit matrix factorization to observed ratings. Continuing in the potential outcomes notation, we posit probabilistic matrix factorization as an *outcome model* [15],

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

We then fit the user preferences θ_u and item attributes β_i to the observed exposure and ratings. Mathematically, this procedure ignores the unexposed items (because $a = 0$). We emphasize that it restates matrix factorization for recommendation [8] from the perspective of potential outcomes.

This approach would correctly estimate potential outcomes if we could assume *ignorability*. Ignorability means that the outcome (a user's rating of a movie, observed or not) is independent of the treatment assignment (whether the user has watched the movie). Formally, for all users u, $\{y_u(0), y_u(1)\} \perp\!\!\!\perp \boldsymbol{a}_u$ where $y_u(a) = (y_{u1}(a), \ldots, y_{uI}(a))$ and $\boldsymbol{a}_u = (a_{u1}, \ldots, a_{uI})$. (The system has I items.) While $y_{ui}(0) = 0$ and thus is ignorable, $y_u(1) \perp\!\!\!\perp a_u$ is often not true—the process by which users find movies is not independent of how they rate them.

2.2 The deconfounded recommender

How can we build a recommendation system when ignorability does not hold? We appeal to causal inference from observational data.

Classical causal inference. When ignorability does not hold, classical causal inference asks us to measure and control for confounders. These are variables that affect both the exposure and the ratings. Consider the social network of a user. It may affect both the movies she is exposed to and her ratings about those movies.

Suppose we measured these per-user confounders w_u. Classical inference controls for them in the outcome model,

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

However, this solution requires we measure *all* confounders. This assumption is known as *strong ignorability*. Unfortunately, it is untestable [5].

The deconfounded recommender. We now develop the deconfounded recommender. It captures dependencies among the exposure data to infer substitutes for the unobserved confounders. It then models the ratings conditional on the substitute confounders.

The first step is to fit a Poisson factorization (PF) model [2] to the exposure data. PF assumes the data come from the following process,

$$U_u \overset{iid}{\sim} \text{Gamma}(c_1, c_2), V_i \overset{iid}{\sim} \text{Gamma}(c_3, c_4), \tag{3}$$

$$a_{ui} \mid U_u, V_i \sim \text{Poisson}(U_u^\top V_i), u = 1, \ldots, U, i = 1, \ldots, I, \tag{4}$$

where both U_u and V_i are nonnegative K-vectors. The user factor U_u captures user preferences (in picking what movies to watch) and the item vector V_i captures item attributes. PF is a scalable variant of nonnegative factorization and is especially suited to binary data [2]. It is fit with coordinate ascent variational inference, which scales with the number of nonzeros in $\boldsymbol{a} = \{a_{ui}\}_{u,i}$.

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,

$$\hat{a}_{ui} = \mathbb{E}_{\text{PF}}[U_u^\top V_i \mid \boldsymbol{A} = \boldsymbol{a}], \tag{5}$$

where $\boldsymbol{A}$ is the exposure random variable, $\boldsymbol{a}$ is the observed exposure, and the expectation is taken over the posteriors computed from the PF model. This is the posterior predictive mean of $U_u^\top V_i$. The reconstructed exposure $\hat{a}_{ij}$ serves as a substitute confounder.

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

$$y_{ui}(a) = \theta_u^\top \beta_i \cdot a + \gamma_u \cdot \hat{a}_{ui} + \epsilon_{ui}, \quad \epsilon_{ui} \sim \mathcal{N}(0, \sigma^2), \tag{6}$$

where γ_u is a user-specific coefficient that describes how much the substitute confounder $\hat{a}$ contributes to the ratings. The recommender fits this outcome model to the observed data, inferring $\theta_u, \beta_i, \gamma_u$, and then calculates all the potential ratings $y_{ui}(1)$ with the fitted $\hat{\theta}_u, \hat{\beta}_i, \hat{\gamma}_u$. To form recommendations for a user, it orders the potential ratings of the unseen movies. These are causal recommendations. Note in fitting Equation (6), the coefficients θ_u, β_i 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 (both a_{ui} = 0 and a_{ui} = 1) because $\hat{a}_{ui}$ is always non-zero.

