Artificial Intelligence Research and Development
I. Sanz et al. (Eds.)

doi:10.3233/FAIA230655

Hydranet: A Neural Network for the Estimation of Multi-Valued Treatment Effects

Borja VELASCO-REGULEZ [a,1], Jesus CERQUIDES [a]
[a] *Artificial Intelligence Research Institute (IIIA-CSIC)*
ORCiD ID: Borja Velasco-Regulez https://orcid.org/0000-0003-4718-3388, Jesus Cerquides https://orcid.org/0000-0002-3752-644X

Abstract.
Neural network-based treatment effect estimation algorithms are well-known in the causal inference community. Many works propose new designs and architectures and report performance metrics over benchmarking data sets, in a *Machine Learning manner*. Nevertheless, most authors focus solely on binary treatment scenarios. This is a limitation, as many real-world scenarios have a multivalued treatment. In this work, we present a novel approach where we generalize a top-performing, neural network-based algorithm for binary treatment effect estimation to a multi-valued treatment setting. Our approach yields an estimator with desirable asymptotic properties, that delivers very good results in a wide range of experiments. To the best of our knowledge, this work is opening ground for the benchmarking of neural network-based algorithms for multi-valued treatment effect estimation.

Keywords. Causal Inference, Multi-valued Treatment Effect Estimation, Neural Networks

1. Introduction

Machine learning and neural networks are becoming a common choice for performing causal analysis tasks (causal inference, causal discovery) due to their power and flexibility for modelling complex functions, especially when dimensionality of the data is high[1]. Several authors have investigated specific network architectures, loss functions, regularization methods, etc. to tackle the task of inferring causal quantities using neural networks[2][3][4]. The performance of those algorithms is being benchmarked in the scientific literature, by using specific data sets and common metrics to achieve comparable results [5][6]. These advancements are happening almost exclusively in binary treatment scenarios. Nevertheless, often real-life applications have multiple-valued treatments (for instance, multi-armed clinical trials) or continuous treatments that can be discretized to multiple values [7][8]. This highlights the need to explore neural network-based causal inference methods for multi-valued treatments, both at the theoretical and empirical levels. In the present work, we select a top performance, neural network-based, binary average

[1]Corresponding Author: Borja Velasco-Regulez, bvelasco@iiia.csic.es

treatment effect estimation algorithm named Dragonnet[9] and test its generalizability to n-valued treatment settings. To the best of our knowledge, this[10] is the first attempt to establish a benchmark for this type of algorithm in the aforementioned setting. We present the theoretical and mathematical formulation, we develop a framework for experiments, and we provide the results obtained in different scenarios.

2. Problem Statement

Let the treatment of interest be a discrete random variable $T \in [0..k]$ that can take $k+1$ different values. Let the outcome be a continuous random variable $Y \in \mathbb{R}$, and let the covariates (i.e. the variables affecting both the treatment and the outcome) be a random vector $X \in \mathbb{R}^j$. Thus, our set of data points is $(Y_i, T_i, X_i),\ i \in [1..N]$, generated independently and identically. This set of data points constitutes our body of observational data. We define the causal effect of the treatment t over the outcome Y as, $\mu_t = \mathbb{E}[Y|do(T=t)]$, using Pearl's *do-calculus* notation [11], which denotes intervention. It can be shown that, if our data meets certain conditions, we can estimate causal (interventional) quantities based on observational data. Those conditions are known as the identifiability conditions: positivity, consistency and "no hidden confounder" conditions. For a more detailed explanation, see [11]. Under such conditions, $\mu_t = \mathbb{E}[Y|X=x, T=t]$, which is a quantity that is inferrable from our body of observational data. Along the rest of the section we assume that the identifiability conditions are fulfilled.

We define the conditional outcome as the expectation of the outcome given the treatment and the covariates, $Q(t,x) = \mathbb{E}[Y|t,x]$. Based on Q, we can construct a simple estimator $\hat{\mu}_t$ of μ_t as $\hat{\mu}_t = \frac{1}{N}\sum_i Q(t,x_i)$. In the following, we will be interested in approximating Q. Let $\hat{Q}$ be an approximation of Q. We define $\mu_t^{\hat{Q}} = \frac{1}{N}\sum_i \hat{Q}(t,x_i)$ as the estimator of μ_t obtained replacing Q by its estimation $\hat{Q}$. Furthermore, we define the Generalized Propensity Score (GPS[12]), expressed as $\mathbf{G}(x) = [g_0(x), g_1(x), \ldots, g_k(x)] \in \mathbb{R}^{k+1}$, with $g_t(x) = P(T=t|x)$.

