Learning Individual Treatment Effects from Networked Observational Data

Ruocheng Guo[1], Jundong Li[1], and Huan Liu[1]

Arizona State University, Tempe, AZ 85287, USA
{rguo12,jundongl,huan.liu}@asu.edu

Abstract. With convenient access to observational data, learning individual causal effects from such data draws more attention in many influential research areas such as economics, healthcare, and education. For example, we aim to study how a medicine (treatment) would affect the health condition (outcome) of a certain patient. To validate causal inference from observational data, we need to control the influence of confounders - the variables which causally influence both the treatment and the outcome. Along this line, existing work for learning individual treatment effect overwhelmingly relies on the assumption that there are no hidden confounders. However, in real-world observational data, this assumption is untenable and can be unrealistic. In fact, an important fact ignored by them is that observational data can come with network information that can be utilized to infer hidden confounders. For example, in an observational study of the individual treatment effect of a medicine, instead of randomized experiments, the medicine is assigned to individuals based on a series of factors. Some factors (e.g., socioeconomic status) are hard to measure directly and therefore become hidden confounders of observational datasets. Fortunately, the socioeconomic status of an individual can be reflected by whom she is connected in social networks. With this fact in mind, we aim to exploit the network structure to recognize patterns of hidden confounders in the task of learning individual treatment effects from observational data. In this work, we propose a novel causal inference framework, the network deconfounder, which learns representations of confounders by unraveling patterns of hidden confounders from the network structure between instances of observational data. Empirically, we perform extensive experiments to validate the effectiveness of the network deconfounder on various datasets.

Keywords: Individual treatment effect · Confounding bias · Networked observational data.

1 Introduction

Recent years have witnessed the rocketing availability of observational data in a variety of highly influential research areas such as economics, healthcare, and education. Such data enables researchers to investigate the fundamental problem of learning individual-level causal effects of a certain interesting treatment (e.g.,

medicine) on an important outcome (e.g., health condition) without performing randomized experiments which can be rather expensive, time consuming, and even unethical [9,6]. For example, the easy access of a sea of electronic health records ease the studies of individual treatment effect of a medicine on patients' health conditions.

Compared to data collected through randomized experiments, an observational dataset is often effortless to obtain and often comes with a large number of instances and an affluent set features. Meanwhile, the instances are often inherently connected with rich auxiliary structure information such as the social networks connecting multiple online users. Learning individual treatment effects from observational data requires us to handle confounding bias. We say there exists confounding bias when the correlation between the outcome and the treatment is distorted by the existence of confounders (a.k.a., the variables causally influence both the treatment and the outcome). For example, the poor socioeconomic status of an individual can limit her access to an expensive medicine and have negative impact on her health condition at the same time. Thus, without controlling the influence of the socioeconomic status, we may overestimate the treatment effect of the expensive medicine. Controlling the confounding bias is known as the main challenge of learning individual treatment effects from observational data [18,6]. To deal with these confounders, a vast majority of existing individual treatment estimation methods rely on the strong ignoralbility assumption [26,23,7,8,24] that all the confounders can be measured and are embedded in the set of observed features. As such, these methods often exploit the available features to mitigate the confounding bias. In the running example, most of existing efforts try to eliminate the influence of socioeconomic status on the chance to take the medicine and the health condition through controlling the impact of the related proxy variables such as annual income, age, and education. However, for observational data, given the fact that the causal relationships between variables are unknown, the strong ignoralbility assumption becomes untenable and it is likely to be unrealistic due to the existence of hidden confounders [18]. Recently, a series of methods are proposed to leverage representation learning to relax the strong ignorability assumption. Nonetheless, they still assumed that we were able to extract a set of latent features as the set of confounders from observational data using neural networks or factor models [14,27].

Despite the existing methods mentioned above, few have recognized the importance of the network structures connecting instances in the task of learning individual treatment effects. In fact, topology of instances is ubiquitous in various observational data such as a social network of patients, an electrical grid of power stations, and a spatial network of geometric objects, to name a few. In addition, when it is notoriously hard to measure some confounders, alternatively, we can capture their influence by incorporating the underlying network structures. Back to the running example, although the socioeconomic status of an individual is often difficult to be quantified by observed instance features, it can be implicitly represented by her social network patterns such as how many people are following her in the social network. Surprisingly, little attention has been

paid to utilizing network structure patterns to mitigate the confounding bias and then achieve precise estimation of individual treatment effects. To bridge the gap, in this work, we focus on leveraging network structural patterns along with observed features to minimize confounding bias in the task of individual treatment effect estimation. It worth noting that this work is significantly different from the existing studies on *spillover effect*, also known as *network entanglement* or *interference* [25,19], where the treatment on an instance may causally influence the outcomes of the connected units. In contrast, we focus on the situations where network structure can be exploited for controlling confounding bias. For example, a patient's network patterns reflect her socioeconomic status but her health condition is not likely to be causally affected by what treatments are assigned to her neighbors.

To exploit network structure patterns for controlling the hidden confounders, we propose the *network deconfounder*, a novel framework that captures the influence of hidden confounders. Fig. 1 illustrates the workflow of the proposed network deconfounder framework. In particular, the network deconfounder learns representations of confounders by mapping the original features as well as the network structure into a shared latent feature space. Then the representations of confounders can be exploited to control confounding bias and learn individual treatment effects from observational data.