Beyond probabilistic matrix factorization. The deconfounder involves two models, one for exposure and one for outcome. We focus on PF as the exposure model but consider other outcome models.

We start with a general form of matrix factorization,

$$y_{ui}(a) \sim p(\cdot \mid m(\theta_u^\top \beta_i, a), v(\theta_u^\top \beta_i, a)), \tag{7}$$

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. Probabilistic matrix factorization [15] is

$$m(\theta_u^\top \beta_i, a) = a \cdot \theta_u^\top \beta_i, \quad v(\theta_u^\top \beta_i, a) = \sigma^2.$$

Weighted matrix factorization [6] is

$$m(\theta_u^\top \beta_i, a) = \theta_u^\top \beta_i, \quad v(\theta_u^\top \beta_i, a) = \sigma_a^2,$$

where $\sigma_0^2 = \alpha \sigma_1^2$. This model leads us to downweight the zeros; we are less confident about the zero ratings. Poisson matrix factorization [2] only takes in a mean parameter

$$m(\theta_u^\top \beta_i, a) = \theta_u^\top \beta_i.$$

With the general matrix factorization of Equation (7), the deconfounded recommender fits an augmented outcome model M_Y. This outcome model M_Y includes the substitute confounder as a covariate,

$$y_{ui}(a) \sim p(\cdot \mid m(\theta_u^\top \beta_i, a) + \gamma_u \hat{a}_{ui} + \beta_0, v(\theta_u^\top \beta_i, a)). \tag{8}$$

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.

The deconfounded outcome model can be fit by *maximum a posteriori* estimation. It solves

$$\begin{aligned}\hat{\theta}_u, \hat{\beta}_i, \hat{\gamma}_u, \hat{\beta}_0 = \arg\max & \textstyle\sum_{u=1}^{U} \sum_{i=1}^{I} \log p(y_{ui}; m(\theta_u^\top \beta_i, a_{ui}) + \gamma_u \hat{a}_{ui} + \beta_0, v(\theta_u^\top \beta_i, a_{ui})) \\ & + \textstyle\sum_{u=1}^{U} \log p(\theta_u) + \sum_{i=1}^{I} \log p(\beta_i) + \sum_{u=1}^{U} \log p(\gamma_u) + \log p(\beta_0),\end{aligned}$$

where $p(\theta_u)$, $p(\beta_i)$, $p(\gamma_u)$, and $p(\beta_0)$ are priors of the latent variables.

Forming recommendations. The deconfounded recommender predicts all of the potential ratings, $y_{ui}(1)$. We use it for two types of prediction: weak generalization and strong generalization [13]. 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. For an existing user u (weak generalization), it computes the potential ratings from the fitted outcome model,

$$\hat{y}_{ui}(1) = m(\hat{\theta}_u^\top \hat{\beta}_i, 1) + \hat{\gamma}_u \cdot \hat{a}_{ui} + \hat{\beta}_0. \tag{9}$$

For a new user u' (strong generalization), it fixes the item vectors V_i and β_i, and compute user vectors for the new user: it fits $U_{u'}$, $\theta_{u'}$, and $\gamma_{u'}$ from the exposure and the ratings of this new user u'. It finally computes the prediction,

$$\hat{y}_{u'i}(1) = m(\hat{\theta}_{u'}^\top \hat{\beta}_i, 1) + \hat{\gamma}_{u'} \cdot \hat{U}_{u'}^\top V_i + \hat{\beta}_0. \tag{10}$$

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 $\hat{y}_{ui}(1)$
Fit PF to the exposures $\{a_{ui}\}_{u,i}$ (Equation (4))
Compute substitute confounders $\{\hat{a}_{ui}\}_{u,i}$ (Equation (5))
Fit the outcome model to the data $\{(a_{ui}, y_{ui}(a_{ui}))\}_{u,i}$ (Equation (8))
Estimate potential outcome given treatment $\hat{y}_{ui}(1)$ (Equations (9) and (10))

The deconfounded recommender ranks all the items for each user based on $\hat{y}_{ui}(1), i = 1, \ldots, I$, and recommends highly ranked items.

