Leveraging Causal Inference for Recommendation

Mahima Gaur mg6827@nyu.edu
Department of Computer Science
Courant Institute of Mathematical Sciences
New York, NY 10012, NY

Anya Trivedi aht324@nyu.edu
Department of Computer Science
Courant Institute of Mathematical Sciences
New York, NY 10012, NY

Utkarsh Khandelwal uk2051@nyu.edu
Department of Mathematics
Courant Institute of Mathematical Sciences
New York, NY 10012, NY

Instructor and TA: Rajesh Ranganath and Mark Goldstein

Abstract

Causal Inference is a powerful machine learning paradigm that aims to capture the cause and effect relationship between variables. In this project, we aim to frame the task of recommendation as a causal question- what would a user rate an item, if we "forced" to interact with it. However, a strong assumption that enables causal inference is ignorability, i.e all confounders are observed. In order to correctly gauge causal effects, we must correct for unobserved confounders by using a "deconfounded" recommender. Following the works of Wang et al. (2018) and Wang and Blei (2019), the deconfounder infers latent variables that act for substitute confounders. This project explores different factor models for inferring the substitute confounders on various recommendation data, to test the strength of the deconfounded recommender in different recommendation settings. In all cases, we observe a significant improvement in recommendation and inference by using the deconfounded recommender, as compared to a non-deconfounded baseline model, irrespective of the factor model used. A more qualitative analysis shows that the inferred latent dimensions are correctly able to capture item genres or types, yielding a more interpretative form of recommendation.

Keywords: Causal Inference, Recommendation, Probabilistic Factor Models

1. Introduction

Causal Inference is a machine learning paradigm that aims to capture the cause-effect relationship between variables- it explores how changes in a variable T, called the treatment, affect variable Y, called the outcome (Yao et al. (2021)). A unique consideration of the causal inference task is the presence of latent "confounding" variables- variables that affect both the treatment as well as the outcome of the experiment. Most causality algorithms are based on the underlying assumption that all confounders are observed, known as the "ignorability" assumption (Rosenbaum and Rubin (1983)). However, ignorability may be violated in certain important use cases.

Take the task of recommendation. The goal of a recommendation system is to make predictions of how a user would rate a particular item, given the users previous ratings. Recommendation can intuitively be posed as a causal inference question, by gauging how a user would rate an item, if we ”forced” them to interact with it (Wang et al. (2018)). Traditional approaches to recommendation only provides valid causal inferences if the users interacted with products at random. This assumption does not hold in reality, since people do not usually interact with items randomly, due to the presence of confounders. For example, if we are working with book variables, people tend to read books (the cause) and rate them (the outcome) based on the genre (confounder) of books they like.

It is thus necessary to use a deconfounded (Wang and Blei (2019)) approach to recommendation. This is done by building two models: an *exposure* model to gauge the items a user is likely to consider, given his past interactions. This model is then used to infer latent variables, which act as substitute confounders. Finally, an *outcome* model aims to predict the ratings the user would give to items, by looking at previous interactions, as well as the inferred substitute confounders (Wang et al. (2018)). In this project, we aim to apply the deconfounded recommender developed by Wang et al. (2018) to various recommendation datasets using various factor models to infer latent variables. We aim to test how well the deconfounded recommender performs for different data distributions and study whether the factor models used to infer the substitute confounders dictate model performance. To the authors’ knowledge, such a study of latent factor models on various recommendation and inference datasets have never been done before. Please note, in this paper we use “substitute confounders” and “latent factors” interchangeably.

2. Related Work

This work draws on two threads of related work.

The first body of related work if on using deconfounder algorithm for multiple causal inference. The Blessings of Multiple Causes Wang and Blei (2019), proposed a deconfounded algorithm that infers latent variables as substitute for confounding variables. Substitute counfounders can be used as control variables while performing causal inference. Following Wang and Blei (2019), The deconfounded recommender, Wang et al. (2018) proposes an causal approach to a recommender system involving two probabilistic models, exposure and outcome. The first models predicts the choice of movies of a user which is then be used as the substitute for confounder by the outcome model to predict user rating. The Medical Deconfounder, Zhang et al. (2019) applies the deconfounder algorithm to access treatment effects using electronic health records.

The second body of related work is on probabilistic modelling. Gopalan et al. (2015) proposes Hierarchical Poisson factorization for recommendation. Ranganath et al. (2015) proposed Deep Exponential Families (DEF), a class of latent variable models that generalizes to multiple setting.