In a binary treatment setting, under the identifiability conditions, the Average Treatment Effect (ATE) is one of the most common causal quantities of interest, and it is defined as $\psi = \mu_1 - \mu_0$. Given an approximation $\hat{Q}$ of Q, we could easily estimate ψ as $\psi^{\hat{Q}} = \mu_1^{\hat{Q}} - \mu_0^{\hat{Q}}$. In a multi-valued treatment setting, a wider class of causal quantities of interest can be defined, and all the conditional outcomes must be computed together in order to obtain valid estimates of those quantities[12]. In this work, we define such quantities of interest as the pair-wise average differences between the several treatments and a treatment considered the control (note that, in practice, the control treatment does not necessarily mean absence of treatment). Thus, we define a vector of ATEs $\psi \in \mathbb{R}^k$, $\psi = [\psi_1, \psi_2, \ldots, \psi_k]$, with $\psi_t = \mu_t - \mu_0$. We can approximate these quantities in a similar fashion as shown before, the t-th element of the vector being $\psi_t^{\hat{Q}} = \mu_t^{\hat{Q}} - \mu_0^{\hat{Q}}$. Note that if the causal quantity of interest was $\psi_{i,j} = \mu_i - \mu_j$, we could easily compute it based on the previous definition, as $\psi_{i,j} = \psi_i - \psi_j$, due to the linearity of the expectation operator.

The subject of interest in this the paper is the estimation of the vector of ATEs ψ. In the next section we generalize the estimation method provided in [9], which has the objective of estimating the ATE in the binary case, to the estimation of ψ in the multivalued treatment case presented above.

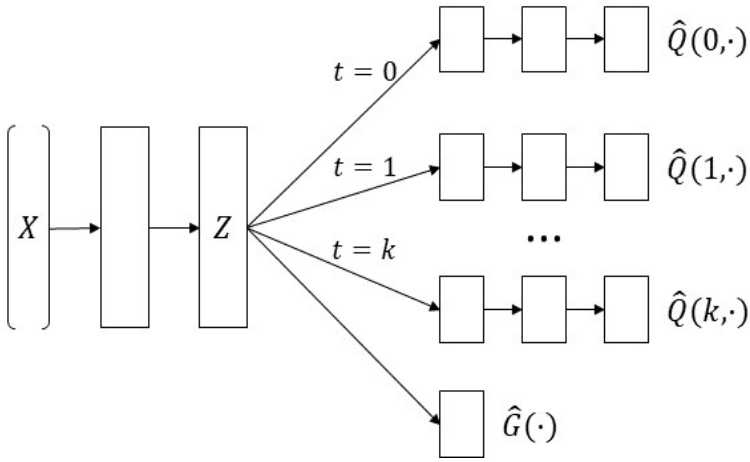


Figure 1. Hydranet architecture, where Z is the representation layer, and the $k+2$ heads correspond to the $k+1$ potential outcomes, $\hat{Q}(k,\cdot)$, and the Generalized Propensity Score, $\hat{G}(\cdot)$.

3. From Dragonnet to Hydranet

Dragonnet is a high-capacity, end-to-end neural network architecture for estimating binary treatment effects[9]. We present here the variation of the architecture, mathematical formulations and proofs for adapting Dragonnet to a multivalued treatment setting. We call this adaptation Hydranet.

3.1. Architecture

The architecture of Hydranet can be seen in Figure 1. It consists of two parts: the representation part, formed by the input layer and two hidden layers, and the heads, formed by $k+2$ ends. Out of those, $k+1$ correspond to the conditional outcomes, and are formed by two more hidden layers plus the output layer. The remaining head corresponds to the GPS, $\mathbf{G}(x) = [g_0(x), g_1(x), \ldots, g_k(x)] \in \mathbb{R}^{k+1}$, with $g_t(x) = P(T = t|x)$, consisting on just the output layer. All layers are fully connected. Recall that we approximate the t-th element of the vector of ATEs as $\psi_t^{\hat{Q}} = \frac{1}{N}\sum_i \hat{Q}(t,x_i) - \hat{Q}(0,x_i)$.