Here, we summarize the main contributions of this work as follows:

– We formulate the novel problem of learning individual treatment effects from networked observational data.
– We propose a novel framework for learning individual treatment effects from networked observational data – *network deconfounder*, which controls confounding bias and estimates individual treatment effects given observational data with auxiliary network information.
– We perform extensive experiments to show that the proposed network deconfounder significantly outperforms the state-of-the-art methods for learning individual treatment effects across two semi-synthetic datasets based on real-world social network data.

2 Problem Statement

In this section, we start with an introduction of technical preliminaries and then formally present the problem of learning individual treatment effects from networked observational data.

First, we describe the notations used in this work. We denote a scalar, a vector, and a matrix with a lowercase letter (e.g., t), a boldface lowercase letter (e.g., $\mathbf{x}$), and a boldface uppercase letter (e.g., $\mathbf{A}$), respectively. Subscripts signify element indexes (e.g., $\mathbf{x}_i$ and $\mathbf{A}_{i,j}$). Superscripts of the a potential outcome variable denotes its corresponding treatment (e.g., y_i^t). Table 1 shows a summary of notations that are frequently referred to throughout this work.
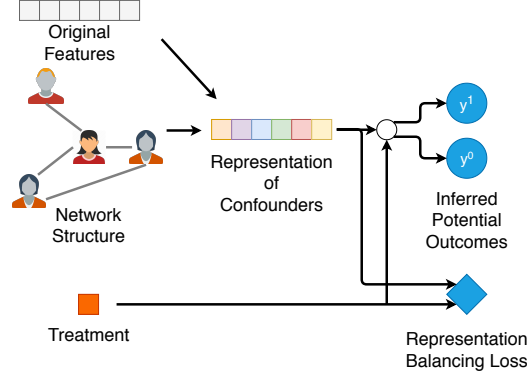
Fig. 1. The work flow of the proposed network deconfounder

Table 1. Notations

Notation	Definition and Description
$\mathbf{x}_i$	features of the i-th instance
t_i	observed treatment of the i-th instance
$\mathbf{A}$	adjacency matrix of the network
$\mathbf{h}_i$	representation of hidden confounders of instance i
y_i^F, y_i^{CF}	observed outcome and counterfactual outcome of the i-th instance
y_i^t	potential outcome of the i-th instance with treatment t
n	number of instances
m	dimension of the feature space
d	dimension of the representation space

Then we introduce networked observational data. In this work, we aim to learn individual treatment effects from networked observational data. Such data can be represented as $(\{\mathbf{x}_i, t_i, y_i\}_{i=1}^n, \mathbf{A})$ where $\mathbf{x}_i$, t_i and y_i denote the features, the observed treatment, and the observed (factual) outcome of the i-th instance, respectively. The symbol $\mathbf{A}$ signifies the adjacency matrix of the auxiliary network information among different data instances. Here, we assume that the network is undirected and all the edges share the same weight[1]. Therefore, $\mathbf{A} \in \{0,1\}^{n \times n}$ and $\mathbf{A}_{i,j} = \mathbf{A}_{j,i} = 1$ ($\mathbf{A}_{i,j} = \mathbf{A}_{j,i} = 0$) denotes that there is an (no) edge between the i-th instance and the j-th instance. We focus on the cases where the treatment variable takes binary values $t \in \{0,1\}$. Without loss of generality, we use $t_i = 1$ ($t_i = 0$) to imply that the i-th instance is under treatment (control). We also let the outcome variable be a scalar and take continuous real values as $y \in \mathbb{R}$. Then we introduce the background knowledge of learning individual treatment effects. To define individual treatment effect (ITE), we start with the definition of potential outcomes which is widely used in the causal inference literature [17,20]:

---

[1] This work can be directly applied to weighted undirected networks. It can also be extended to directed networks using the Graph Convolutional Neural Networks for directed networks [16]

Definition 1. *Potential Outcomes.* *Given an instance i and the treatment t, the potential outcome of i under treatment t, denoted by y_i^t, is defined as the value of y would have taken if the treatment of instance i had been set to t.*

Then we are able to provide mathematical formulation of ITE for the i-th instance in the setting of networked observational data as:

$$\tau_i = \tau(\mathbf{x}_i, \mathbf{A}) = \mathbb{E}[y_i^1 | \mathbf{x}_i, \mathbf{A}] - \mathbb{E}[y_i^0 | \mathbf{x}_i, \mathbf{A}] \tag{1}$$

Intuitively, ITE is defined as the expected outcome under treatment subtracted by the expected outcome under control, which reflects how much improvement of the outcome is caused by the treatment. It is worth noting that with the network information, we are able to go beyond the limited information provided by the features and distinguish two instances with the similar features but different network patterns in the task of learning individual treatment effects. With ITE defined, we can formulate the average treatment effect (ATE) by taking the average of ITE over the instances as: $ATE = \frac{1}{n} \sum_{i=1}^{n} \tau_i$. Finally, we formally present the definition of the problem of learning individual treatment effects from networked observational data as follows:

Definition 2. *Learning Individual Treatment Effects from Networked Observational Data.* *Given the networked observational data $(\{\mathbf{x}_i, t_i, y_i\}_{i=1}^{n}, \mathbf{A})$, we aim to develop a causal inference model to learn the individual treatment effects τ_i of each instance i.*

