Counterfactual Prediction for Bundle Treatment

Hao Zou
Tsinghua Univ.

Peng Cui
Tsinghua Univ.

Bo Li
Tsinghua Univ.

Zheyan Shen
Tsinghua Univ.

Jianxin Ma
Alibaba Group

Hongxia Yang
Alibaba Group

Yue He
Tsinghua Univ.

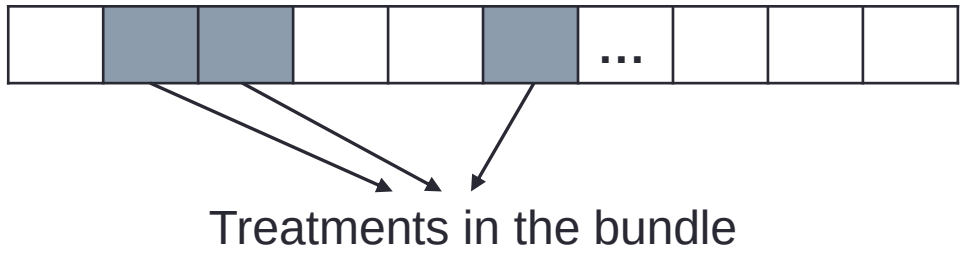
Background

- Decision making problem is widespread in practice.
  - In healthcare, decide the medicine to improve patient's health.
  - In education, decide the teaching method to improve the grade of students
  - In recommender system, decide the exposed item/advertisement to improve CTR.
  - …
- Estimating the individual outcome of different treatment can help it.

Background

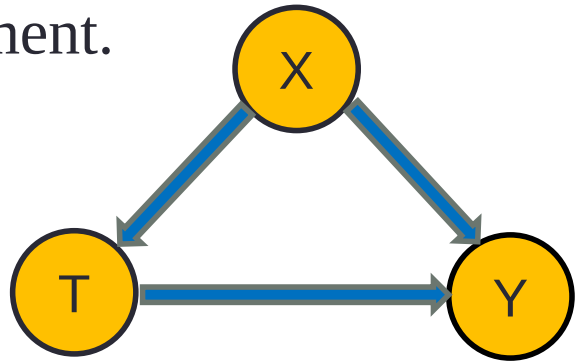
- Traditional literature investigate the treatment of the single variable.
  - Binary Treatment e.g. take medicine or not $\mathbf{T} \in \{0,1\}$
  - Multi-level Treatment e.g. educational level $\mathbf{T} \in \{0,1,2\ldots,\mathrm{m}\}$
  - Continuous Treatment e.g. time/dosage $\mathbf{T} \in [a,b]$

- We consider bundle treatment setting $\mathbf{T} \in \{\mathbf{0},\mathbf{1}\}^{\boldsymbol{p}}$ (a combination of different binary treatments)
  - E.g. a bundle of items selected from a candidate pool



Challenge

- Golden standard for causal inference--RCT
  - Expensive and sometimes limited
- Alternate approach: Observational dataset + Machine learning technology
- Some Challenge:
  - Only observe the outcome of one treatment for each sample
  - Confounding bias in the observational data induced by the treatment assignment policy. (Treatment is correlated with confounders). Decorrelating T and X is more difficult for bundle treatment.



Problem Formulation

- **Observational Dataset** $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{t}_i, y_i)\}_{i=1,2,3\ldots,n}$
  - $\mathbf{x}_i \in \mathbb{R}^d$: Confounder vector
  - $\mathbf{t}_i \in \{0,1\}^p$: Treatment vector
  - $y_i \in \mathbb{R}$: Outcome value
- **Target:** Learn a predictive model $f_{\theta_p}(\mathbf{X}, \mathbf{T}) \rightarrow y$ to predict the individual outcome given the confounder and treatment.

Related Works

- GANITE adopt generative adversarial nets framework to impute the counterfactual treatment outcome.
  - A generator function $g : \mathcal{X} \times \{0,1\}^k \times \mathcal{Y} \times [-1,1]^{k-1} \rightarrow \mathcal{Y}^k$ to generate the outcome of different treatments.
  - A discriminator to distinguish the factual outcome.

$$\min_{\mathbf{G}} \max_{\mathbf{D_G}} \mathbb{E}_{(\mathbf{x},\mathbf{t},y_f)\sim\mu_f} \left[\mathbb{E}_{\mathbf{z_G}\sim\mathcal{U}((-1,1)^k)} \left[\mathbf{t}^T \log \mathbf{D_G}(\mathbf{x},\tilde{\mathbf{y}}) + (\mathbf{1}-\mathbf{t})^T \log(1-\mathbf{D_G}(\mathbf{x},\tilde{\mathbf{y}}))\right]\right]$$

- This method does not solve the confounding bias problem.