The baseline objective function has the shape

$$\hat{R}(\theta) = \frac{1}{N}\sum_i [(Q^{nn}(t_i,x_i;\theta) - y_i)^2 + \alpha \text{CrossEntropy}(g_t^{nn}(x_i;\theta),t_i)] \tag{1}$$

where the quadratic term relates to the errors of the potential outcomes' heads and the cross entropy term relates to the errors of the propensity score's head. The model parameters are

$$\hat{\theta} = \arg\min_{\theta}[\hat{R}(\theta)] \tag{2}$$

3.2. Targeted Regularization

Now, following the reasoning in [9], we present targeted regularization. Targeted regularization is a modification of the objective function that introduces an extra parameter, epsilon. In our setting, ε is a vector in $\mathbb{R}^k, \varepsilon = (\varepsilon_1, \varepsilon_2, ...\varepsilon_k)$, and the new objective function is

$$\bar{F}(\theta, \varepsilon) = \hat{R}(\theta) + \beta \frac{1}{N} \sum_i \gamma_i(y_i, t_i, x_i; \theta, \varepsilon), \text{ where} \tag{3}$$

$$\gamma_i(y_i, t_i, x_i; \theta, \varepsilon) = (y_i - \bar{Q}_i(\theta, \varepsilon))^2, \text{ and} \tag{4}$$

$$\bar{Q}_i(\theta, \varepsilon) = Q^{nn}(t_i, x_i) + \varepsilon_1 \left(\frac{\mathbf{I}(T=1)}{g_1^{nn}(x_i)} - \frac{\mathbf{I}(T=0)}{g_0^{nn}(x_i)} \right) + \ldots + \varepsilon_k \left(\frac{\mathbf{I}(T=k)}{g_k^{nn}(x_i)} - \frac{\mathbf{I}(T=0)}{g_0^{nn}(x_i)} \right) \tag{5}$$

with $\mathbf{I}(T=t)$ the indicator function, and thus the sought model parameters are defined by

$$\hat{\theta}, \hat{\varepsilon} = \arg\min_{\theta, \varepsilon} [\hat{R}(\theta) + \beta \frac{1}{N} \sum_i \gamma_i(y_i, t_i, x_i; \theta, \varepsilon)] \tag{6}$$

But why this modification in the first place? The answer lies in semiparametric estimation theory (SET) and targeted maximum likelihood estimation (TMLE). Very generally, SET provides us with conditions that ensure desirable properties of our estimator ψ when they are fulfilled, and TMLE is an efficient method to achieve the fulfillment of those conditions. The conditions are the set of non-parametric estimating equations, defined as

$$\mathbf{0} = \left[\frac{1}{N} \sum_i \varphi_{i,1}, \frac{1}{N} \sum_i \varphi_{i,2}, \ldots \frac{1}{N} \sum_i \varphi_{i,k} \right] \tag{7}$$

and they employ the elements of the vector of efficient influence curves, defined as $\varphi \in \mathbb{R}^k, \varphi = [\varphi_1, \varphi_2, ...\varphi_k]$, with

$$\varphi_{i,t} = Q^{nn}(t, x_i) - Q^{nn}(0, x_i) + \left(\frac{\mathbf{I}(T=t)}{g_t^{nn}(x_i)} - \frac{\mathbf{I}(T=0)}{g_0^{nn}(x_i)} \right) (y_i - Q^{nn}(t, x_i)) - \psi_t \tag{8}$$

Finally, recall that what we want is that the minimization of the modified objective function ensures the fulfillment of the non-parametric estimation equations. This can be expressed mathematically as

$$\mathbf{0} = \nabla \bar{F}|_{\hat{\varepsilon}} = \left[\frac{\partial \bar{F}}{\partial \varepsilon_1}, \frac{\partial \bar{F}}{\partial \varepsilon_2}, \ldots \frac{\partial \bar{F}}{\partial \varepsilon_k}\right]\Bigg|_{\hat{\varepsilon}} = \left[\frac{\beta}{N}\sum_i \varphi_{i,1}, \frac{\beta}{N}\sum_i \varphi_{i,2}, \ldots \frac{\beta}{N}\sum_i \varphi_{i,k}\right] \tag{9}$$

and the proof can be found in the supplementary material. This warrants the aforementioned desirable properties of the estimator ψ, i.e. double robustness, fast convergence, and efficiency.

4. The Data and the Metrics

We have tested Hydranet in two datasets, a fully synthetic one and a semi-synthetic one. In the remainder of the text we will refer to them as the synthetic dataset (or SynD for short) and the IHDP dataset, respectively. In order to generate these datasets we have designed and implemented algorithms mimicking different Data Generating Processes (DGP). For the synthetic dataset, the covariates, treatments and outcomes have been synthetically generated, and we have taken inspiration from [13]. For the IHDP dataset, the covariates are taken from a study with real participants, while the treatments and outcomes are synthetically generated. Those real covariates were collected for a Randomized Controlled Trial (RCT) carried out in 1985 [14], and are routinely used for benchmarking causal inference algorithms, usually following the configuration in[15]. We have followed a similar strategy but adapting the DGP to our needs (a multi-valued treatment scenario). With both datasets the number of treatments has been set to 5. In the remainder of this section we provide a more detailed explanation of the DGP of each dataset and its output.