3. Method

As discussed in the Introduction, our approach is divided into two major steps. The first step is to fit the exposure model. Our goal is to find latent variables for each user that fits well the exposure data. We need to model the distribution for the assigned causes given the latent variables. Factor models perfectly capture this because they make the causes conditionally independent for given latent variables for that particular user. Hence, if for a user i, there are

m causes affecting it, the factor model will provide us with:

$$p(a_{i1}, a_{i2}, \ldots, a_{im}|z_i) = \prod_{j=1}^{m} p(a_{ij}|z_i)$$

Factor model postulate per-individual latent variables z_i and use it to model assigned causes, i.e fit the exposure data.

$$z_i \sim p(.|\alpha)\ i = 1, \ldots k$$

$$a_{ij}|z_i \sim p(.|z_i, \theta_j)\ i = 1, \ldots m$$

In our study, we have used different factor models to fit exposure data. These different model places different conditions over prior (z_i) and likelihood $(a_{i1}, a_{i2}, \ldots, a_{im}|z_i)$ distributions. We talk about each factor model in detail in Appendix A. Using this prior and likelihood we compute the posterior distribution $(p(z_i|a_{i1}, a_{i2}, \ldots, a_{im}))$ of the latent variables using the Bayes theorem. Since computing the exact posterior is intractable, we use stochastic variational inference (Hoffman et al. (2013)) to infer the posterior.

The second step is fitting the outcome model. We infer the posterior distribution of z_i in the exposure model, and use it to compute a substitute confounder for all the causes. For a collection of causes A_{ij} for user i, the substitute confounder will be

$$\hat{z_i} = E[z_i|A_{ij}]$$

The substitute confounder is used to fit the outcome training data as

$$y_{ij} = \alpha\ a_{ij} + \beta\ \hat{z_i}$$

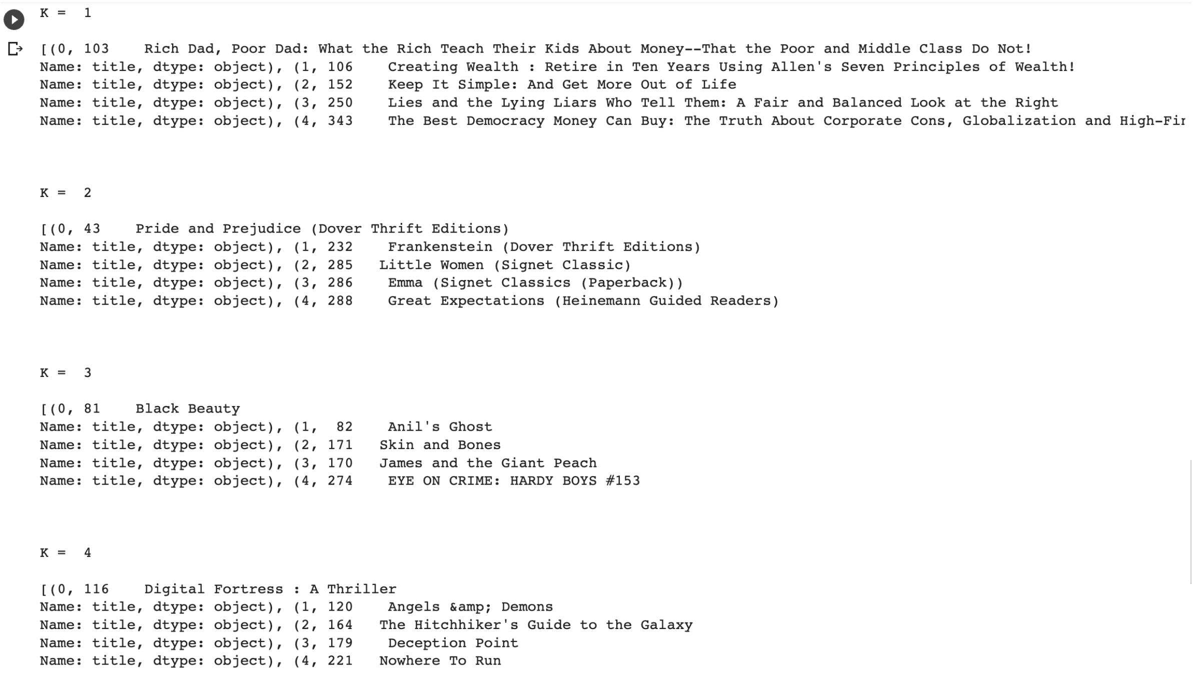
Here, i denotes user and j denotes the item and y_{ij} denotes the rating of item j by user i and a_{ij} denotes the exposure of user item j to user i and $\hat{z_i}$ is the substitute confounder. The derived coefficients $\hat{\alpha}$ and $\hat{\beta}$ can then be used to make predictions of ratings for user i to unexposed item j