Yoon, Jinsung, James Jordon, and Mihaela van der Schaar. "GANITE: Estimation of individualized treatment effects using generative adversarial nets." *International Conference on Learning Representations*. 2018.

Related Works

- CFR adopt the generalization bound in domain adaptation and learn the treatment (domain) invariant representation to remove the distribution discrepancy between the two treatment groups.

Bound: $\epsilon_{CF}(h,\Phi) \leq$

$$(1-u)\epsilon_F^{t=1}(h,\Phi) + u\epsilon_F^{t=0}(h,\Phi) + B_\Phi \cdot IPM_G\left(p_\Phi^{t=1}, p_\Phi^{t=0}\right),$$

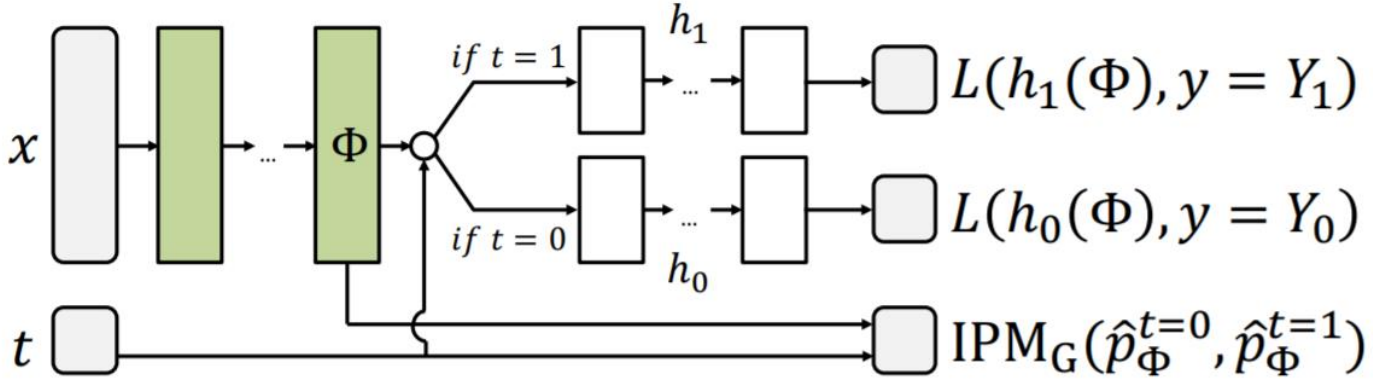
Loss function:

$$\min_{\substack{h,\Phi \\ \|\Phi\|=1}} \frac{1}{n}\sum_{i=1}^{n} w_i \cdot L\left(h(\Phi(x_i), t_i), y_i\right) + \lambda \cdot \Re(h) + \alpha \cdot \text{IPM}_G\left(\{\Phi(x_i)\}_{i:t_i=0}, \{\Phi(x_i)\}_{i:t_i=1}\right),$$

with $w_i = \frac{t_i}{2u} + \frac{1-t_i}{2(1-u)}$, where $u = \frac{1}{n}\sum_{i=1}^{n} t_i$,

and $\Re$ is a model complexity term.

CFR:



Shalit U, Johansson F D, Sontag D. Estimating individual treatment effect: generalization bounds and algorithms[C]//International Conference on Machine Learning. PMLR, 2017: 3076-3085.

