Deep Counterfactual Networks with Propensity-Dropout

Ahmed M. Alaa, [1] **Michael Weisz,** [2] **Mihaela van der Schaar** [1 2 3]

Abstract

We propose a novel approach for inferring the individualized causal effects of a treatment (intervention) from observational data. Our approach conceptualizes causal inference as a *multitask* learning problem; we model a subject's potential outcomes using a deep multitask network with a set of shared layers among the factual and counterfactual outcomes, and a set of outcome-specific layers. The impact of selection bias in the observational data is alleviated via a *propensity-dropout* regularization scheme, in which the network is thinned for every training example via a dropout probability that depends on the associated propensity score. The network is trained in alternating phases, where in each phase we use the training examples of one of the two potential outcomes (treated and control populations) to update the weights of the shared layers and the respective outcome-specific layers. Experiments conducted on data based on a real-world observational study show that our algorithm outperforms the state-of-the-art.

1. Introduction

The problem of inferring individualized treatment effects from observational datasets is a fundamental problem in many domains such as precision medicine (Shalit et al., 2017), econometrics (Abadie & Imbens, 2016), social sciences (Athey & Imbens, 2016), and computational advertising (Bottou et al., 2013). A lot of attention has been recently devoted to this problem due to the recent availability of electronic health record (EHR) data in most of the hospitals in the US (Charles et al., 2015), which paved the way for using machine learning to estimate the individual-level causal effects of treatments from observational EHR data as an alternative to the expensive clinical trials.

[*]Equal contribution [1]University of California, Los Angeles, US. [2]University of Oxford, UK. [3]Alan Turing Institute, UK.. Correspondence to: Ahmed M. Alaa <ahmedmalaa@ucla.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

A typical observational dataset comprises a subject's features, a treatment assignment indicator (i.e. whether the subject received the treatment), and a "factual outcome" corresponding to the subject's response. Estimating the effect of a treatment for any given subject requires inferring her "counterfactual outcome", i.e. her response had she experienced a different treatment assignment. Classical works have focused on estimating "average" treatment effects through variants of *propensity score matching* (Rubin, 2011; Austin, 2011; Abadie & Imbens, 2016; Rosenbaum & Rubin, 1983; Rubin, 1973). More recent works tackled the problem of estimating "individualized" treatment effects using representation learning (Johansson et al., 2016; Shalit et al., 2017), Bayesian inference (Hill, 2012), and standard supervised learning (Wager & Athey, 2015).

In this paper, we propose a novel approach for individual-level causal inference that casts the problem in a *multitask* learning framework. In particular, we model a subject's potential (factual and counterfactual) outcomes using a deep multitask network with a set of layers that are shared across the two outcomes, and a set of idiosyncratic layers for each outcome (see Fig. 1). We handle selection bias in the observational data via a novel *propensity-dropout* regularization scheme, in which the network is thinned for every subject via a dropout probability that depends on the subject's propensity score. Our model can provide individualized measures of uncertainty in the estimated treatment effect by applying Monte Carlo propensity-dropout at inference time (Gal & Ghahramani, 2016).

Learning is carried out through an *alternate training* approach in which we divided the observational data into a "treated batch" and a "control batch", and then update the weights of the shared and idiosyncratic layers for each batch separately in an alternating fashion. We conclude the paper by conducting a set of experiments on data based on a real-world observational study showing that our algorithm outperforms the state-of-the-art.

2. Problem Formulation

Throughout this paper, we adopt Rubin's *potential outcomes* model (Rubin, 2011; 1973; Rosenbaum & Rubin, 1983). That is, we consider a population of subjects where