Why does it work? Causal inference for recommendation systems is a *multiple causal inference*: there are multiple treatments. Specifically, each user is attached to a causal inference problem and her exposure (or non-exposure) to each movie is a treatment. Thus there are I treatments and an I-dimensional outcome of ratings. It is the multiplicity of treatments that enables causal inference with unobserved confounders [22]. Consider the I treatments of a user u, $a_u = (a_{u1}, \ldots, a_{uI}), a_{ui} \in \{0, 1\}$. PF learns a per-user latent variable θ_u from the exposure matrix a_{ui}. If the exposure model fits the data well, then the treatments $(a_{u1}, \ldots, a_{uI})$ are *conditionally independent* given θ_u. Per [22], the per-user latent variable θ_u captures all multi-treatment confounders, i.e., variables that correlate with multiple exposures and the ratings vector. Hence this θ_u (or functions of it, like $\hat{a}_{ui}$) can serve as a substitute confounders for causal inference.

This reasoning relies on one assumption, that no confounders affect the users' exposure to only one of the items. We posit that this assumption is often plausible in recommender systems. When there are more than thousands of items, it is unlikely to have a confounder that affects only one of them.

The following theorem formalizes this reasoning.

Theorem 1. *If no confounders affect the exposure to only one of the items, and if* $A_{ui} \sim$ *Poisson*$(U_u^\top V_i)$ *for some independent random vectors* U_u*'s and* V_i*'s, then the deconfounded recommender forms unbiased causal inference*

$$E[Y_{ui}(a)] = E[E[Y_{ui}(A_{ui}) \mid A_{ui} = a, U_u^\top V_i]] \quad \textit{for all } u, i.$$

This theorem follows from Corollary 7 of [22] and the fact that $U_u \perp\!\!\!\perp \mathbb{1}\{A_{ui} = a\} \mid p(A_{ui} = a \mid U_u^\top V_i)$ [4]. See Appendix A for a full proof, and see [22] for the theoretical underpinnings of this approach.

3 Empirical Studies

We study the recommendation performance of the deconfounded recommender across four datasets. The deconfounded recommender improves recommendation. Further, its performance is more stable across regular and intervened test sets, indicating that it is a causal model. Before presenting the empirical performance, we first discuss the evaluation criteria of causal recommendation models.

3.1 Evaluation of causal recommendation models

Causal inference on recommender systems poses challenges for evaluation. We need to evaluate how a model performs across all potential outcomes:

$$\tfrac{1}{U} \textstyle\sum_{u=1}^{U} \ell(\{\hat{y}_{ui}\}_{i \in \{1,\ldots,I\}}, \{y_{ui}(1)\}_{i \in \{1,\ldots,I\}}),$$

where ℓ is a loss function, such as mean squared error or normalized discounted cumulative gain (NDCG). The challenge is that we don't observe all potential outcomes. The traditional trick of random train-test splitting no longer works; it provides a biased sample of the user-item ratings, emphasizing popular items and active users.

An expensive solution is to create a randomized test set from experiments. We can randomly select from all items, and ask our users to interact and rate all them:

$$\frac{1}{U}\sum_{u=1}^{U}\ell(\{\hat{y}_{ui}\}_{i\in\mathcal{I}_u},\{y_{ui}(1))\}_{i\in\mathcal{I}_u}),$$

where $\mathcal{I}_u$ is a random subset of $\{1,...,I\}$. Such test sets are golden baselines, but they are difficult to obtain. The Yahoo! R3 [14] and coat shopping [20] datasets are accompanied with such random test sets.

What if randomized test sets are not available? How can we evaluate causal models?

Our solution is to "intervene" on heldout data. Causal models enjoy the property of invariance: the conditional distribution of the outcome given the treatments will not change if we intervene on a_u, that is if we change the distribution of the treatment assignments A_u. Thus, predictions from a causal model will work just as well under intervention as not. In contrast, predictions from an observation model can potentially be wrong if we actively intervene on a_u [16].

We exploit this invariance property for evaluation. We create intervened test sets, that is we change the distribution of which exposed cells we are using to test the predictions. For example, we select a (u,i)-entry in the heldout set into the intervened test set with probability: $p(u,i)\propto\frac{1}{pop_i}$, where pop_i counts how many times item i is rated in the training set.