Related Works

- Based on the framework of the CFR, re-weight the samples to remove the distribution discrepancy between the two treatment groups further.
- Loss Function:

$$J(h,\Phi) = \frac{1}{N}\sum_{i=1}^{N}\omega_i \cdot L\left[\, y_i,\, h^{t_i}\left(\Phi(x_i)\right)\right] + \lambda \cdot \mathfrak{R}(h) + \alpha \cdot \mathrm{IPM}\left(\{\Phi(x_i)\}_{i:t_i=0}, \{\Phi(x_i)\}_{i:t_i=1}\right)$$

$$\omega_i = 1 + \frac{\Pr(\,\phi_i \mid \neg t_i\,)}{\Pr(\,\phi_i \mid t_i\,)}$$

- Weight is calculated based on propensity score:

$$\omega_i = 1 + \frac{\Pr(\,\phi_i \mid \neg t_i\,)}{\Pr(\,\phi_i \mid t_i\,)} = 1 + \frac{\Pr(\,t_i\,)}{1-\Pr(\,t_i\,)} \cdot \frac{1-\pi_0(\,t_i \mid \phi_i\,)}{\pi_0(\,t_i \mid \phi_i\,)}$$

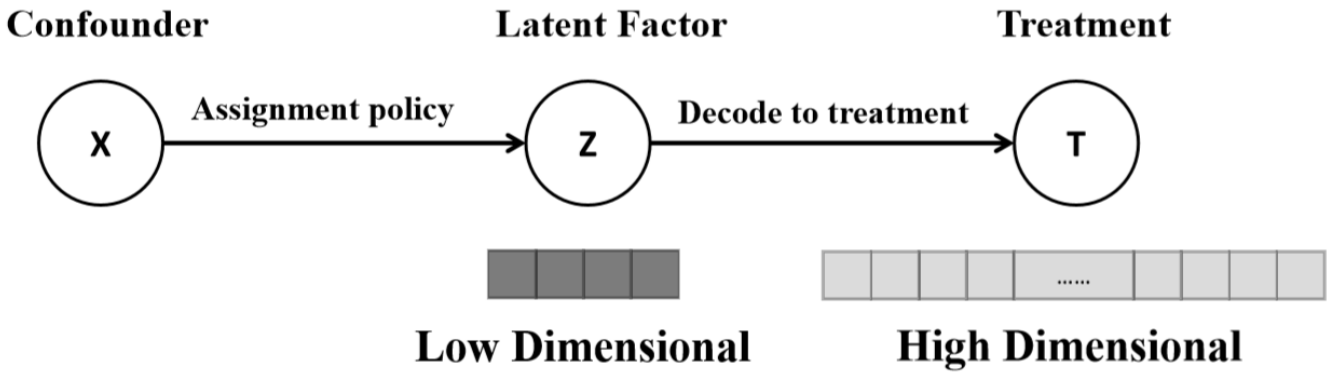
Hassanpour N, Greiner R. CounterFactual Regression with Importance Sampling Weights[C]//IJCAI. 2019: 5880-5887.

Related Works

- Sample re-weighting in causal inference:
- Inverse (generalized) propensity score
  - $e_i = P(\mathbf{T_i}|\mathbf{X_i})$
  - Shortcoming: Need correct model specification
- Confounder balancing
  - Directly learn sample weights to balance moment of confounder in treatment groups
  - Shortcoming: Only finite moments can be involved in computation.
- Under the bundle treatment setting, high dimensional property brings challenge.

Method

- We assume the high dimensional bundle treatment has low dimensional latent structure and can be determined by several latent factors.



- We can remove the confounding bias through decorrelate confounders and the latent representation of treatments.

Method

- We apply VAE to learn the latent factors of treatments.
  - Get the encoder $q_\phi(\mathbf{Z}|\mathbf{T})$ and decoder $p_\varphi(\mathbf{T}|\mathbf{Z})$.
- Transform the original dataset $\mathcal{D}$ into latent space $\{(\mathbf{x}_i, \mathbf{z})\}_{1\le i\le n}, \mathbf{z}\sim q_\phi(\mathbf{Z}|\mathbf{t}_i)$. Re-weight it to the ideal dataset $\{(\mathbf{x}_i, \mathbf{z})\}_{1\le i\le n}, \mathbf{z}\sim\mathcal{N}(\mathbf{0}, \mathbf{I})$.
  - Label the data points $\{(\mathbf{x}_i, \mathbf{z})\}_{1\le i\le n}, \mathbf{z}\sim q_\phi(\mathbf{Z}|\mathbf{T})$ as positive points (L=1) and the data points $\{(\mathbf{x}_i, \mathbf{z})\}_{1\le i\le n}, \mathbf{z}\sim\mathcal{N}(\mathbf{0}, \mathbf{I})$ as negative points (L=0).
  - Train a binary classifier to learn $p(L|\mathbf{X}, \mathbf{Z})$.