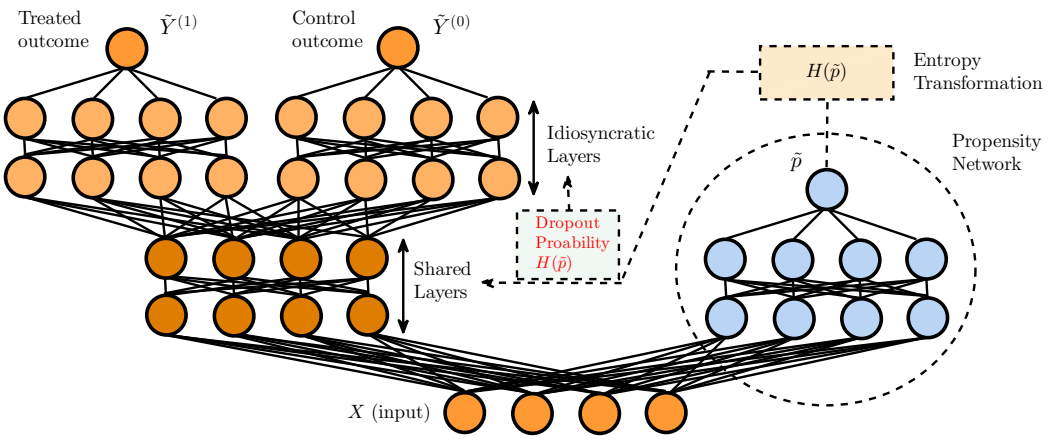

Figure 1. Depiction of the network architecture for our model with $L_p = L_s = L_{i,0} = L_{i,1} = 2$.

each subject i is associated with a d-dimensional *feature* $X_i \in \mathcal{X}$, and two *potential outcomes* $Y_i^{(1)}, Y_i^{(0)} \in \mathbb{R}$ that are drawn from a distribution $(Y_i^{(1)}, Y_i^{(0)})|X_i = x \sim \mathbb{P}(.|X_i = x)$. The *individualized treatment effect* for a subject i with a feature $X_i = x$ is defined as

$$T(x) = \mathbb{E}[Y_i^{(1)} - Y_i^{(0)} \mid X_i = x]. \tag{1}$$

Our main goal is to estimate the function $T(x)$ from an observational dataset $\mathcal{D}$ comprising n independent samples of the tuple $\{X_i, W_i, Y_i^{(W_i)}\}$, where $Y_i^{(W_i)}$ and $Y_i^{(1-W_i)}$ are the *factual* and the *counterfactual* outcomes, respectively, and $W_i \in \{0, 1\}$ is a treatment assignment indicator that indicates whether or not subject i has received the treatment. Treatment assignments are random variables that depend on the subjects' features, i.e. $W_i \not\perp\!\!\!\perp X_i$. The quantity $p(x) = \mathbb{P}(W_i = 1|X_i = x)$ is known as the *propensity score* of subject i (Rosenbaum & Rubin, 1983; Rubin, 1973), and it reflects the underlying policy for assigning the treatment to subjects.

3. Model Description

Most previous works adopted a *direct modeling* approach for estimating $T(x)$ in which a single-output regression model $f(.,.) : \mathcal{X} \times \{0, 1\} \to \mathbb{R}$ that treats the treatment assignment $W_i \in \{0, 1\}$ as an input feature is used to estimate the two potential outcomes, i.e. $\tilde{T}(x) = f(x, 1) - f(x, 0)$ (Shalit et al., 2017; Wager & Athey, 2015; Xu et al., 2016; Hill, 2012; Johansson et al., 2016). Such a modeling approach clearly limits the interaction between the treatment assignment and the subjects' features, especially in high dimensional feature spaces, which can lead to serious consequences in settings where the response surfaces $\mathbb{E}[Y_i^{(1)} \mid X_i = x]$ and $\mathbb{E}[Y_i^{(0)} \mid X_i = x]$ have drastically different properties (i.e. different relevant features and different nature for the interactions among the covariates). Another less popular modeling approach is the "virtual twin" approach, which simply fits a separate regression model for each of the treated and control populations (Lu et al., 2017). Such an approach sacrifices statistical efficiency for the sake of the modeling flexibility ensured by fitting separate models for the two potential outcomes. In the following Subsections, we propose a novel approach that ensures both modeling flexibility and statistical efficiency, and in addition, is capable of dealing with selection bias.

