Causal Embeddings for Recommendation: An Extended Abstract *

Stephen Bonner[1,2] and **Flavian Vasile**[2]
[1]Durham University, Durham, UK
[2]Criteo AI Lab, Paris, France
s.a.r.bonner@durham.ac.uk, f.vasile@criteo.com

Abstract

Recommendations are commonly used to modify user's natural behavior, for example, increasing product sales or the time spent on a website. This results in a gap between the ultimate business objective and the classical setup where recommendations are optimized to be coherent with past user behavior. To bridge this gap, we propose a new learning setup for recommendation that optimizes for the Incremental Treatment Effect (ITE) of the policy. We show this is equivalent to learning to predict recommendation outcomes under a fully random recommendation policy and propose a new domain adaptation algorithm that learns from logged data containing outcomes from a biased recommendation policy and predicts recommendation outcomes according to behaviour under random exposure. We compare our method against state-of-the-art factorization methods, in addition to new approaches of causal recommendation and show significant improvements.

1 Introduction

In recent years, online commerce has outpaced the growth of traditional commerce. As such, research work on *recommender systems* has also grown significantly, with recent *Deep Learning (DL)* approaches achieving state-of-the-art results. Broadly, these DL approaches frame the recommendation task as either:

- A distance learning problem between pairs of products or pairs of users and products, measured with Mean Squared Error (MSE) and Area Under the Curve (AUC), like in the work by [Grbovic *et al.*, 2015; Vasile *et al.*, 2016; Pennington *et al.*, 2014].
- A next item prediction problem that models user behavior and predicts the next action, measured with ranking metrics such as Precision@K and Normalized Discounted Cumulative Gain (NDCG), as presented in [Hidasi *et al.*, 2015; Hidasi *et al.*, 2016].

*Work first published as: Bonner, S., Vasile, F. (2018). Causal embeddings for recommendation. In Proceedings of the 12th ACM Conference on Recommender Systems (pp. 104-112). ACM.

However, we argue that both approaches fail to model the inherent interventionist nature of recommendation, which should not only attempt to model the organic user behavior, but to actually attempt to optimally influence it according to a preset objective.

Using a causal vocabulary, we are interested in finding the treatment recommendation policy that maximizes the reward obtained from each user with respect to the control recommendation policy. This objective is traditionally denoted as the Individual Treatment Effect (ITE) [Rubin, 1974].

In our work, we introduce a modification to the classical matrix factorization approach which leverages both a large biased sample of biased recommendation outcomes and a small sample of randomized recommendation outcomes in order to create user and products representations. We show that using our method, we improve the ITE prediction performance over traditional matrix factorization and causal inference approaches.

1.1 Causal Vocabulary

Below we briefly introduce the causal vocabulary and notation that we will be using throughout the paper.

The Causal Inference Objective
In the *classical setup*, we want to determine the causal effect of one single action which constitutes the treatment versus the control case where no action or a placebo action is undertaken (do vs. not do). In the *stochastic setup*, we want to determine the causal effect of a stochastic treatment policy versus the baseline control policy. In this case, both treatment and control are distributions over all possible actions. We retrieve the classical setup as a special case.

Recommendation Policy
We assume a *stochastic policy* π_x that associates to each user u_i and product p_j a probability for the user u_i to be exposed to the recommendation of product p_j:

$$p_j \sim \pi_x(.|u_i)$$

For simplicity we assume showing no products is also a valid intervention in $\mathcal{P}$.

Policy Rewards
Reward r_{ij} is distributed according to an unknown conditional distribution r depending on u_i and p_j:

$$r_{ij} \sim r(.|u_i, p_j)$$

The reward R^{π_x} associated with a policy π_x is equal to the sum of the rewards collected across all incoming users by using the associated personalized product exposure probability:

$$R^{\pi_x} = \sum_{ij} r_{ij}\pi_x(p_j|u_i)p(u_i) = \sum_{ij} R_{ij}^{\pi_x}$$

Individual Treatment Effect

The *Individual Treatment Effect (ITE)* value of a policy π_x for a given user i and a product j is defined as the difference between its reward and the control policy reward:

$$ITE_{ij}^{\pi_x} = R_{ij}^{\pi_x} - R_{ij}^{\pi_c}$$

We are interested in finding the policy π^* with the *highest sum of ITEs*:

$$\pi^* = arg\max_{\pi_x}\{ITE^{\pi_x}\}$$

where: $ITE^{\pi_x} = \sum_{ij} ITE_{ij}^{\pi_x}$

Optimal ITE Policy

It is easy to show that, starting from any control policy π_c, the best incremental policy π^* is the policy that shows deterministically to each user u_i the product p_i^* with the highest personalized reward r_i^*:

$$\pi^* = \pi_{det} = \begin{cases} 1, & \text{if } p_j = p_i^* \\ 0, & \text{otherwise} \end{cases}$$

IPS Solution For π^*

In order to find the optimal policy π^* we need to find for each user u_i the product with the highest personalized reward r_i^*.

In practice we do not observe directly r_{ij}, but $y_{ij} \sim r_{ij}\pi_c(p_j|u_i)$.

The current approach to estimate r_{ij} constitutes in using *Inverse Propensity Scoring (IPS)*-based methods to predict the unobserved reward r_{ij}:

$$\hat{r}_{ij} \approx \frac{y_{ij}}{\pi_c(p_j|u_i)}$$

This assumes we have incorporated randomization in the current policy π_c. Even with the existence of randomization, the main shortcoming of IPS-based estimators is that they do not handle well big shifts in exposure probability between treatment and control policies (products with low probability under the logging policy π_c will tend to have higher predicted rewards).

Addressing the Variance Issues of IPS

Ideally, in order to maintain minimum variance, we should collect data using fully randomized recommendations, e.g. when: $\pi_c = \pi_{rand}$. However, this means zero recommendation performance and therefore cannot be a solution in practice.

Our question: Could we learn from π_c a predictor for performance under π_{rand} and use it to compute the optimal product recommendations p_i^*?

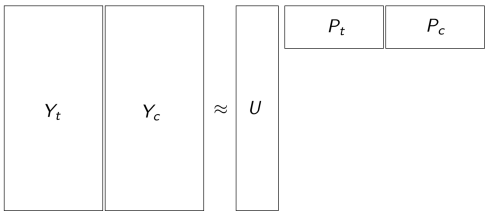

Figure 1: The joint MF problem.

2 Our Approach: Causal Embeddings (CausE)

We are interested in building a good predictor for recommendation outcomes under random exposure for all the user-product pairs, which we denote as $\hat{y}_{ij}^{rand}$. We make the assumption that we have access to a large sample S_c from the logging policy π_c and a small sample S_t from the randomized treatment policy $\pi_{t=rand}$ (e.g. the logging policy π_c uses *e-greedy* randomization).

To this end, we propose a multi-task objective that jointly factorizes the matrix of observations $y_{ij}^c \in S_c$ and the matrix of observations $y_{ij}^t \in S_t$. Our approach is inspired by the work in [Rosenfeld *et al.*, 2016] and shares similarities with other domain-adaptation based models for counterfactual inference such as the work in [Johansson *et al.*, 2016; Shalit *et al.*, 2017].

2.1 Predicting Rewards Via Matrix Factorization

By using a matrix factorization model, we assume that both the expected factual control and treatment rewards can be approximated as linear predictors over the **shared** user representations u_i, as shown in Fig. 1.

$$y_{ij}^c \approx < p_j^c, u_i >$$
$$y_{ij}^t \approx < p_j^t, u_i >$$

As a result, we can approximate the ITE of a user-product pair i, j as the difference between the two, see eq.1 below:

$$\widehat{ITE}_{ij} = < p_j^t, u_i > - < p_j^c, u_i > = < w_j^{\Delta}, u_i > \quad (1)$$

Proposed Joint Optimization Solution

The joint optimization objective contains two terms, one measuring the performance of the solution on the treatment sample and one on the control sample. The novel part of the objective comes from the additional constraint on the distance between the treatment and control vectors for the same action/item, that can be directly linked to the ITE effect of the item.