3 The Proposed Framework

3.1 Background

It is not difficult to find that as only one of the two potential outcomes can be observed, the main challenge of learning individual treatment effects is to infer the *counterfactual outcome* $y_i^{CF} = y_i^{1-t_i}$. In previous work [7,26,8,24], with the strong ignorability assumption, controlling observed features is often considered to be enough to eliminate confounding bias. Formally, strong ignorability can be defined as:

Definition 3. *Strong Ignorability.* *With strong ignorability, it is assumed: (1) The potential outcomes of an instance are independent of whether it receives treatment or control given its features. (2) For each instance the probability to get treated is larger than 0 and less than 1. Formally, given the set of possible values of features $\mathcal{X}$, we can write the strong ignorability as:*

$$y^1, y^0 \perp\!\!\!\perp t | \mathbf{x} \quad and \quad 1 > Pr(t = 1 | \mathbf{x}) > 0, \forall \mathbf{x} \in \mathcal{X}, t \in \{0, 1\}. \tag{2}$$

It implies $\mathbb{E}[y^t | \mathbf{x}] = \mathbb{E}[y | \mathbf{x}, t]$ due to the independence between the treatment and the potential outcomes, where y denotes the outcome resulting from the features $\mathbf{x}$ and the treatment t. Based on this observation, many existing methods boil
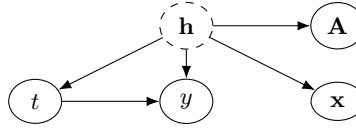
Fig. 2. The assumed causal graph: the network structure **A** along with the observed features **x** are proxy variables of the hidden confounders **h**, which can be utilized to learn representation of hidden confounders. The directed edges signify causal relationships, solid circles represent observed variables, and the dashed circle stand for hidden confounders.

down the task of learning ITE from observational data to learning a model $f : \mathcal{X} \times \{0,1\} \rightarrow \mathbb{R}$ to approximate $\mathbb{E}[y|\mathbf{x}, t]$.

However, in this work, we assume that there exist unobserved confounders. As a result, inferring counterfactual outcomes based on the features and the treatment alone would result in biased estimator ($\mathbb{E}[y|\mathbf{x}, t] \neq \mathbb{E}[y^t|\mathbf{x}]$) because dependencies between the treatment variable and the two potential outcome variables are introduced by hidden confounders.

3.2   Network Deconfounder

In this subsection, we propose the network deconfounder, a novel framework that addresses the challenges of learning individual treatment effects from networked observational data. Given the adjacency matrix **A**, feature vector **x**, the treatment t, and the outcome y, Fig. 2 shows the causal model assumed in the framework of network deconfounder. Instead of relying on the strong ignorability assumption, we assume that both the features and the network structure are two sets of proxy variables of the hidden confounders, which is a more practical assumption than strong ignorability. For example, although we cannot directly measure socioeconomic status of an individual, we can collect features such as age, job type, zip code, and the social network to describe her socioeconomic status. Based on this assumption, network deconfounder attempts to learn representations to approximate hidden confounders and estimate ITE from networked observational data simultaneously.

Unlike eliminating the confounding bias based on the features, leveraging the underlying network structure for controlling confounding bias raises special challenges: (1) instances are inherently interconnected with each other through the network structure and hence they are not independent identically distributed (i.i.d.), (2) the adjacency matrix of a network is often high-dimensional ($\mathbf{A} \in \{0,1\}^{n \times n}$) and can be very sparse.

To tackle these special challenges of controlling confounding bias with network structure information, we propose the network deconfounder framework. The task can be divided into two steps. First, we aim to learn representations of hidden confounders by mapping the features and the network structure into the representation space. Then an output function is learned to infer potential

outcomes based on the treatment and the representation of hidden confounders. Then we present how the two tasks are accomplished by the network deconfounder.

Learning Representation of Confounders. The first component of network deconfounder is a representation learning function g that can map the features and the underlying network into the latent space of hidden confounders, which can be formulated as $g : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$. We parameterize the g function using Graph Convolutional Neural Networks (GCN) [4,11], whose effectiveness have been verified in various machine learning tasks across different types of networked data. To the best of our knowledge, this is the first work introducing GCN to the task of learning individual treatment effects. In particular, the representation of confounders of the i-th instance is learned through GCN layers. Here, we describe the representation learning function g with a single GCN layer for the simplicity of notation. The representation learning function g is parameterized as:

$$\mathbf{h}_i = g(\mathbf{x}_i, \mathbf{A}) = \sigma((\hat{\mathbf{A}}\mathbf{X})_i \mathbf{U}), \tag{3}$$

where $\hat{\mathbf{A}}$ denotes the normalized adjacency matrix, $(\hat{\mathbf{A}}\mathbf{X})_i$ signifies the i-th row of the matrix product $\hat{\mathbf{A}}\mathbf{X}$, $\mathbf{U} \in \mathbb{R}^{m \times d}$ represents the weight matrix, and σ stands for the ReLU activation function [5]. Specifically, with $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$ and $\tilde{\mathbf{D}}_{j,j} = \sum_j \tilde{\mathbf{A}}_{j,j}$ defined, the normalized adjacency matrix $\hat{\mathbf{A}}$ can be calculated using the renormalization trick [11]:

$$\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \tag{4}$$

We can compute $\hat{\mathbf{A}}$ in a pre-processing step to avoid repeating the computation. Then the weight matrix $\mathbf{U} \in \mathbb{R}^{m \times d}$ along with the ReLU activation function maps the input signal into the low-dimensional representation space. It is worth noting that more than one GCN layers can be stacked to catch the non-linearity between hidden confounders and the input data.