3.1. Multitask Networks

We propose a neural network model for estimating the individualized treatment effect $T(x)$ by learning a *shared representation* for the two potential outcomes. Our model, depicted in Fig. 1, comprises a *propensity network* (right) and a *potential outcomes network* (left). The propensity network is a standard feed-forward network with L_p layers and $h_p^{(l)}$ hidden units in the l^{th} layer, and is trained separately to estimate the propensity score $p(x)$ via the samples (X_i, W_i) in $\mathcal{D}$. The potential outcomes network is a *multitask network* (Collobert & Weston, 2008) that comprises L_s *shared layers* (with $h_s^{(l)}$ hidden units in the l^{th} shared layer), and $L_{i,j}$ *idiosyncratic layers* (with $h_{i,j}^{(l)}$ hidden units in the l^{th} layer) for potential outcome $j \in \{0, 1\}$.

The potential outcomes network approaches the problem of learning the two response surfaces $\mathbb{E}[Y_i^{(1)} \mid X_i = x]$ and $\mathbb{E}[Y_i^{(0)} \mid X_i = x]$ via a multitask learning framework. That is, we view the potential outcomes as two separate, but related, learning tasks. The observational dataset $\mathcal{D}$ is thus viewed as comprising two batches of task-specific data: a *treated batch* $\mathcal{D}^{(1)} = \{i \in \mathcal{D} : W_i = 1\}$ comprising all treated subjects, and a *control batch* $\mathcal{D}^{(0)} = \{i \in \mathcal{D} :$

$W_i = 0\}$ comprising all untreated subjects. The treatment assignment W_i is viewed as equivalent to the *task index* in conventional multitask learning. The shared layers in the potential outcomes network ensure *statistical efficiency* as they use the data in both $\mathcal{D}^{(0)}$ and $\mathcal{D}^{(1)}$ to capture the "commonality" between the two learning tasks. The idiosyncratic layers for task (outcome) j ensure *modeling flexibility* as they only use the data in $\mathcal{D}^{(j)}$ to capture the peculiarities of the response surface $\mathbb{E}[Y_i^{(j)} \mid X_i = x]$. Since the feature distributions in $\mathcal{D}^{(0)}$ and $\mathcal{D}^{(1)}$ are different (due to the selection bias), we use the outputs of the propensity network to regularize the potential outcomes network.

3.2. Propensity-Dropout

In order to ameliorate the impact of selection bias, we use the outputs of the propensity network to regularize the potential outcomes network. We do so through a dropout scheme that we call *propensity-dropout*. In propensity-dropout, the dropout procedure is applied in such a way that it assigns "simple models" to subjects with very high or very low propensity scores ($p(x)$ close to 0 or 1), and more "complex models" to subjects with balanced propensity scores ($p(x)$ close to 0.5). That is, we use a different dropout probability for each training example depending on the associated score: the dropout probability is higher for subjects with features that belong in a region of poor treatment assignment overlap in the feature space. We implement the propensity-dropout by using the following formula for the dropout probability:

$$\text{Dropout Probability}(x) = 1 - \frac{\gamma}{2} - \frac{1}{2} H(\tilde{p}(x)), \qquad (2)$$

where $0 \leq \gamma \leq 1$ is an offset hyper-parameter (which we typically set to 1), $H(p) = -p\log(p) - (1-p)\log(1-p)$ is the Shannon entropy, and $\tilde{p}$ is the output of the propensity network for an input x. Thus, when the propensity score is 0 or 1, the dropout probability is equal to $1 - \frac{\gamma}{2}$, whereas when the propensity score is 0.5, the dropout probability is equal to $\frac{1}{2} - \frac{\gamma}{2}$. Propensity-dropout is simply a feature-dependent dropout scheme that imposes larger penalties on training examples with "bad" propensity scores, and hence prevents hidden units from co-adapting with "unreliable" training examples, which allows the learned potential outcomes network to generalize well to the actual feature distribution. The idea of propensity-dropout can be thought of as the conceptual analog of propensity-weighting (Abadie & Imbens, 2016) applied for conventional dropout networks (Srivastava et al., 2014). We dub our potential outcomes model a *deep counterfactual network* (DCN), and we use the acronym DCN-PD to refer to a DCN with propensity-dropout regularization. Since our model captures both the propensity scores and the outcomes, then it is a *doubly-robust* model (Dudík et al., 2014; 2011).