$$y_{ij} = \hat{\alpha}\ a_{ij} + \hat{\beta}\ \hat{z_i}$$

4. Experimental Evaluation

To asses how well the factor model captures the population distribution of the causes, we follow the predictive check as described in Wang and Blei (2019), which tests how similar the likelihood of a replicated, heldout dataset, generated by the latent factors, is to that of the original data set. More details can be found in Appendix B.

Following the approach of Wang et al. (2018), we must take into account the unique challenges posed by evaluating causal recommendation systems. Traditionally, recommendation systems evaluate the loss (specifically, root mean squared error (RMSE)) of the predicted ratings over a randomly held out test set. However, causal inference involves evaluating over all possible outcomes, i.e the ratings of users whether or not they interacted with the item. Because we do not observe all potential outcomes, our evaluation may be biased. Intuitively, say we generate a test set by randomly splitting the data, as done in traditional recommendation. Ratings derived from this test set will be biased towards both popular users and popular items. A more concrete estimate would be to evaluate the ”per-item” RSME by averaging the RSMEs of all the items. This corrects the bias towards popular items which may historically receive more ratings.



```
K =  1

[(0, 103    Rich Dad, Poor Dad: What the Rich Teach Their Kids About Money--That the Poor and Middle Class Do Not!
Name: title, dtype: object), (1, 106    Creating Wealth : Retire in Ten Years Using Allen's Seven Principles of Wealth!
Name: title, dtype: object), (2, 152    Keep It Simple: And Get More Out of Life
Name: title, dtype: object), (3, 250    Lies and the Lying Liars Who Tell Them: A Fair and Balanced Look at the Right
Name: title, dtype: object), (4, 343    The Best Democracy Money Can Buy: The Truth About Corporate Cons, Globalization and High-Fir



K =  2

[(0, 43    Pride and Prejudice (Dover Thrift Editions)
Name: title, dtype: object), (1, 232    Frankenstein (Dover Thrift Editions)
Name: title, dtype: object), (2, 285   Little Women (Signet Classic)
Name: title, dtype: object), (3, 286    Emma (Signet Classics (Paperback))
Name: title, dtype: object), (4, 288    Great Expectations (Heinemann Guided Readers)



K =  3

[(0, 81    Black Beauty
Name: title, dtype: object), (1,  82    Anil's Ghost
Name: title, dtype: object), (2, 171   Skin and Bones
Name: title, dtype: object), (3, 170   James and the Giant Peach
Name: title, dtype: object), (4, 274    EYE ON CRIME: HARDY BOYS #153



K =  4

[(0, 116    Digital Fortress : A Thriller
Name: title, dtype: object), (1, 120    Angels &amp; Demons
Name: title, dtype: object), (2, 164   The Hitchhiker's Guide to the Galaxy
Name: title, dtype: object), (3, 179    Deception Point
Name: title, dtype: object), (4, 221   Nowhere To Run
```

Figure 1: Latent Dimensions of the Factor Models encode Book Genres. one dimension captures non-fiction books such as Rich Dad Poor Dad, another dimension captures Fiction novels like Pride and Prejudice, and another captures children's stories like James and the Giant Peach. We also see that a few dimensions capture author information- the forth dimension is almost exclusively Dan Brown novels.

5. Results

We evaluate the deconfounder on Book Reviews(Ziegler et al. (2005)), Job Suitability (Dehejia and Wahba (1999)), the MovieLens dataset, Song Ratings[1] and Amazon Product reviews[2]. The experimental Setup is summarized in Appendix C. Our results are summarized in Table 1. As seen, the deconfounded recommender improves the performance significantly compared to the baseline (non-deconfounded) approach for all data sets. The choice of the factor model does not make a significant difference, however. We then use the deconfounder to explore the books data, in order to evaluate the causal value of book genres and authors. We observe that each inferred latent dimension is able to capture specific genres, as well as author information, as seen in Figure 1.