$$W_Z(\mathbf{X}, \mathbf{Z}) = \frac{p(\mathbf{X}, \mathbf{Z}|L=0)}{p(\mathbf{X}, \mathbf{Z}|L=1)} = \frac{p(L=1)}{p(L=0)} \cdot \frac{p(L=0|\mathbf{X}, \mathbf{Z})}{p(L=1|\mathbf{X}, \mathbf{Z})} = \frac{p(L=0|\mathbf{X}, \mathbf{Z})}{p(L=1|\mathbf{X}, \mathbf{Z})}$$

Method

- For one sample, $\mathbf{t}_i$ corresponds to a distribution of latent representation $\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{t}_i)$. Need to aggregate weights $W_Z(\mathbf{X},\mathbf{Z})$ in $\{(\mathbf{x}_i,\mathbf{z})\}_{1\le i\le n}, \mathbf{z}\sim q_\phi(\mathbf{Z}|\mathbf{t}_i)$ to calculate variational sample weights.

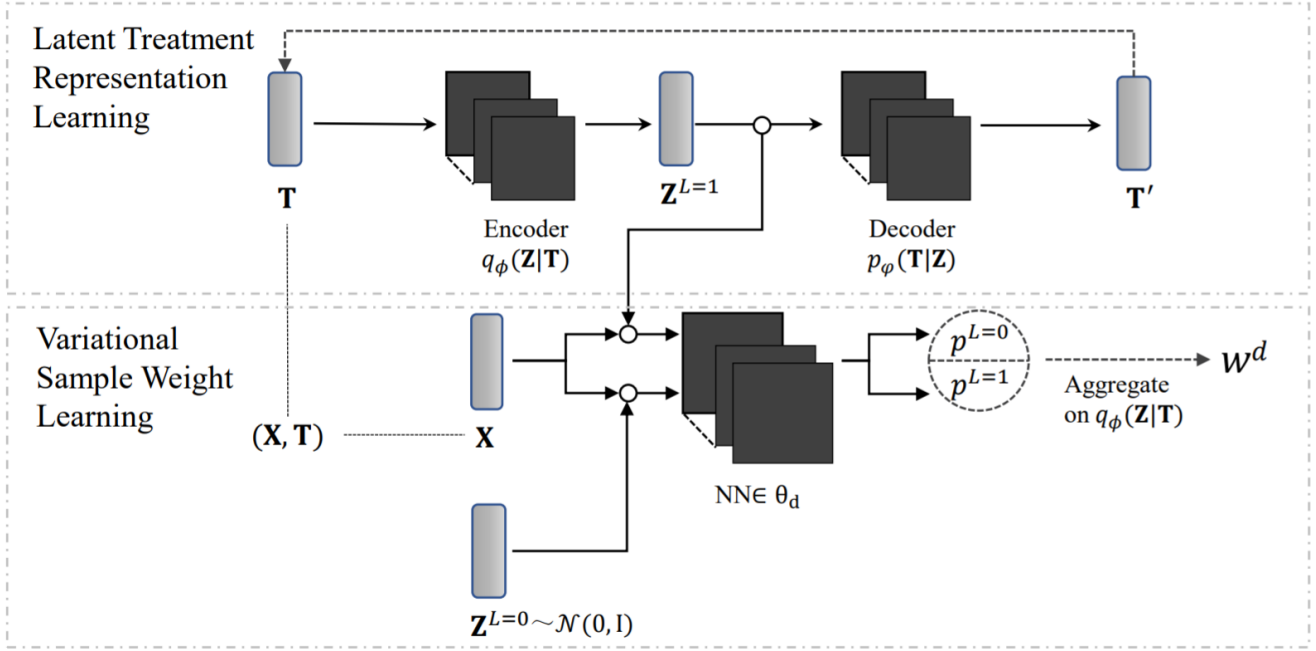
$$
\begin{aligned}
w_i^d = W_T(\mathbf{x}_i,\mathbf{t}_i) &= \frac{p(\mathbf{t}_i)}{p(\mathbf{t}_i|\mathbf{x}_i)} = \frac{p(\mathbf{t}_i)}{\int_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}_i)p(\mathbf{t}_i|\mathbf{z})d\mathbf{z}} = \frac{1}{\int_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}_i)\frac{p(\mathbf{t}_i|\mathbf{z})}{p(\mathbf{t}_i)}d\mathbf{z}} \\
&= \frac{1}{\int_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}_i)\frac{p(\mathbf{z}|\mathbf{t}_i)}{p(\mathbf{z})}d\mathbf{z}} = \frac{1}{\int_{\mathbf{z}} p(\mathbf{z}|\mathbf{t}_i)\frac{p(\mathbf{z}|\mathbf{x}_i)}{p(\mathbf{z})}d\mathbf{z}} = \frac{1}{\int_{\mathbf{z}} p(\mathbf{z}|\mathbf{t}_i)\frac{p(\mathbf{z},\mathbf{x}_i)}{p(\mathbf{z})p(\mathbf{x}_i)}d\mathbf{z}} \\
&= \frac{1}{\int_{\mathbf{z}} p(\mathbf{z}|\mathbf{t}_i)\frac{1}{W_Z(\mathbf{x}_i,\mathbf{z})}d\mathbf{z}} = \frac{1}{\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{t}_i)}\left[\frac{1}{W_Z(\mathbf{x}_i,\mathbf{z})}\right]},
\end{aligned}
$$