An important feature of a DCN-PD is its ability to associate its estimate $\tilde{T}(x)$ with a pointwise measure of confidence, which is a crucially important quantity in applications related to precision medicine (Athey & Imbens, 2016; Wager & Athey, 2015). This is achieved at inference time via a Monte Carlo propensity-dropout scheme in which we draw samples of $\tilde{T}(x)$ from our model (Gal & Ghahramani, 2016). Given a subject's feature x, a sample of $\tilde{T}(x)$ can be drawn from a DCN-PD as follows:

$$\begin{aligned}
\tilde{p}(x) &= f(\ldots f((\mathbf{w}_p^{(1)})^T x)\ldots), \\
\mathbf{r}_s^{(l)}, \mathbf{r}_{i,0}^{(l)}, \mathbf{r}_{i,1}^{(l)} &\sim \text{Bernoulli}(1 - \gamma/2 - H(\tilde{p}(x))/2), \\
\tilde{s}(x) &= f(\ldots f(\mathbf{r}_s^{(1)} \odot (\mathbf{w}_s^{(1)})^T x)\ldots), \\
\tilde{Y}^{(1)} &= f(\ldots f(\mathbf{r}_{i,1}^{(1)} \odot (\mathbf{w}_{i,1}^{(1)})^T \tilde{s}(x))\ldots), \\
\tilde{Y}^{(0)} &= f(\ldots f(\mathbf{r}_{i,0}^{(1)} \odot (\mathbf{w}_{i,0}^{(1)})^T \tilde{s}(x))\ldots), \\
\tilde{T} &= \tilde{Y}^{(1)} - \tilde{Y}^{(0)},
\end{aligned}$$

where $\mathbf{w}_p^{(l)}$, $\mathbf{w}_s^{(l)}$, $\mathbf{w}_{i,0}^{(l)}$ and $\mathbf{w}_{i,1}^{(l)}$ are the weight matrices for the l^{th} layer of the propensity, shared and idiosyncratic layers, respectively, $\mathbf{r}_s^{(l)}$, $\mathbf{r}_{i,0}^{(l)}$ and $\mathbf{r}_{i,1}^{(l)}$ are dropout masking vectors, and $f(.)$ is any activation function.

3.3. Training the Model

We train the network in alternating phases, where in each phase, we either use the treated batch $\mathcal{D}^{(1)}$ or the control batch $\mathcal{D}^{(0)}$ to update the weights of the shared and idiosyncratic layers. As shown in Algorithm 1, we run this process over a course of K epochs; the shared layers are updated in all epochs, wheres only one set of idiosyncratic layers is updated in any given epoch. Dropout is applied as explained in the previous Subsection with $\gamma = 1$. As visualized in Fig. 2, we can think of alternate training as deterministically dropping all units of one of the idiosyncratic layers in every epoch. We update the weights of all layers in each epoch using the *Adam* optimizer with default settings and Xavier initialization (Kingma & Ba, 2014).

Algorithm 1 Training a DCN-PD

Input: Dataset $\mathcal{D}$, number of epochs K
Output: DCN-PD parameters $(\mathbf{w}_s^{(l)}, \mathbf{w}_{i,1}^{(l)}, \mathbf{w}_{i,0}^{(l)})$
for $k = 1, k \leftarrow k+1, k \leq K$ **do**
  if k is even **then**
    $(\mathbf{w}_s^{(l)}, \mathbf{w}_{i,1}^{(l)}) \leftarrow \texttt{Adam}(\mathcal{D}^{(1)}, \mathbf{w}_s^{(l)}, \mathbf{w}_{i,1}^{(l)})$
  else
    $(\mathbf{w}_s^{(l)}, \mathbf{w}_{i,0}^{(l)}) \leftarrow \texttt{Adam}(\mathcal{D}^{(0)}, \mathbf{w}_s^{(l)}, \mathbf{w}_{i,0}^{(l)})$
  end if
end for