We can also bin the items by their popularity and evaluate the models within each bin. Equivalently, we can also bin the users by their activity. We use such intervened test sets for evaluation in Section 3.3.

3.2 Recommendation with the deconfounded recommender

We first study the deconfounded recommender on two real-world datasets: Yahoo! R3 [14] and coat shopping [20]. 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?

Datasets. Yahoo! R3 [14] 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 [20] 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.

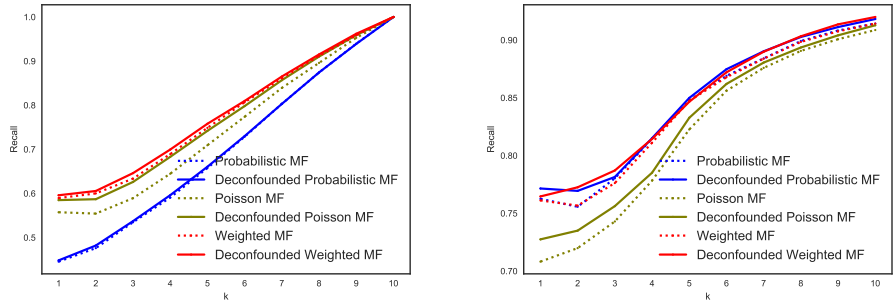
Experimental setup. For each dataset, we randomly split 80/20 the training set into training/-validation sets. We leave the random test set intact. Across all experiments, we select the best performing (according to log likelihood) latent dimensions for the recommendation models from $\{1,2,5,10,20,50,100\}$ and other hyperparameters using the validation set.

For the deconfounded recommender, we choose the best latent dimension combination of the treatment assignment model and the outcome model based on the performance (according to log likelihood) on the same validation set. Convergence is determined using log likelihood on the validation set. We predict potential ratings for both existing users (weak generalization) and new users (strong generalization) as in Section 2. Based on this prediction, we finally rank the items with nonzero ratings in the random test set.

Performance measures. To evaluate recommendation performance, we report three standard measures: NDCG, recall, and precision. See Appendix B for formal definitions.

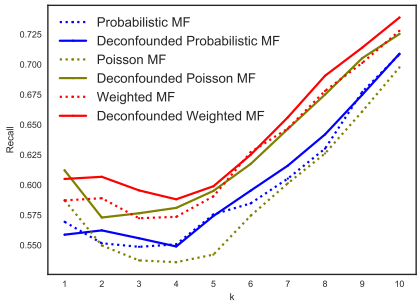
Results. Tables 1 and 2 show the recommendation performance of the deconfounded recommender and its classical counterpart. We explore probabilistic matrix factorization [15], Poisson matrix factorization [2], and weighted matrix factorization [6]; see Section 2 for details of these baseline models. Across datasets, the deconfounded recommender outperforms classical approaches for both weak and strong generalization: it produces better item rankings and improves retrieval quality.

Figures 1 and 2 show the complete recall curves and precision curves for $k=1,\ldots,10$. (k is chosen from 1–10 because the test set of Yahoo! R3 has 10 ratings per user; the coat dataset has 16.) The

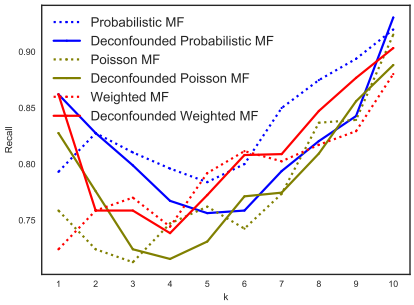


(a) Yahoo! R3 (weak)

(b) Yahoo! R3 (strong)



(c) coat (weak)



(d) coat (strong)

Figure 1: Recall@$k, k = 1, \ldots, 10$ for weak/strong generalization. The deconfounded recommender improves retrival quality for $k \leq 3$ in both datasets. (Higher is better.)