Inferring Potential Outcomes. Then we introduce the second component of network deconfounder, the output function $f : \mathbb{R}^d \times \{0, 1\} \to \mathbb{R}$, which maps the representation of hidden confounders as well as the treatment to the corresponding potential outcome. With $\mathbf{h}_i \in \mathbb{R}^d$ denoting the representation of the hidden confounders of the i-th instance and $t \in \{0, 1\}$ signifying the treatment, to infer the corresponding potential outcome, we parameterize the output function f as:

$$f(\mathbf{h}_i, t) = \begin{cases} f_1(\mathbf{h}_i) \text{ if } t = 1 \\ f_0(\mathbf{h}_i) \text{ if } t = 0 \end{cases}, \tag{5}$$

where f_1 and f_0 are the output functions for treatment $t = 1$ and $t = 0$. Specifically, we parameterize the output functions f_1 and f_0 using L fully connected layers and an output regression layer as:

$$f_1 = \mathbf{w}^1 \sigma(\mathbf{W}_L^1 ... \sigma(\mathbf{W}_1^1 \mathbf{h}_i)), \quad f_0 = \mathbf{w}^0 \sigma(\mathbf{W}_L^0 ... \sigma(\mathbf{W}_1^0 \mathbf{h}_i)), \tag{6}$$

where $\mathbf{h}_i$ is the representation of hidden confounders (output of the g function) of the i-th instance, $\{\mathbf{W}_l^t\}, l = 1, ..., L$ denote the weight matrices of the fully

connected layers, and $\mathbf{w}^t$ is the weight for the regression prediction layers. The bias terms of the fully connected layers and the output regression layer are dropped for simplicity of notation. We can either set $t = t_i$ to infer the observed factual outcome y_i^{CF} or $t = 1 - t_i$ to estimate the counterfactual outcome.

With the two components of network deconfounder formulated, given the features of the i-th instance $\mathbf{x}_i$, the treatment t, and the adjacency matrix $\mathbf{A}$, we can infer the potential outcome as:

$$\hat{y}_i^t = f(g(\mathbf{x}_i, \mathbf{A}), t), \tag{7}$$

where $\hat{y}_i^t$ denotes the inferred potential outcome of instance i corresponding to treatment t by the network deconfounder framework.

Objective Function. Then we introduce three essential components of the loss function for the proposed network confounder.

Factual Outcome Inference. First, we aim to minimize the error of inferring the observed factual outcomes. This leads to the first component of the loss function, the mean squared error on the inferred factual outcomes:

$$\frac{1}{n} \sum_{i=1}^{N} (\hat{y}_i^{t_i} - y_i)^2. \tag{8}$$

Representation Balancing. Minimizing the error on the factual outcomes does not necessarily mean the error on counterfactual outcomes is also minimized. In other words, in the problem of learning ITE from observational data, we essentially confront the challenge of domain adaptation [8,24]. In particular, the network deconfounder would be trained on the conditional distribution of factual outcomes $Pr(y_i^F|\mathbf{x}_i, \mathbf{A}, t_i)$ but the task is to infer the conditional distribution of counterfactual outcomes $Pr(y_i^{CF}|\mathbf{x}_i, \mathbf{A}, 1 - t_i)$. In [24, Lemma 1.], the authors have shown that the inference error on the counterfactual outcomes is upper-bounded by a weighted sum of (1) the error on the factual outcomes; and (2) the integral probability metrics (IPM) measuring the difference between the distributions the treated instances and the controlled instances in terms of their confounder representations. Therefore, we also aim to minimize the IPM measuring the divergence between the distributions of the two treatment groups regarding their representations of hidden confounders. With $P(\mathbf{h}) = Pr(\mathbf{h}|t_i = 1)$ and $Q(\mathbf{h}) = Pr(\mathbf{h}|t_i = 0)$ being the empirical distributions of representation of hidden confounders, we let $\rho_{\mathcal{Z}}(P, Q)$ denote the IPM defined in the functional space $\mathcal{Z}$ measuring the distance between the two distributions of representations of hidden confounders. Assuming that $\mathcal{Z}$ denotes the set of 1-Lipschitz functions, the IPM reduces to the Wasserstein-1 distance which is defined as:

$$\rho_{\mathcal{Z}}(P, Q) = \inf_{k \in \mathcal{K}} \int_{\mathbf{h} \in \{\mathbf{h}_i\}_{i:t_i=1}} ||k(\mathbf{h}) - \mathbf{h}|| P(\mathbf{h}) d\mathbf{h} \tag{9}$$

where $\mathcal{K} = \{k|k : \mathbb{R}^d \to \mathbb{R}^d \ s.t. \ Q(k(\mathbf{h})) = P(\mathbf{h})\}$ denotes the set of push-forward functions that can transform the representation distribution of the

treated P to that of the controlled Q. By minimizing $\alpha\rho_\mathcal{Z}(P,Q)$, we approximately minimize the gap between the distributions of representation of latent confounders, where $\alpha \geq 0$ signifies the hyperparameter controlling the trade-off between representation balancing and the other terms. We use the efficient approximation algorithm proposed by [3] to compute the Wasserstein distance (Eq. (9)) and its gradients for training the model.