4.1. Synthetic Dataset DGP

For generating fully synthetic data, we have designed DGPs with tunable parameters of dataset size *D*, bias size *B* and number of confounders *NC*. The number of treatments has been set to 5. The potential covariates are constituted by vectors $\mathbf{x} \in \mathbb{R}^{30}$ with each value sampled from a uniform distribution $\mathscr{U}(-1,1)$. The number of such vectors is equal to the data size parameter *D*, forming a matrix $\mathbf{X} \in \mathbb{R}^{Dx30}$. The actual confounders, i.e., the variables that participate in the determination of both the treatment and the outcome, are the first *NC* (number of confounders) elements of each covariate vector, thus forming a matrix $\mathbf{C} \in \mathbb{R}^{DxNC}$. The treatment for each datapoint has been obtained in two steps. First, we square the confounder vector element-wise and sum the elements, apply a $min - max$ scaler to the range $[0,4]$ (for 5 treatments), and round to the closest integer. Then, in order to fulfill the positivity condition, we draw the final treatment value from a categorical distribution such that

$$p(t|\mathbf{c}) = \begin{cases} 0.8, & \text{if } t = m(\mathbf{c}) \\ \frac{0.2}{k-1}, & \text{otherwise} \end{cases}$$

with $m(\cdot)$ the operation defined in the first step. Finally, for computing the potential outcomes, we have defined three outcome functions $(l_a(t,\mathbf{x}), l_b(t,\mathbf{x}), l_c(t,\mathbf{x}))$ that map a combination of the covariates and the treatment to the output space, for each datapoint. The outcome functions have the shape

$$l_a(t,\mathbf{x}) = 30\mathbf{v}_0^T\mathbf{x} + 10\,t^2\,\mathbf{v}_t^T\mathbf{x} + \varepsilon$$

$$l_b(t,\mathbf{x}) = 20\mathbf{v}_0^T\mathbf{x} + 5\,B\,t\,\mathbf{v}_t^T\mathbf{x} + \varepsilon$$

$$l_c(t,\mathbf{x}) = 10\mathbf{v}_0^T\mathbf{x} + 5log(|B\,t\,\mathbf{v}_t^T\mathbf{x}|) + \varepsilon$$

with B the bias parameter, $\mathbf{v}_0$ the baseline effect parameter, defined as $\mathbf{u}_0/||\mathbf{u}_0||$ with $||\cdot||$ the euclidean norm and $\mathbf{u}_0 \sim \mathscr{U}(0,1)$ a randomly sampled vector ($\mathbf{u}_0 \in \mathbb{R}^{30}$), and $\varepsilon \sim \mathscr{N}(0,1)$. Recall that a potential outcome, denoted $\mathbf{y}^t$, is the outcome that a datapoint would have had, had it been treated with a particular treatment t. The matrix of potential outcomes $\mathbf{Y} \in \mathbb{R}^{Dx5}$ is defined as $\mathbf{Y} = [\mathbf{Y}^0, \mathbf{Y}^1, \mathbf{Y}^2, \mathbf{Y}^3, \mathbf{Y}^4] = [l_a(\mathbf{0},\mathbf{X})^T,\ l_b(\mathbf{1},\mathbf{X})^T,\ l_c(\mathbf{2},\mathbf{X})^T,\ l_b(\mathbf{3},\mathbf{X})^T,\ l_a(\mathbf{4},\mathbf{X})^T]$, with $\mathbf{0} = (0,0,...0) \in \mathbb{R}^D$, $\mathbf{1} = (1,1,...1) \in \mathbb{R}^D$, etc.

We have generated datasets under varying values of the three parameters of interest, bias size $B = [2,5,10,30]$, dataset size $D = [1000,2000,5000,10000]$ and number of confounders $NC = [2,5,10,18]$, varying one parameter at a time. When kept fixed, the values have been set to $B = 20$, $D = 2000$ and $NC = 2$.