	Yahoo! R3			Coat		
	NDCG	Recall@2	Precision@2	NDCG	Recall@2	Precision@2
Probabilistic MF [15]	0.820	0.476	0.108	0.731	0.552	0.528
Deconfounded Probabilistic MF	**0.821**	**0.482**	**0.109**	**0.735**	**0.562**	**0.534**
Poisson MF [2]	0.846	0.554	0.125	0.734	0.550	0.522
Deconfounded Poisson MF	**0.858**	**0.587**	**0.132**	**0.747**	**0.573**	**0.545**
Weighted MF [6]	0.860	0.600	0.135	0.745	0.589	0.557
Deconfounded Weighted MF	**0.863**	**0.606**	**0.136**	**0.750**	**0.607**	**0.569**

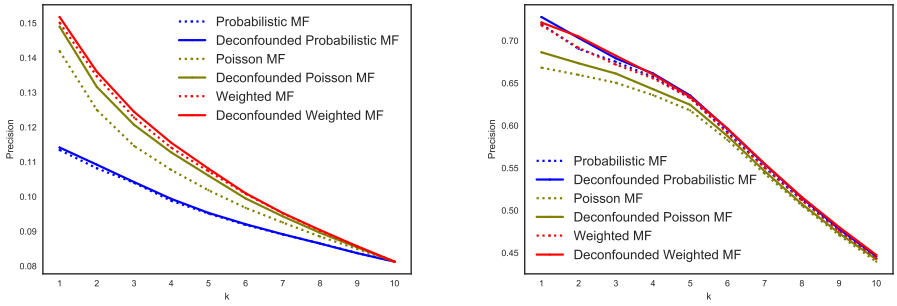
Table 1: Recommendation on random test sets for existing users (Weak generalization). The deconfounded recommender improves recommendation over classical approaches. (Higher is better.)

deconfounded recommender has better recall and precision curves across all ks for existing users. For new users, the deconfounded recommender improves recall and precision for $k \leq 3$.

Across the three metrics and two datasets, the deconfounded recommender improves over common recommendation models; it produces accurate predictions for *potential* user preferences by accommodating unobserved confounders in treatment assignments.

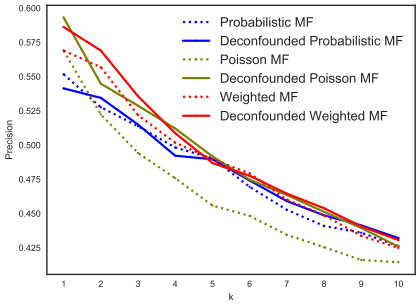
3.3 Causal vs noncausal recommendation

We next explore the differences between causal and noncausal recommendation models. Noncausal models are those that do not acknowledge the confounding bias as in Equation (2). In particular, traditional matrix factorization models, such as probabilistic or Poisson or weighted matrix factorization, are not causal. The deconfounded recommender is causal; Wang and Blei [22] provides a justification.

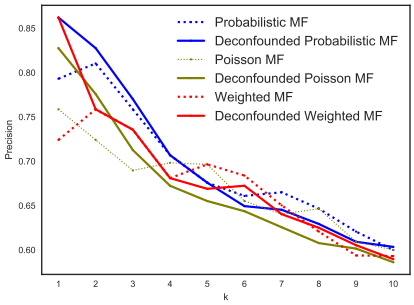


(a) Yahoo! R3 (weak)

(b) Yahoo! R3 (strong)



(c) coat (weak)



(d) coat (strong)

Figure 2: Precision@$k, k = 1, \ldots, 10$ for weak/strong generalization. The deconfounded recommender improves retrieval quality for $k \leq 3$ in both datasets. (Higher is better.)

	Yahoo! R3			Coat		
	NDCG	Recall@2	Precision@2	NDCG	Recall@2	Precision@2
Probabilistic MF [15]	0.828	0.756	0.690	0.850	0.818	0.810
Deconfounded Probabilistic MF	**0.835**	**0.769**	**0.703**	**0.858**	**0.828**	**0.830**
Poisson MF [2]	0.812	0.720	0.660	0.809	0.744	0.729
Deconfounded Poisson MF	**0.818**	**0.735**	**0.673**	**0.817**	**0.766**	**0.776**
Weighted MF [6]	0.829	0.757	0.692	0.830	0.756	0.749
Deconfounded Weighted MF	**0.834**	**0.773**	**0.705**	**0.841**	**0.759**	**0.759**

Table 2: Recommendation on random test sets for new users (Strong generalization). The deconfounded recommender improves recommendation over classical approaches. (Higher is better.)