ℓ_2 *Regularization.* Third, we let $\boldsymbol{\theta}$ signify the vector of the model parameters of network deconfounder. Then a squared ℓ_2 norm regularization term on the model parameters - $\lambda||\boldsymbol{\theta}||_2^2$, is added to mitigate the overfitting problem, where $\lambda \geq 0$ denotes the hyperparamter controlling the trade-off between the ℓ_2 regularization and the other two terms.

Formally, We present the objective function of the network deconfounder as:

$$\mathcal{L}(\{\mathbf{x}_i, t_i, y_i\}_{i=1}^n, \mathbf{A}) = \frac{1}{n}\sum_{i=1}^{N} w_i(\hat{y}_i^{t_i} - y_i)^2 + \alpha\rho_\mathcal{Z}(P,Q) + \lambda||\boldsymbol{\theta}||_2^2, \qquad (10)$$

4 Experiments

4.1 Dataset Description

It is notoriously hard to obtain ground truth for ITE because for a vast majority of cases, we can only observe one of the potential outcomes. For example, a patient can only choose to take the medicine or not to take it, but not both. To resolve this problem, we follow the existing literature [8,24,14,22] to create semi-synthetic datasets. In particular, we introduce two new networked observational datasets which have ground truth features, network structures, synthetic treatments, and outcomes for the task of learning ITE from networked observational data in the presence of hidden confounders.

BlogCatalog. BlogCatalog[2] is an online community where users can post blogs. Each instance is a blogger. Each edge signifies the social relationship between two bloggers. The features are the keywords in bloggers' blog descriptions. We extend the BlogCatalog dataset used in [12,13] by synthesizing (a) the outcome – the opinions of readers on each blogger; and (b) the treatment – whether content from a blogger is read more on mobile devices or on desktops. Similar to the News dataset used in previous work [8,23,22], we make the following assumptions: (1) Readers either read on mobile devices or desktops. We say a blogger get treated (controlled) if her blogs are read more on mobile devices (desktops). (2) Readers prefer to read some topics from mobile devices, others from desktops. (3) A blogger and her neighbors' topics influence treatment. (4) A blogger and her neighbors' topics causally affect readers' opinions on them. Here, we aim to study the causal effect of being read more on mobile devices on readers' opinions of each blogger. To synthesize treatments and outcomes in accordance to the assumptions mentioned above, we train a topic model on a large set of documents. Then two centroids in the topic space are defined as: (i)

---

[2] https://www.blogcatalog.com/

we randomly sample a blogger and let the topic distribution of her description be the centroid of the treated instances, denoted by r_1^c. (ii) The centroid of the controlled, r_0^c, is selected to be the centroid of the topic distribution of all the bloggers' description. Then we introduce how the treatments and outcomes are synthesized based on the similarity between the topic distribution of a blogger's description and the two centroids. With $r(\mathbf{x}_i)$ denoting the topic distribution of the i-th blogger's description, we model the preference of the readers of the i-th blogger's content as:

$$
Pr(t = 1|\mathbf{x}_i, \mathbf{A}) = \frac{\exp(p_1^i)}{\exp(p_1^i) + \exp(p_0^i)}
$$
$$
p_1^i = \kappa_1 r(\mathbf{x}_i)^T r_1^c + \kappa_2 \sum_{j \in \mathcal{N}(i)} r(\mathbf{x}_j)^T r_1^c = \kappa_1 r(\mathbf{x}_i)^T r_1^c + \kappa_2 (\mathbf{A}r(\mathbf{x}_j))^T r_1^c \quad (11)
$$
$$
p_0^i = \kappa_1 r(\mathbf{x}_i)^T r_0^c + \kappa_2 \sum_{j \in \mathcal{N}(i)} r(\mathbf{x}_j)^T r_0^c = \kappa_1 r(\mathbf{x}_i)^T r_0^c + \kappa_2 (\mathbf{A}r(\mathbf{x}_j))^T r_0^c
$$

where $\kappa_1, \kappa_2 \geq 0$ signifies the strength of the confounding bias resulting from a blogger's topics and her neighbors' topics. When $\kappa_1 = 0, \kappa_2 = 0$ the treatment assignment is random and the greater the value κ is, the more significant the bias of device preference is. Then factual outcome of the i-th blogger is simulated as:

$$
y^F(\mathbf{x}_i) = C(p_0^i + t_i p_1^i) + \epsilon, \quad (12)
$$

where C is a scaling factor and the noise is sampled as $\epsilon \sim \mathcal{N}(0, 1)$. In this work, we set $C = 5, \kappa_1 = 10, \kappa_2 \in \{0.5, 1, 2\}$. 50 LDA topics are learned from the training corpus. Then we reduce the vocabulary by taking the union of the most probable 100 words from each topic. By doing this, we end up with 2,173 bag-of-word features. We perform the above mentioned simulation 10 times for each setting of κ_2. Figure 3 shows the distribution of topics in one of the simulations which is projected to two-dimensional space using TSNE [15]. We observe that there are more treated instances (red dots) near the centroid r_1^c (green diamond) and more control instances (blue dots) close to the centroid r_0^c (yellow diamond). In addition, a significant shift from the centroids can be perceived which shows the impact of the network structure.