Method

- Train a predictive model on the re-weighted dataset.

$$\mathcal{L}_{pre} = \frac{1}{n}\sum_{i=1}^{n} w_i \cdot \mathcal{L}(f_{\theta_p}(\mathbf{x}_i, \mathbf{t}_i), y_i)$$

- The sample weights is calculated as following:



Experiments

- Compare the learned model under variational sample re-weighting (VSR) with the following baselines:
  - *DNN*: It directly uses deep neural networks to predict.
  - $DNN\&W_{raw}$: deep predictive model + sample weights calculated by density ratio estimation of raw treatments.
  - $DNN\&W_{AE}$: deep predictive model + sample weights calculated by density ratio estimation of AE representation.
  - $DNN\&W_{IR}$: deep predictive model + regularizer constraining independence of treatment and confounders.

Experiments

- **Data generation**
  - Generate confounders $\mathbf{X} = (x_1, x_2, ..., x_d)$

    $$x_1, x_2, ..., x_d \overset{iid}{\sim} \mathcal{N}(0, 1)$$
  - Generate treatment **T** (**L** is latent factor, $\mathbf{L} \in \mathbb{R}^k$)

    $$\mathbf{L} = \mathbf{X} \cdot \mathbf{A} + \varepsilon_L, \quad \mathbf{F} = \mathbf{L} \cdot \mathbf{B}$$
  - Assume the bits $\{i_1, i_2, .., i_s\}$ with largest value in F.

    $$t_j = \begin{cases} 1 & j \in \{i_1, i_2, .., i_s\} \\ 0 & j \notin \{i_1, i_2, .., i_s\} \end{cases}$$
  - Outcome generation $\mathbf{y} = \sum_{i=1}^{d} \sum_{j=1}^{p} x_i d_{i,j} t_j + \varepsilon_y$
  - Confounder dim d=10, Latent dim k=3, Number of one-bits in treatment s=5
  - Metric: RMSE on the test dataset, where the treatments are randomly assigned regardless of confounders.