How can we tell a causal recommendation model from a noncausal one? We compare them on regular and intervened test sets on two MovieLens datasets (http://grouplens.org/datasets/movielens/). We also study their performance stratified by item popularity and user activity. The performance of causal recommendation models is more robust to intervention and stratification.

Datasets. The Movielens 100k dataset contains 100,000 ratings from 1000 users on 1700 movies. The Movielens 1M dataset contains 1 million ratings from 6000 users on 4000 movies. Both datassets are not accompanied with random test sets as those in Section 3.2.

Experimental setup and performance measures. We employ the same experimental protocols for hyperparameter selection as before. We also use the same performance measure: NDCG, recall, and precision. For each dataset, we randomly split the training set into training/validation/test sets with 60/20/20 proportions. Given the test proportion, we create two test sets: one regular and one intervened. The regular test set comes from randomly selecting 30% from the test proportion. The

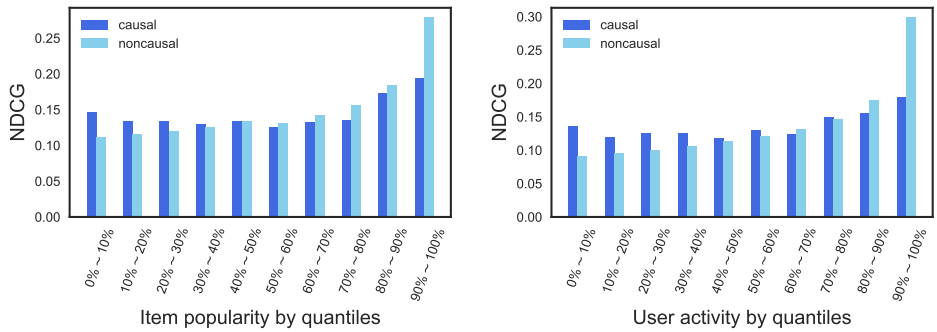


Figure 3: Recommendation performance of causal/noncausal models after binning (Movielens 1M). Noncausal recommendation does well on high-activity users and popular movies. But it does less well on lower-activity users and unpopular movies.

	Movielens 100K			Movielens 1M		
	NDCG	Recall@2	Precision@2	NDCG	Recall@2	Precision@2
Probabilistic MF [15]	0.915	0.938	0.921	0.914	0.952	0.937
Deconfounded Probabilistic MF	**0.920**	**0.946**	**0.931**	**0.920**	**0.959**	**0.946**
Poisson MF [2]	0.917	0.940	0.923	0.916	0.954	0.939
Deconfounded Poisson MF	**0.922**	**0.949**	**0.934**	**0.921**	**0.963**	**0.952**
Weighted MF [6]	0.918	0.941	0.924	0.916	0.953	0.938
Deconfounded Weighted MF	**0.920**	**0.947**	**0.933**	**0.922**	**0.961**	**0.949**

Table 3: Recommendation performance on intervened test sets. Causal models offer better recommendation when test sets have different exposure patterns from training sets. Causal models are robust to interventions on test sets.

intervened test sets comes from selecting each entry with a probability inversely proportional to its item popularity; see Section 3.1 for details.

Results. Table 3 presents the recommendation performance of the deconfounded recommender and their classical counterpart on intervened test sets. On intervened test sets, causal models often outperform by a large margin. Predictions from noncausal models suffer when the test set comes from a different treatment assignment distribution than the training set. It is because noncausal models ignore the confounding bias in treatment assignments.

On a regular test set, which is biased towards popular movies and popular users, the classical MF methods perform about the same (marginally better) than the deconfounded methods. We refer the reader to Appendix C for detailed results and discussions.