Flickr. Flickr[3] is an image and video sharing service. Each instance refers to a user and each edge represents the social relationship between two users. The features of each user represent a list of tags of interest. We adopt the same settings and assumptions as we do for the BlogCatalog dataset. Thus, we also study the individual-level causal effects of being viewed on mobile devices on readers' opinions on the user. In particular, we also learn 50 topics from the training corpus using LDA and concatenate the top 25 words of each topic. Thus, we reduce the data dimension to 1,210. We maintain the same settings of parameters as the BlogCatalog dataset ($C = 5, \kappa_1 = 10$ and $\kappa_2 \in \{0.5, 1, 2\}$).
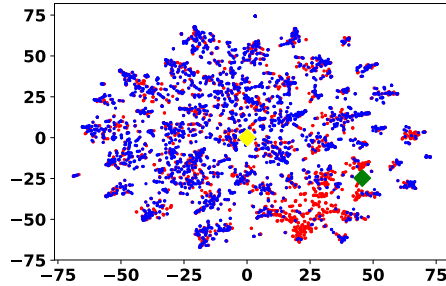
---

[3] https://www.flickr.com

Fig. 3. Distribution of treated (red) and control (blue) instances in the topic space. The green and yellow diamonds signify the centroids r_1^c and r_0^c.

Table 2. Dataset Description

Dataset Name	# Instances	# Edges	# Features	κ_2	Average ATE	STD ATE
BlogCatalog	5,196	173,468	8,189	0.5	4.366	0.553
				1	7.446	0.759
				2	13.534	2.309
Flickr	7,575	239,738	12,047	0.5	6.672	3.068
				1	8.487	3.372
				2	20.546	5.718

In Table 2, we present a summary of the statistics of the semi-synthetic datasets described in this subsection. The average and standard deviation of ATE are calculated over the 10 runs under each setting of parameters.

4.2 Experimental Settings

Following the original implementation of GCN [11][4], we train the model with all the training instances along with the complete adjacency matrix. ADAM [10] is the optimizer we use to minimize the objective function of the network deconfounder (Eq. (10)). We randomly sample 60% and 20% of the instances as the training set and validation set and let the remaining be the test set. We perform 10 times of random sampling for each simulation of the datasets and report the average results. Grid search is applied to find the optimal combination of hyperparameters for the network deconfounder. In particular, we search learning rate in $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$, the number of output layers in $\{1, 2, 3\}$, dimensionality of the outputs of the GCN layers and the number of hidden units of the fully connected layers in $\{50, 100, 200\}$, α and λ in $\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. For the baseline methods, we adopt the default setting of hyperparameters[5].

---

[4] https://github.com/tkipf/gcn

[5] As a reproducible research paper, the code of this work can be find at https://github.com/rguo12/network-deconfounder/

Then, we describe the baselines methods which are the state-of-the-art methods for learning ITE from observational data:

Counterfactual Regression (CFR) [24]. CFR is based on the strong ignorability assumption. It learns representations of confounders by mapping the original features into a latent space. It minimizes the factual outcome inference error and the representation balancing error. Following [24], two types of representation balancing error are considered: Wasserstein-1 distance (CFR-Wass) and maximum mean discrepancy (CFR-MMD).

Treatment-agnostic Representation Networks (TARNet) [24]. TARnet is a variant of CFR without the representation balancing penalty term.

Causal Effect Variational Autoencoder (CEVAE) [14]. CEVAE is a deep latent-variable model developed for learning ITE. It learns representation of confounders as Gaussian distributions through propagating information from original features, observed treatments, and factual outcomes.

Causal Forest [26]. Causal Forest is an extension of Breiman's random forest [1] for learning ITE. It works with the strong ignorability assumption.

Bayesian Additive Regression Trees (BART) [7]. BART is a Bayesian regression tree based ensemble model which is widely used in the literature of learning ITE. It is also based on the strong ignorability assumption.

Two widely used evaluation metrics, the Rooted Precision in Estimation of Heterogeneous Effect ($\sqrt{\epsilon_{PEHE}}$) and Mean Absolute Error on ATE (ϵ_{ATE}), are adopted by this work. Formally, they are defined as:

$$\sqrt{\epsilon_{PEHE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{\tau}_i - \tau_i)^2}, \ \epsilon_{ATE} = |\frac{1}{n}\sum_{i=1}^{n}(\hat{\tau}_i) - \frac{1}{n}\sum_{i=1}^{n}(\tau_i)|, \qquad (13)$$

where $\hat{\tau}_i = \hat{y}_i^1 - \hat{y}_i^0$ and $\tau_i = y_i^1 - y_i^0$ denote the inferred ITE and the ground truth ITE for the i-th instance.

4.3   Results

Effectiveness. First, we compare the effectiveness of the proposed network deconfounder with the aforementioned state-of-the-art methods. Table 3 summarizes the empirical results evaluated on the BlogCatalog and Flickr datasets with $C = 5, \kappa_1 = 10$ and $\kappa_2 \in \{0.5, 1, 2\}$. We summarize the observations from these experimental results as follows:

– The proposed network deconfounder consistently outperforms the state-of-the-art baseline methods on the semi-synthetic datasets with treatments and outcomes generated under various settings. We perform one-tailed T-test to verify the statistical significance and the results indicate that the network deconfounder achieves significantly better estimations on individual treatment effects with a significant level of 0.05.
– Benefiting from the capability to capture the patterns of hidden confounders from the network structure, the network deconfounder suffers the least when the influence of hidden confounders grows (from $\kappa_2 = 0.5$ to $\kappa_2 = 2$) in terms of the increase in the error metrics $\sqrt{\epsilon_{PEHE}}$ and ϵ_{ATE}.