Sub-objective #1: treatment loss term L_t. We define the first part of our joint prediction objective as the supervised predictor for y_{ij}^t, trained on the limited sample S_t, as shown in the eq. 2 below:

$$L_t(P_t) = \sum_{(i,j,y_{ij})\in S_t} l_{ij}^t = L(UP_t, Y_t) + \Omega(P_t) \quad (2)$$

where:

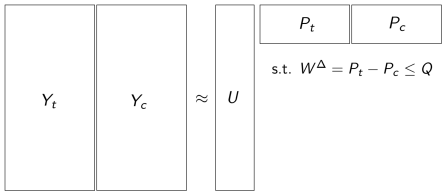

Figure 2: The final joint MF objective.

- P_t is the parameter matrix of treatment product representations.
- U is the fixed matrix of the user representations.
- Y_t is the observed rewards matrix.
- L is an arbitrary loss function.
- $\Omega(.)$ is a regularization term over the model parameters.

Linking the control and treatment effects. Additionally, we can use the translation factor in order to be able to use the model built from the treatment data S_t to predict outcomes from the control distribution S_c:

$$< p_j^c, u_i >=< p_j^t - w_j^\Delta, u_i >$$

Sub-objective #2: control loss term L_c. Now we want to leverage our ample control data S_c and we can use our treatment product representations through a translation:

$$L_c(P_t, W^\Delta) = \sum_{(i,j,y_{ij}) \in S_c} l_{ij}^c$$
$$= L(U(P_t - W^\Delta), Y_c) + \Omega(P_t, W^\Delta)$$

which can be written equivalently as:

$$L_c(P_t, P_c) = \sum_{(i,j,y_{ij}) \in S_c} l_{ij}^c = L(UP_c, Y_c) + \Omega(P_c, W^\Delta) \tag{3}$$

where we regularize the control P_c against the treatment embeddings P_t ($W^\Delta = P_t - P_c$). As shown in the eq. 4 below, we can see that IPS is a function of W^Δ. Therefore, by regularizing W^Δ we are effectively putting a constraint on the magnitude of the IPS term.

$$IPS_{ij} = \frac{\pi_t(p_j|u_i)}{\pi_c(p_j|u_i)} = \frac{< u_i, p_j^t >}{< u_i, p_j^c >} = 1 + \frac{< u_i, w_j^\Delta >}{< u_i, p_j^c >} \tag{4}$$

Overall Joint Objective

By putting the two tasks together (L_t and L_c) and regrouping the loss functions and the regularizer terms, we have that:

$$L_{CausE}^{prod}(P_t, P_c) = L(P_t, P_c)$$
$$+\Omega_{disc}(P_t - P_c) + \Omega_{embed}(P_t, P_c) \tag{5}$$

where $L(.)$ is the reconstruction loss function for the concatenation matrix of P_t and P_c, $\Omega_{disc}(.)$ is a regularization function that weights the discrepancy between the treatment and control product representations and $\Omega_{embed}(.)$ is a regularization function that weights the representation vectors.

Generalization to User Shift

Our objective function can be altered to allow for the user representations to change, we obtain the equation below:

$$L_{CausE}^{user}(U_t, U_c) = L(U_t, U_c)$$
$$+\Omega_{disc}(U_t - U_c) + \Omega_{embed}(U_t, U_c)$$

Putting the loss functions associated with the user and product dimension together (L_{CausE}^{prod}, L_{CausE}^{user}), we reach the final loss function for our method:

$$L_{CausE}(P_t, P_c, U_t, U_c) = L(P_t, P_c, U_t, U_c)$$
$$+\Omega_{disc}(P_t - P_c, U_t - Uc) + \Omega_{embed}(P_t, P_c, U_t, U_c) \tag{6}$$

3 Experimental Results

3.1 Experimental Setup

The task is predicting the outcomes y_{ij}^{rand} under treatment policy π_{rand}, where all of the methods have available at training time a large sample of observed recommendations outcomes from π_c and a small sample from π_{rand}. Essentially this is a classical conversion-rate prediction problem so we measure *Mean-Squared Error (MSE)* and *Negative Log-Likelihood (NLL)*. We report lift over average conversation rate from the test dataset:

$$lift_x^{metric} = \frac{metric_x - metric_{AvgCR}}{metric_{AvgCR}}$$

3.2 Baselines

We compare our method with the following Matrix Factorization baselines:

Algorithm 1: CausE - Causal embeddings for recommendations.