Finally, we compare causal and noncausal models on bins of users and items. We bin entries of the test set by quantiles of item popularity and user activity. We evaluate models within each bin. Figure 3 illustrates the difference. (This figure shows a comparison between probabilistic MF (noncausal) and the deconfounded probabilistic MF (causal). Comparisons on Poisson MF and Weighted MF are similar.) The causal model is less sensitive to binning; its recommendation performance is stable across bins of items or users. Noncausal recommendation does well on high-activity users and popular movies. But it does less well on lower-activity users and unpopular movies.

Evaluations on regular-vs-intervened test sets and binned items/users reveal the key difference between causal and noncausal models: causal models are robust to interventions on test sets while noncausal model are not. This robustness-to-intervention could serve to differentiate causal and noncausal models. It may also serve as a proxy of model predictability on potential outcomes, especially when random test sets are not available: causal models tend to predict the potential outcomes better.

4 Discussion

We develop the *deconfounded recommender*, a strategy to leverage classical recommendation models for causal predictions: how would a user rate a recommended movie? The deconfounded recommender utilizes Poisson factorization to infer confounders in treatment assignments; it then augments common recommendation models to correct for potential confounding bias. The deconfounded recommender improves recommendation and it enjoys stable performance against interventions on test sets.

References

[1] Ding, P., Li, F., et al. (2018). Causal inference: A missing data perspective. *Statistical Science*, 33(2):214–237.

[2] Gopalan, P., Hofman, J. M., and Blei, D. M. (2015). Scalable recommendation with hierarchical poisson factorization. In *UAI*, pages 326–335.

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

[4] Hirano, K. and Imbens, G. W. (2004). The propensity score with continuous treatments.

[5] Holland, P. W., Glymour, C., and Granger, C. (1985). Statistics and causal inference. *ETS Research Report Series*, 1985(2).

[6] Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE.

[7] Imbens, G. W. and Rubin, D. B. (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

[8] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

[9] 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 Conference on World Wide Web*, pages 929–934. ACM.

[10] 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*, pages 661–670. ACM.

[11] 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. International World Wide Web Conferences Steering Committee.

[12] Ling, G., Yang, H., Lyu, M. R., and King, I. (2012). Response aware model-based collaborative filtering. *arXiv preprint arXiv:1210.4869*.

[13] Marlin, B. M. (2004). Modeling user rating profiles for collaborative filtering. In *Advances in neural information processing systems*, pages 627–634.

[14] 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. ACM.

[15] Mnih, A. and Salakhutdinov, R. R. (2008). Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264.

[16] Peters, J., Bühlmann, P., and Meinshausen, N. (2016). Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012.

[17] Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

[18] Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688.

[19] Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331.

[20] Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., and Joachims, T. (2016). Recommendations as treatments: Debiasing learning and evaluation. *arXiv preprint arXiv:1602.05352*.

[21] Vanchinathan, H. P., Nikolic, I., De Bona, F., and Krause, A. (2014). Explore-exploit in top-n recommender systems via gaussian processes. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 225–232. ACM.

[22] Wang, Y. and Blei, D. M. (2018). The blessings of multiple causes. *arXiv preprint arXiv:1805.06826*.

[23] Zhao, X., Zhang, W., and Wang, J. (2013). Interactive collaborative filtering. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1411–1420. ACM.

Supplementary Material

A Proof of Theorem 1

Corollary 7 of Wang and Blei [5] implies

$$E[Y_{ui}(a)] = E[E[Y_{ui}(A_{ui}) \mid A_{ui} = a, U_u]] \quad \text{for all } u, i.$$

Therefore, we have

$$\begin{aligned}
E[Y_{ui}(a)] &= E[E[Y_{ui}(A_{ui}) \mid A_{ui} = a, U_u]] \\
&= E[E[Y_{ui}(A_{ui}) \mid \mathbb{1}\{A_{ui} = a\}, U_u]] \\
&= E[E[E[Y_{ui}(A_{ui}) \mid \mathbb{1}\{A_{ui} = a\}, U_u, p(A_{ui} = a \mid U_u^\top V_i)]]] \\
&= E[E[E[Y_{ui}(A_{ui}) \mid \mathbb{1}\{A_{ui} = a\}, p(A_{ui} = a \mid U_u^\top V_i)]]] \\
&= E[E[E[Y_{ui}(A_{ui}) \mid \mathbb{1}\{A_{ui} = a\}, U_u^\top V_i]]] \\
&= E[E[E[Y_{ui}(A_{ui}) \mid A_{ui} = a, U_u^\top V_i]]]
\end{aligned}$$