Table 3. Experimental Results.

BlogCatalog						
κ_2	0.5		1		2	
	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}
NetDeconf (ours)	**4.532**	**0.979**	**4.597**	**0.984**	**9.532**	**2.130**
CFR-Wass	10.904	4.257	11.644	5.107	34.848	13.053
CFR-MMD	11.536	4.127	12.332	5.345	34.654	13.785
TARNet	11.570	4.228	13.561	8.170	34.420	13.122
CEVAE	7.481	1.279	10.387	1.998	24.215	5.566
Causal Forest	7.456	1.261	7.805	1.763	19.271	4.050
BART	4.808	2.680	5.770	2.278	11.608	6.418
Flickr						
κ_2	0.5		1		2	
	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}
NetDeconf (ours)	**4.286**	**0.805**	**5.789**	**1.359**	**9.817**	**2.700**
CFR-Wass	13.846	3.507	27.514	5.192	53.454	13.269
CFR-MMD	13.539	3.350	27.679	5.416	53.863	12.115
TARNet	14.329	3.389	28.466	5.978	55.066	13.105
CEVAE	12.099	1.732	22.496	4.415	42.985	5.393
Causal Forest	8.104	1.359	14.636	3.545	26.702	4.324
BART	4.907	2.323	9.517	6.548	13.155	9.643

Parameter Study. Then we investigate how the values of the two important hyperparameters α and λ affect the performance of the network deconfounder. Here, we fix the learning rate to be 10^{-2}, the number of epochs to be 200, the number of GCN layers and the number of output layers to be 2, the number of hidden units and the dimensionality of the representations to be 100. Then we vary α and λ in the range of $\{1, 10^{-2}, 10^{-4}, 10^{-6}\}$. The results are shown in Fig. 4. Due to the space limit, we only report the results on the BlogCatalog dataset with $\kappa_2 = 1$ in terms of both error metrics $\sqrt{\epsilon_{PEHE}}$ and ϵ_{ATE}. Based on the observations that the $\sqrt{\epsilon_{PEHE}}$ and ϵ_{ATE} do not change significantly when $\alpha \leq 10^{-6}$ and $10^{-6} \leq \lambda \leq 1$, we can conclude that the performance of the network deconfounder is not sensitive to both parameters α and λ. However, when α is too large ($\alpha > 0.01$), the performance of the network deconfounder degrades. This is because when α is too large, the objective function would emphasize the importance of balancing the representation of confounders of the two treatment groups too much and sacrifice the accuracy on inferring ITE.

5   Related Work

In this section, we present an introduction of two directions of related work: learning ITE from observational data and graph convolutional neural networks. **Learning Individual Treatment Effects** Recently, the fundamental problem of learning ITE from observational data has attracted considerable attention in many high-impact research areas including statistics, machine learning, and data
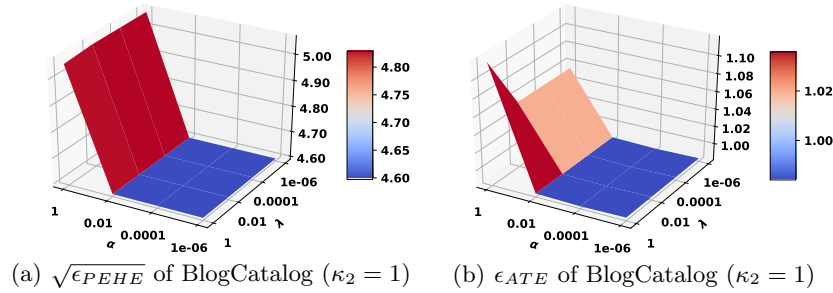
(a) $\sqrt{\epsilon_{PEHE}}$ of BlogCatalog ($\kappa_2 = 1$)   (b) ϵ_{ATE} of BlogCatalog ($\kappa_2 = 1$)

Fig. 4. Impact of α and λ on the performance of the proposed framework.

mining. Hill [7] proposed to apply BART to learn ITE because it requires little hyperparameter tuning. Wager and Athey [26] proposed the Causal Forest, which extended Breiman's random forest [1] for learning ITE. In [8,24], two methods are proposed to learn representations of confounders using neural networks. The work also showed that balancing the distributions of confounders of the treated and controlled instances in the representation space can improve the performance in the task of learning ITE. However, the methods mentioned above rely on the strong ignorability assumption to handle the hidden confounders, which is often untenbale and can be unrealistic in real-world observational datasets. Recently, Louizos et al. [14] proposed to consider observed features as proxy variables of real confounders and use a deep latent-variable model to learn representation of confounders through variational inference. However, none of the previous work utilized the network structure to capture patterns of hidden confounders.