Input : Mini-batches of $S_c = \{(u_i, p_j^c, \delta_{ij}^c)\}_{i=1}^{M_c}$ and $S_t = \{(u_i, p_j^t, \delta_{ij}^t)\}_{i=1}^{M_t}$, regularization parameters λ_{embed} and λ_{dist}, learning rate η

Output: P_t, P_c, U_t, U_c - Product and User Control and Treatment Matrices

```
Random initialization of P_t, P_c, U_t, U_c ;
while not converged do
    read batch of training samples;
    for each product p_j in P_c, P_t do
        Update product vector:
        p_j ← p_j − η∇L^prod_CausE(p, λ_embed, λ_dist))
    end
    for each user u_i in U_c, U_t do
        Update user vector:
        u_i ← u_i − η∇L^user_CausE(u, λ_embed, λ_dist))
    end
end
return P_t, P_c, U_t, U_c
```

Method	MovieLens10M (SKEW)			Netflix (SKEW)		
	MSE lift	**NLL lift**	**AUC**	**MSE lift**	**NLL lift**	**AUC**
BPR-no	–	–	0.693(±0.001)	–	–	0.665(±0.001)
BPR-blend	–	–	0.711(±0.001)	–	–	0.671(±0.001)
SP2V-no	+3.94%(±0.04)	+4.50%(±0.04)	0.757(±0.001)	+10.82%(±0.02)	+10.19%(±0.01)	0.752(±0.002)
SP2V-blend	+4.37%(±0.04)	+5.01%(±0.05)	0.768(±0.001)	+12.82%(±0.02)	+11.54%(±0.02)	0.764(±0.003)
SP2V-test	+2.45%(±0.02)	+3.56%(±0.02)	0.741(±0.001)	+05.67%(±0.02)	+06.23%(±0.02)	0.739(±0.004)
WSP2V-no	+5.66%(±0.03)	+7.44%(±0.03)	0.786(±0.001)	+13.52%(±0.01)	+13.11%(±0.01)	0.779(±0.001)
WSP2V-blend	+6.14%(±0.03)	+8.05%(±0.03)	0.792(±0.001)	+14.72%(±0.02)	+14.23%(±0.02)	0.782(±0.002)
BN-blend	–	–	0.794(±0.001)	–	–	0.785(±0.001)
CausE-avg	+12.67%(±0.09)	+15.15%(±0.08)	0.804(±0.001)	+15.62%(±0.02)	+15.21%(±0.02)	0.799(±0.002)
CausE-prod-T	+07.46%(±0.08)	+10.44%(±0.09)	0.779(±0.001)	+13.97%(±0.02)	+13.52%(±0.02)	0.789(±0.003)
CausE-prod-C	**+15.48%(±0.09)**	**+19.12%(±0.08)**	**0.814(±0.001)**	**+17.82%(±0.02)**	**+17.19%(±0.02)**	**0.821(±0.003)**

Table 1: Results for MovieLens10M and Netflix on the Skewed (SKEW) test datasets. All three versions of the *CausE* algorithm outperform both the standard and the IPS-weighted causal factorization methods, with *CausE-avg* and *CausE-prod-C* also out-performing BanditNet. We can observe that our best approach *CausE-prod-C* outperforms the best competing approaches *WSP2V-blend* by a large margin (21% MSE and 20% NLL lifts on the MovieLens10M dataset) and *BN-blend* (5% AUC lift on MovieLens10M).

- **Bayesian Personalized Ranking (BPR)** To compare our approach against a ranking based method, we use Bayesian Personalized Ranking (BPR) for matrix factorization on implicit feedback data [Rendle *et al.*, 2009].
- **Supervised-Prod2Vec (SP2V):** As a second factorization baseline we will use a Factorization Machine-like method [Rendle, 2010] that approximates y_{ij} as a sigmoid over a linear transform of the inner-product between the user and product representations.

And these Causal Inference baselines:

- **Weighted-SupervisedP2V (WSP2V):** We employ the SP2V algorithm on propensity-weighted data, this method is similar to the Propensity-Scored Matrix Factorization (PMF) from [Schnabel *et al.*, 2016] but with cross-entropy reconstruction loss instead of MSE/MAE.
- **BanditNet (BN):** To utilize BanditNet [Joachims *et al.*, 2018] as a baseline, we use SP2V as our target policy π_w. For the existing policy π_c, we model the behavior of the recommendation system as a popularity-based solution, described by the marginal probability of each product in the training data.