The first three equalities are due to basic probability facts. The fourth equality is due to $U_u \perp\!\!\!\perp \mathbb{1}\{A_{ui} = a\} \mid p(A_{ui} = a \mid U_u^\top V_i)$ [2]. The fifth equality is due to $A_{ui} \sim \text{Poisson}(U_u^\top V_i)$. The last equality is again due to basic probability facts.

B Performance measures

Denote $rank(u, i)$ as the rank of item i in user u's predicted list; let y_u^{test} as the set of relevant items in the test set of user u.[1]

- NDCG measures ranking quality:

$$DCG = \frac{1}{U} \sum_{u=1}^{U} \sum_{i=1}^{L} \frac{2^{rel_{ui}-1}}{\log_2(i+1)}, \quad NDCG = \frac{DCG}{IDCG},$$

  where L is the number of items in the test set of user u, the relevance rel_{ui} is set as the rating of user u on item i, and $IDCG$ is a normalizer that ensures $NDCG$ sits between 0 and 1.

- Recall@k. Recall@k evaluates how many relevant items are selected:

$$\text{Recall@k} = \frac{1}{U} \sum_{u=1}^{U} \sum_{i \in y_u^{test}} \frac{\mathbb{1}\{rank(u, i) \leq k\}}{\min(k, |y_u^{test}|)}.$$

- Precision@k: Precision@k evaluates how many selected items are relevant:

$$\text{Precision@k} = \frac{1}{U} \sum_{u=1}^{U} \sum_{i: rank(u,i) \leq k} \frac{\mathbb{1}\{i \in y_u^{test}\}}{k}.$$

C Causal models on regular test sets

On regular test sets, causal models and noncausal models perform similarly; noncausal models often outperform by a small margin. This difference in performance is often due to the causal models coming from an augmentation of noncausal models. The richer causal model often leads to higher-variance estimators and suboptimal performance on regular heldout data.

[1] We consider items with a rating greater than or equal to three as relevant.

	Movielens 100K			Movielens 1M		
	NDCG	Recall@2	Precision@2	NDCG	Recall@2	Precision@2
Probabilistic MF [4]	**0.923**	**0.952**	**0.939**	**0.923**	0.962	0.952
Deconfounded Probabilistic MF	**0.923**	0.951	0.937	0.922	**0.964**	**0.953**
Poisson MF [1]	**0.925**	**0.949**	**0.934**	**0.923**	**0.962**	**0.950**
Deconfounded Poisson MF	0.924	0.948	0.933	0.922	0.961	0.948
Weighted MF [3]	**0.925**	**0.950**	**0.937**	**0.924**	**0.963**	**0.953**
Deconfounded Weighted MF	0.923	0.949	0.934	0.922	0.962	0.952

Table 4: Recommendation performance on regular test sets. Noncausal models and causal models offer similar-quality recommendation. Noncausal models performs slightly better.

D Experimental Details

For weighted matrix factorization, we set weights of the observation by $c_{ui} = 1 + \alpha y_{ui}$ where $\alpha = 40$ following [3].

For Gaussian latent variables in probabilistic matrix factorization, we use priors $\mathcal{N}(0, 1)$ or $\mathcal{N}(0, 0.1^2)$.

For Gamma latent variables in PF, we use prior Gamma$(0.3, 0.3)$.

References

[1] Gopalan, P., Hofman, J. M., and Blei, D. M. (2015). Scalable recommendation with hierarchical poisson factorization. In *UAI*, pages 326–335.

[2] Hirano, K. and Imbens, G. W. (2004). The propensity score with continuous treatments.

[3] Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE.

[4] Mnih, A. and Salakhutdinov, R. R. (2008). Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264.

[5] Wang, Y. and Blei, D. M. (2018). The blessings of multiple causes. *arXiv preprint arXiv:1805.06826*.