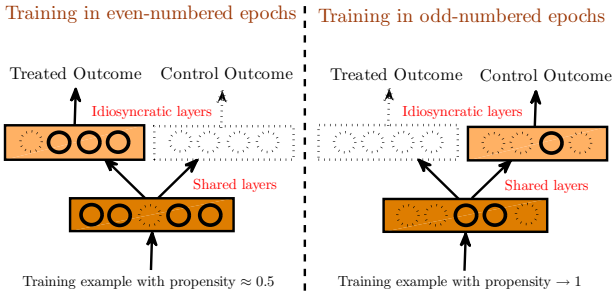

Figure 2. Visualization of the training algorithm.

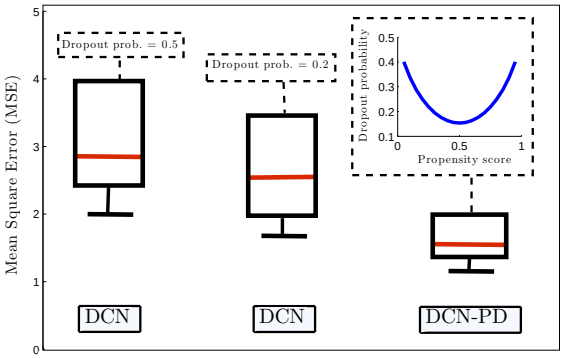

Figure 3. Performance gain achieved by propensity-dropout.

4. Experiments

The ground truth counterfactual outcomes are never available in an observational dataset, which hinders the evaluation of causal inference algorithms on real-world data. Following (Hill, 2012; Johansson et al., 2016), we adopt a semi-synthetic experimental setup in which the covariates and treatment assignments are real but outcomes are simulated. We conduct our experiments using the Infant Health and Development Program (IHDP) dataset introduced in (Hill, 2012). (The IHDP is a social program applied to premature infants aiming at enhancing their IQ scores at the age of three.) The dataset comprises 747 subjects (139 treated and 608 control), with 25 covariates associated with each subject. Outcomes are simulated based on the data generation process designated as the "Response Surface B" setting in (Hill, 2012).

We evaluate the performance of a DCN-PD model with $L_s = 2, L_{i,1} = L_{i,2} = 1$ (a total of 4 layers), and with 200 hidden units in all layers (ReLU activation), in terms of the mean squared error (MSE) of the estimated treatment effect. We divide the IHDP data into a training set (80%) and an out-of-sample testing set (20%), and then evaluate the MSE on the testing sample in 100 different experiments, were in each experiment a new realization for the outcomes is drawn from the data generation model in (Hill, 2012). (We implemented the DCN-PD model in a `Tensorflow` environment.) The propensity network is implemented as a standard 2-layer feed-forward network with 25 hidden layers, and is trained using the Adam optimizer.

The marginal benefits conferred by the propensity-dropout regularization scheme are illustrated in Fig. 3, which depicts box plots for the MSEs achieved by the DCN-PD model, and two DCN models with conventional dropout (dropout probabilities of 0.2 and 0.5 for all layers and all training examples). As we can see in Fig. 3, the DCN-PD model offers a significant improvement over the two DCN models for which the dropout probabilities are uniform over all the training examples. This result implies that the DCN-PD model generalizes better to the true feature distribution when trained with a biased dataset as compared to DCN with regular dropout, which suggests that propensity-dropout is a good regularizer for causal inference.

Table 1. Performance on the IHDP dataset.