3.3 Experimental Datasets

We use the *Netflix* and *MovieLens10M* explicit rating datasets (1-5). In order to validate our method, we preprocess them as follows: We binarize the ratings y_{ij} by setting 5-star ratings to 1 (click) and everything else to zero (view only) and generate a skewed dataset (SKEW) with 70/10/20 train/validation/test event split that simulates rewards collected from uniform exposure π_t^{rand}, following a similar protocol with the one presented in previous counterfactual estimation work such as in [Liang *et al.*, 2016; Swaminathan and Joachims, 2015] and described in detail in the long version of our paper [Bonner and Vasile, 2018].

3.4 Experimental Setup: Exploration Sample S_t

We define 5 setups for incorporating the exploration data:

- **No adaptation** *(no)* - trained only on S_c.

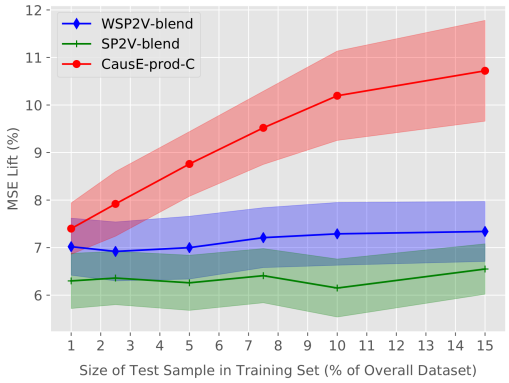

Figure 3: Change in MSE lift as more test set is injected into the blend training dataset.

- **Blended adaptation** *(blend)* - trained on the blend of the S_c and S_t samples.
- **Test adaptation** *(test)* - trained only on the S_t samples.
- **Product adaptation** *(prod)* - separate treatment embedding for each product based on the S_t sample.
- **Average adaptation** *(avg)* - average treatment product by pooling all the S_t sample into a single vector.

3.5 Results

Table 1 displays the results for running all the approaches on the datasets. Our proposed *CausE* method significantly outperforms all baselines across both datasets, demonstrating that it has a better capacity to leverage the small test distribution sample S_t. We observe that, out of the three *CausE* variants, *CausE-prod-C*, the variant that is using the regularized control matrix, clearly out-performs the others. Further, figure 3 highlights how CausE is able to make better use of increasing quantities of test distribution present in the training data compared with the baselines.

References

[Bonner and Vasile, 2018] Stephen Bonner and Flavian Vasile. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 104–112. ACM, 2018.

[Grbovic *et al.*, 2015] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1809–1818, New York, NY, USA, 2015. ACM.

[Hidasi *et al.*, 2015] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[Hidasi *et al.*, 2016] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 241–248. ACM, 2016.

[Joachims *et al.*, 2018] Thorsten Joachims, Artem Grotov, Adith Swaminathan, and Maarten de Rijke. Deep learning with logged bandit feedback. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[Johansson *et al.*, 2016] Fredrik D Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. *arXiv preprint arXiv:1605.03661*, 2016.

[Liang *et al.*, 2016] Dawen Liang, Laurent Charlin, and David M Blei. Causal inference for recommendation. 2016.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014. Association for Computational Linguistics.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press, 2009.

[Rendle, 2010] Steffen Rendle. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 995–1000. IEEE, 2010.

[Rosenfeld *et al.*, 2016] Nir Rosenfeld, Yishay Mansour, and Elad Yom-Tov. Predicting counterfactuals from large historical data and small randomized trials. *arXiv preprint arXiv:1610.07667*, 2016.

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

[Schnabel *et al.*, 2016] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1670–1679, 2016.

[Shalit *et al.*, 2017] Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3076–3085. JMLR. org, 2017.

[Swaminathan and Joachims, 2015] Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, 16:1731–1755, 2015.

[Vasile *et al.*, 2016] Flavian Vasile, Elena Smirnova, and Alexis Conneau. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 225–232. ACM, 2016.