Graph Convolutional Neural Networks. Previous work on Graph Convolutional Neural Networks (GCN) mainly focused on development of spatially localized[6] and computationally efficient convolutional filters for network data. Bruna et al. [2] proposed to use the first-order graph Laplacian matrix as the filter in the spectrum domain. However, this filter has a large number of trainable parameters and its the spatial locality is not guaranteed. In [4], Defferrard et al. introduced a more efficient and properly localized filter for graph convolution. This filter is parameterized as l-th order polynomials of the graph Laplacian matrix to ensure the locality. Then the polynomials are approximated by its Chebyshev expansion to reduce the computational cost. Then, Kipf and Welling [11] proposed the renormalization trick to further improve the computational efficiency of GCN. Recently, GCN has also been applied to solve a variety of problems on network data such as recommendation systems [28] and knowledge graph [21]. Different from the existing work, this paper is the first work exploiting GCN for the causal inference problem - learning ITE from observational data.

---

[6] Here, spatial locality refers to the constraint that information of a node only propagates to its l-hop neighbors.

6   Conclusion

New challenges are presented by the prevalence of networked observational data for learning individual treatment effects. In this work, we study a novel problem, learning individual treatment effects from networked observational data. As the underlying network structure could capture useful information of hidden confounders, we propose the network deconfounder framework, which leverages the network structural patterns along with original features for learning better representations of confounders. Empirically, we perform extensive experiments across multiple real-world datasets. Results show that the network deconfounder learns better representation of confounders than the state-of-the-art methods.

Here, we also introduce two most interesting directions of future work. First, we are interested in leveraging other types of structure between instances for learning ITE from observational data. For example, temporal dependencies can also be utilized to capture patterns of hidden confounders. Second, in this work, focus on static network structure. But the real-world networks can evolve over time. Hence, we would like to investigate how to exploit dynamics in evolving networks for learning ITE.

References

1. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
2. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
3. Cuturi, M., Doucet, A.: Fast computation of wasserstein barycenters. In: International Conference on Machine Learning. pp. 685–693 (2014)
4. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems. pp. 3844–3852 (2016)
5. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 315–323 (2011)
6. Guo, R., Cheng, L., Li, J., Hahn, P.R., Liu, H.: A survey of learning causality with data: Problems and methods. arXiv preprint arXiv:1809.09337 (2018)
7. Hill, J.L.: Bayesian nonparametric modeling for causal inference. Journal of Computational and Graphical Statistics **20**(1), 217–240 (2011)
8. Johansson, F., Shalit, U., Sontag, D.: Learning representations for counterfactual inference. In: International conference on machine learning. pp. 3020–3029 (2016)
9. Kallus, N., Zhou, A.: Confounding-robust policy improvement. In: Advances in Neural Information Processing Systems. pp. 9289–9299 (2018)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
12. Li, J., Hu, X., Tang, J., Liu, H.: Unsupervised streaming feature selection in social media. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 1041–1050. ACM (2015)

13. Li, J., Hu, X., Wu, L., Liu, H.: Robust unsupervised feature selection on networked data. In: Proceedings of the 2016 SIAM International Conference on Data Mining. pp. 387–395. SIAM (2016)
14. Louizos, C., Shalit, U., Mooij, J.M., Sontag, D., Zemel, R., Welling, M.: Causal effect inference with deep latent-variable models. In: Advances in Neural Information Processing Systems. pp. 6446–6456 (2017)
15. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008)
16. Monti, F., Otness, K., Bronstein, M.M.: Motifnet: a motif-based graph convolutional network for directed graphs. In: 2018 IEEE Data Science Workshop (DSW). pp. 225–228. IEEE (2018)
17. Neyman, J.S.: On the application of probability theory to agricultural experiments. essay on principles. section 9.(tlanslated and edited by dm dabrowska and tp speed, statistical science (1990), 5, 465-480). Annals of Agricultural Sciences **10**, 1–51 (1923)
18. Pearl, J., et al.: Causal inference in statistics: An overview. Statistics surveys **3**, 96–146 (2009)
19. Rakesh, V., Guo, R., Moraffah, R., Agarwal, N., Liu, H.: Linked causal variational autoencoder for inferring paired spillover effects. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 1679–1682. ACM (2018)
20. Rubin, D.B.: Bayesian inference for causal effects: The role of randomization. The Annals of statistics pp. 34–58 (1978)
21. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference. pp. 593–607. Springer (2018)
22. Schwab, P., Linhardt, L., Bauer, S., Buhmann, J.M., Karlen, W.: Learning counterfactual representations for estimating individual dose-response curves. arXiv preprint arXiv:1902.00981 (2019)
23. Schwab, P., Linhardt, L., Karlen, W.: Perfect match: A simple method for learning representations for counterfactual inference with neural networks. arXiv preprint arXiv:1810.00656 (2018)
24. Shalit, U., Johansson, F.D., Sontag, D.: Estimating individual treatment effect: generalization bounds and algorithms. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 3076–3085. JMLR. org (2017)
25. Toulis, P., Volfovsky, A., Airoldi, E.M.: Propensity score methodology in the presence of network entanglement between treatments. arXiv preprint arXiv:1801.07310 (2018)
26. Wager, S., Athey, S.: Estimation and inference of heterogeneous treatment effects using random forests. Journal of the American Statistical Association **113**(523), 1228–1242 (2018)
27. Wang, Y., Blei, D.M.: The blessings of multiple causes. arXiv preprint arXiv:1805.06826 (2018)
28. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 974–983. ACM (2018)