Algorithm	MSE
k-NN	5.30 $\pm$0.30
Causal Forest	3.86 $\pm$0.20
BART	3.50 $\pm$0.20
BNN	2.45 $\pm$0.10
NN-4	2.88 $\pm$0.10
DCN	2.58 $\pm$0.06
DCN-PD	2.05 $\pm$0.03

In order to assess the marginal performance gain achieved by the proposed multitask model when combined with the propensity-dropout scheme, we compare the performance of DCN-PD with other state-of-the-art models in Table 1. In particular, we compare the MSE (averaged over 100 experiments) achieved by the DCN-PD with those achieved by k nearest neighbor matching (k-NN), Causal Forests with double-sample trees (Wager & Athey, 2015), Bayesian Additive Regression Trees (BART) (Chipman et al., 2010; Hill, 2012), and Balancing neural networks (BNN) (Johansson et al., 2016). (For BNNs, we use 4 layers with 200 hidden units per layer to ensure a fair comparison.) We also provide a direct comparison with a standard single-output feed-forward neural network (with 4-layers and 200 hidden units per layer) that treats the treatment assignment as an input feature (NN-4), and a DCN with a standard dropout with a probability of 0.2. As we can see in Table 1, DCN-PD outperforms all the other models, with the BNN model being the most competitive. (BNN is a strong benchmark as it handles the selection by learning a "balanced representation" for the input features (Johansson et al., 2016).) DCN-PDs significantly

outperforms the NN-4 benchmark, which suggests that the multitask modeling framework is a more appropriate conception of causal inference compared to direct modeling by assuming that the treatment assignment is an input feature.

References

Abadie, Alberto and Imbens, Guido W. Matching on the estimated propensity score. *Econometrica*, 84(2):781–807, 2016.

Athey, Susan and Imbens, Guido. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.

Austin, Peter C. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate behavioral research*, 46(3): 399–424, 2011.

Bottou, Léon, Peters, Jonas, Candela, Joaquin Quinonero, Charles, Denis Xavier, Chickering, Max, Portugaly, Elon, Ray, Dipankar, Simard, Patrice Y, and Snelson, Ed. Counterfactual reasoning and learning systems: the example of computational advertising. *Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

Charles, Dustin, Gabriel, Meghan, and Henry, J. Electronic capabilities for patient engagement among us nonfederal acute care hospitals: 2012-2014. *The Office of the National Coordinator for Health Information Technology*, 2015.

Chipman, Hugh A, George, Edward I, and McCulloch, Robert E. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, pp. 266–298, 2010.

Collobert, Ronan and Weston, Jason. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167. ACM, 2008.

Dudík, Miroslav, Langford, John, and Li, Lihong. Doubly robust policy evaluation and learning. *Proceedings of The 28th International Conference on Machine Learning*, 2011.

Dudík, Miroslav, Erhan, Dumitru, Langford, John, Li, Lihong, et al. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

Gal, Yarin and Ghahramani, Zoubin. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Hill, Jennifer L. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 2012.

Johansson, Fredrik D, Shalit, Uri, and Sontag, David. Learning representations for counterfactual inference. *Proceedings of The 33rd International Conference on Machine Learning*, 2016.

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lu, Min, Sadiq, Saad, Feaster, Daniel J, and Ishwaran, Hemant. Estimating individual treatment effect in observational data using random forest methods. *arXiv preprint arXiv:1701.05306*, 2017.

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

Rubin, Donald B. Matching to remove bias in observational studies. *Biometrics*, pp. 159–183, 1973.

Rubin, Donald B. Causal inference using potential outcomes. *Journal of the American Statistical Association*, 2011.

Shalit, Uri, Johansson, Fredrik, and Sontag, David. Estimating individual treatment effect: generalization bounds and algorithms. *Proceedings of The 34th International Conference on Machine Learning*, 2017.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1): 1929–1958, 2014.

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

Wager, Stefan and Athey, Susan. Estimation and inference of heterogeneous treatment effects using random forests. *arXiv preprint arXiv:1510.04342*, 2015.

Xu, Yanbo, Xu, Yanxun, and Saria, Suchi. A bayesian nonparametic approach for estimating individualized treatment-response curves. *arXiv preprint arXiv:1608.05182*, 2016.