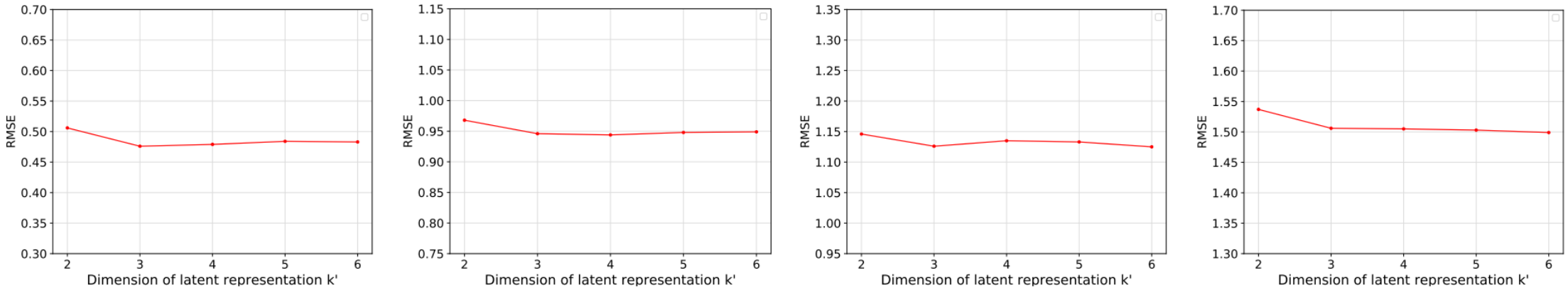
Experiments

Setting 1:Fix sample size $n = 10000$, varying dimension of treatments p								
p	$p = 10$		$p = 20$		$p = 30$		$p = 50$	
Methods	Mean	STD	Mean	STD	Mean	STD	Mean	STD
DNN	0.617	0.043	0.997	0.139	1.380	0.155	1.940	0.278
DNN&W_{raw}	0.528	0.044	0.997	**0.056**	1.197	0.092	1.543	**0.108**
DNN&W_{AE}	0.529	0.045	0.977	0.069	1.201	0.092	1.520	0.170
DNN&IR	0.624	0.059	1.059	0.118	1.377	0.164	1.930	0.302
DNN&W_{VSR}	**0.476**	**0.037**	**0.946**	0.067	**1.126**	**0.085**	**1.506**	0.152

Setting 2:Fix dimension of treatments $p = 10$, varying sample size n								
n	$n = 5000$		$n = 10000$		$n = 15000$		$n = 20000$	
Methods	Mean	STD	Mean	STD	Mean	STD	Mean	STD
DNN	0.677	0.083	0.617	0.043	0.658	0.159	0.434	0.063
DNN&W_{raw}	0.647	0.073	0.528	0.044	0.631	0.160	0.385	0.075
DNN&W_{AE}	0.624	0.063	0.529	0.045	0.589	0.072	0.400	0.066
DNN&IR	0.667	0.119	0.624	0.059	0.639	0.096	0.435	0.068
DNN&W_{VSR}	**0.572**	**0.053**	**0.476**	**0.037**	**0.518**	**0.064**	**0.367**	**0.044**

VSR can improve the performance of trained predictive model

Experiment



(a) $p = 10$ (b) $p = 20$ (c) $p = 30$ (d) $p = 50$

Figure 3: The testing RMSE of DNN&$\mathrm{W}_{\mathrm{VSR}}$ when varying the dimension of latent space of VAE k'. The true dimension of latent space $k = 3$.

Experiments

- **Data generation**
  - Document i is characterized by topic c_i and quality q_i. Confounder $X \in \mathbb{R}^d$ is the user's affinity to each topic.
  - Latent factors $\mathbf{L} = \mathbf{X} + \varepsilon_L, \varepsilon_L \sim \mathcal{N}(0, 0.81\mathbf{I})$, document score $Score_i = l_{c_i} + q_i$
  - Select s documents with highest score as recommended document to form the bundle treatment.
  - Predict the user's click rate on the bundle.
  - Number of topics d=4, selected documents s=4, sample size n=10000

RMSE of click rate prediction ($\times 10^{-2}$)								
Document number p	$p = 10$		$p = 20$		$p = 30$		$p = 50$	
Methods	Mean	STD	Mean	STD	Mean	STD	Mean	STD
DNN	2.694	0.589	3.941	0.716	4.415	0.582	4.443	0.613
DNN&W_{raw}	1.950	0.517	3.258	0.621	3.856	**0.455**	3.788	0.625
DNN&W_{AE}	1.711	0.407	3.312	0.741	3.683	0.515	3.623	0.619
DNN&IR	3.032	0.766	3.954	0.803	4.663	0.697	4.600	0.620
DNN&W_{VSR}	**1.596**	**0.349**	**2.923**	**0.407**	**3.318**	0.459	**3.385**	**0.598**

Conclusions

- Challenges of counterfactual prediction for bundle treatment:
  - Confounding bias in the observational data
  - High dimensional property and complexity of bundle treatment

We propose Variational Sample Re-weighting (VSR) algorithm for counterfactual prediction.

VSR algorithm utilize the low dimensional latent structure of bundle treatment.

Experiments show that reducing confounding bias can help to predict counterfactual outcome better.

Thank you!