6. Conclusion and Future Work

We evaluate the performance of the decounfounder on various datasets, and observe a significant improvement over the baseline model for all the datasets. Additionally we evaluate the performance with different probabilistic factor models and observe an improvement over

[1] `https://www.kaggle.com/code/mananhingorani/songs-recommendation-collaborative-filtering/data`

[2] `https://www.kaggle.com/code/saurav9786/recommender-system-using-amazon-reviews/data`

Data Set	Factor Model	P-Score	RMSE
Jobs Revenue	Baseline (Non-Deconfounded)	-	10.625
	HPMF	0.231	8.649
	PCA	0.216	8.218
	PPCA (Linear)	0.214	8.134
	PPCA (Quadratic)	0.486	**7.561**
	DEF	0.612	8.102
Book Recommendation	Baseline	-	1.223
	HPMF	0.421	**0.814**
	PCA	0.215	0.823
	PPCA (Linear)	0.228	0.821
	PPCA (Quadratic)	0.223	0.818
	DEF	0.479	0.817
Movie Recommendation	Baseline	-	1.238
	HPMF	0.256	**0.781**
	PCA	0.113	0.783
	PPCA (Linear)	0.417	0.789
	PPCA (Quadratic)	0.801	0.789
	DEF	0.468	0.791
E-Commerce Recommendation	Baseline	-	2.056
	HPMF	0.347	**0.782**
	PCA	0.178	0.784
	PPCA (Linear)	0.291	0.784
	PPCA (Quadratic)	0.113	0.785
	DEF	0.389	0.785
Songs Recommendation	Baseline	-	1.764
	HPMF	0.398	**0.882**
	PCA	0.201	0.891
	PPCA (Linear)	0.237	0.896
	PPCA (Quadratic)	0.318	0.893
	DEF	0.216	0.885

Table 1: Experimental Results: Performance of different Factor Models as measured by Root Mean Squared Error (RMSE) and Predictive Check Score (P-Score). P-Score test how similar the replicated dataset generated from the inferred latent variables is to the original dataset. how well the The smaller the P -value is, the more different the original dataset is from the replicated dataset. Ideal values are close to 0.5- the log likelihood of the replicated data is similar to that of the original data. RMSE scores of the deconfounded recommender with different factor models show improvement as compared to the baseline (non deconfounded) results for all datasets.

baseline for all the models. We also observe the although Hierarchical Poisson Factorization performs the best, the results do not vary much with choice of factor model.

Future scope involves making HPF and DEF model more robust. We could consider latent variables per item as well. So, if latent variables for a user i is π_i and latent variables for item j is λ_j, substitute confounder $\hat{a_{ij}}$ for the user i and item i will be

$$\hat{a_{ij}} = E[{\pi_i}^T \lambda_j | \mathbf{a}]$$

Here $\mathbf{a}$ represents all available exposure data. We can create a more robust outcome model using the below equation:

$$y_{ij}(a) = {\theta_i}^T \beta_j \cdot a + \gamma_i \cdot \hat{a_{ui}} + \epsilon_{ij}$$

Here ϵ_{ij} is the noise sampled from a gaussian distribution.So, $\epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$ We can fit this improved outcome model using the observed data using Maximum A Posteriori Estimation (MAP). Using MAP we will get $\hat{\theta_i}$, $\hat{\beta_j}$, $\hat{\gamma_i}$ We can use parameters obtained after fitting to make recommendations for the unrated items by the users using:

$$y_{ij}(a) = \hat{\theta_i}^T \hat{\beta_j} \cdot a + \hat{\gamma_i} \cdot \hat{a_{ui}} + \epsilon_{ij}$$

Additionally, improvements can be made to the exposure model by using causal methods like Inverse Propensity weighing (IPW).

References

Rajeev H Dehejia and Sadek Wahba. Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American statistical Association*, 94 (448):1053–1062, 1999.

Prem Gopalan, Jake M. Hofman, and David M. Blei. Scalable recommendation with hierarchical poisson factorization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, page 326–335, Arlington, Virginia, USA, 2015. AUAI Press. ISBN 9780996643108.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

Rajesh Ranganath, Linpeng Tang, Laurent Charlin, and David Blei. Deep Exponential Families. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 762–771, San Diego, California, USA, 09–12 May 2015. PMLR. URL `https://proceedings.mlr.press/v38/ranganath15.html`.

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

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

Yixin Wang and David M Blei. The blessings of multiple causes. *Journal of the American Statistical Association*, 114(528):1574–1596, 2019.

Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. The deconfounded recommender: A causal inference approach to recommendation. *arXiv preprint arXiv:1808.06581*, 2018.

Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. A survey on causal inference. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(5): 1–46, 2021.

Linying Zhang, Yixin Wang, Anna Ostropolets, Jami J. Mulgrave, David M. Blei, and George Hripcsak. The medical deconfounder: Assessing treatment effects with electronic health records. *Machine Learning for Healthcare*, 2019.

Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32, 2005.

Appendix A. Factor Models for Substitute Confounders

Here we discuss how different factor model impose different conditions on priors and likelihood.

1. **Probabilistic Principal Component Analysis (PPCA)**
   PPCA was proposed in Tipping and Bishop (1999). We have worked with two different models for PPCA. One assumes a linear and other a quadratic polynomial form for likelihood distribution.
   Linear Model:
   Priors:
   $$w \sim \mathcal{N}(0, \mathbf{I})$$
   $$z_i \sim \mathcal{N}(0, \mathbf{I})$$
   Likelihood:
   $$a_{ij}|z_i \sim (z_i^T w, \sigma^2 \mathbf{I})$$
   Quadratic Model:
   Priors:
   $$b \sim \mathcal{N}(0, \mathbf{I})$$
   $$w \sim \mathcal{N}(0, \mathbf{I})$$
   $$w_2 \sim \mathcal{N}(0, \mathbf{I})$$
   $$z_i \sim \mathcal{N}(0, \mathbf{I})$$
   Likelihood:
   $$a_{ij}|z_i \sim (b + z_i^T w + z_i^{2\ T} w_2, \sigma^2 \mathbf{I})$$

2. **Hierarchical Poisson Factorization (HPF)**
   HPF was introduced in Gopalan et al. (2015).
   Priors: For every user i,
   $$\epsilon_i \sim Gamma(a', \frac{a'}{b'})$$
   $$z_i \sim Gamma(a, \epsilon_i)$$
   For every item j,
   $$\eta_j \sim Gamma(c', \frac{c'}{d'})$$
   $$\beta_j \sim Gamma(c, \eta_j)$$
   Likelihood:
   $$a_{ij} \sim Poisson(z_i^T \beta_j)$$

3. **Deep Exponential Family (DEF)**
   DEF was intorduced in Ranganath et al. (2015).
   This assumes the following distributions:
   Priors:
   $$w_{top} \sim Gamma(\alpha_w, \beta_w)$$
   $$w_{mid} \sim Gamma(\alpha_w, \beta_w)$$
   $$w_{bottom} \sim Gamma(\alpha_w, \beta_w)$$

$$z_{i,top} \sim Gamma(\alpha_z, \beta_z)$$

$$z_{i,mid} \sim Gamma(\alpha_z, \frac{\beta_z}{{w_{top}}^T z_{i,top}})$$

$$z_{i,bottom} \sim Gamma(\alpha_z, \frac{\beta_z}{{w_{mid}}^T z_{i,mid}})$$

Likelihood:

$$a_{ij}|z_{i,bottom} \sim Poisson({w_{bottom}}^T z_{i,bottom})$$

Appendix B. Predictive Checks for the Factor Models

In order to gauge the accuracy of the derived latent factor models, we ”replicate” (or sample) values from a held-out portion of our data, which was not used for training the factor model. Let the latent variable be denoted as z, and the held out data as a_{held}. We train on the non-heldout, observed data a_{obs}. We sample the held out data as:

$$p(a_{held}^{rep}|a_{obs}) = \int p(a_{held}|z)p(z|a_{obs})dz$$

Then, we compare the replicted data to the heldout data by calculating the predictive score:

$$PScore = p(t(a_{held}^{rep}) < t(a_{held}))$$

where

$$t(a_{held}) = E_z[logp(a_{held}|z)|a_{obs}]$$

Appendix C. Implementation Details

Source code can be found in the following GitHub repository: DeconfounderCausalInference
Hyperparameters used for PPCA Model:
Number of epochs = 500

Hyperparameters used for HPF Model:
Latent Dimensions = 30
$a' = 0.3$, $b' = 1.0$, $c' = 0.3$, $d' = 1.0$, $a = 0.3$, $c = 0.3$
Number of epochs = 100

Hyperparameters used for DEF Model:
Top Layer Latent Dimensions = 10
Mid Layer Latent Dimensions = 4
Bottom Layer Latent Dimensions = 5
$\alpha_z = 0.1$, $\beta_z = 0.1$, $\alpha_w = 0.1$, $\beta_w = 0.3$
Number of epochs = 30