4.2. IHDP Dataset DGP

For generating the IHDP dataset, we have followed a similar strategy, but fixing $NC = 2$ and $B = 10$, and $D = 985$ being the size of the original IHDP covariate set. The treatment assignment function is based in two variables present in the set, mom ethnicity and weeks preterm. We assign treatment 0 to individuals with mom ethnicity equalling "black", treatment 1 to individuals with mom ethnicity equalling "white", treatment 2 to individuals with mom ethnicity equalling "hispanic", treatment 3 to individuals with mom ethnicity equalling "hispanic" and weeks preterm being bigger than 6, and treatment 4 to individuals with mom ethnicity equalling "black" and weeks preterm smaller than 6. Note that this setting is completely made up and has no connection with any real-life situation. Then, the final treatment is sampled from a probability distribution as explained in the previous section. The outcome functions are defined as

$$l_1(t,\mathbf{x}) = \exp(\mathbf{x}\beta) + B * MB + t^2 + \varepsilon$$

$$l_2(t,\mathbf{x}) = log(|\mathbf{x}\beta|) + B * MW * t + \varepsilon$$

$$l_3(t,\mathbf{x}) = \mathbf{x}\beta + B * MH + t^2 + \varepsilon$$

$$l_4(t,\mathbf{x}) = \exp(\mathbf{x}\beta) + B * WP + t + \varepsilon$$

$$l_5(t,\mathbf{x}) = log(|\mathbf{x}\beta|) + B * WP * t + \varepsilon$$

where β is a vector of parameters, B is the bias parameter, MB, MW and MH are the components of the one-hot encoding of mom ethnicity, and WP is weeks preterm.

4.3. Metrics

For performance benchmarking purposes, we have employed the sum of errors of the vector of ATEs. This is computed as the sum of the absolute values of the differences of

all estimated ATE components with respect to their true values, $E = \sum_{t=1}^{k} |\psi_t - \hat{\psi}_t|$. This choice allows to have a single real number as a final result, making comparisons simpler. All values have been computed as averages across 20 dataset realizations to increase the robustness of the results, and 95% confidence intervals have been computed with Bootstrapping.

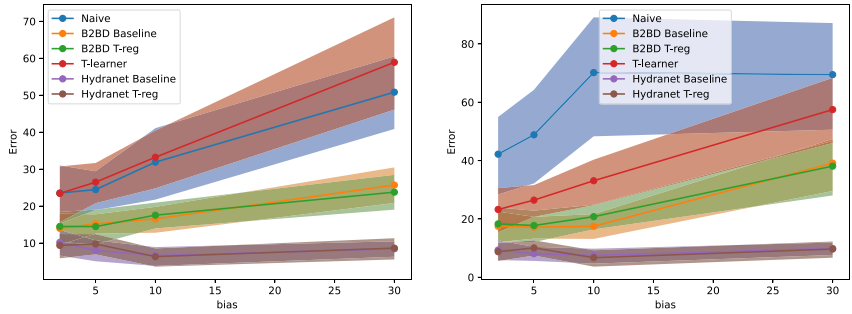
5. Experiments and Results

In the case of binary treatment settings, there are *de facto* benchmarking datasets and metrics, i.e., datasets and metrics that are widely used in the literature and thus serve for algorithmic performance comparison purposes. The IHDP dataset and the metrics presented in[15] are an example of this. This is not the case in multi-valued treatment settings, where comparators are scarce. Nevertheless, we have developed and implemented algorithms that can be considered comparable to Hydranet, to benchmark its performance. Thus, in every experiment, we present the results of the following algorithms: 1) **Naive**, a naive estimator of the treatment effect that employs only the observable data, without controlling, and serves to visualize the impact of confounding 2) **B2BD**, back to back Dragonnets, a strategy that uses 4 Dragonnets (with the same setup as in [9]), each one estimating one element of the vector of ATEs ψ, 3) **T-learner**, a T-learner [16] estimator that employs a gradient boosting machine (GBM) model[2], and finally 4) **Hydranet**, both in its baseline form and with targeted regularization. Hydranet performs well in all the tested scenarios and outperforms the comparators, both with in-sample (train set) data and with out-sample (test set) data, reaching low or very low error values for different dataset sizes, bias sizes and number of confounders. The employed training scheme consists of a first stage with the ADAM optimizer and a second stage with the Stochastic Gradient Descent (SGD) optimizer, with hyperparameters similar to [9].

5.1. Synthetic Data Experiments

Figure 2 and Table 1 show the error of the different algorithms for varying values of bias. As it can be seen, Hydranet performs better than the comparators. Hydranet is only mildly affected by the increasing size of the bias, while the comparators show bigger error increases. Similarly, Figure 3 and Table 2 show the performance of the algorithms for varying dataset sizes. As expected, all algorithms reduce their error with bigger data set sizes, but Hydranet with targeted regularization outperforms the rest, and shows a smaller error even for small dataset sizes, proving its (data) efficiency. Note that in this experiment, the output of Baseline Hydranet has been plugged into an Augmented Inverse Probability Weighting (A-IPTW) estimator, instead of the simple estimator presented in Section 1. The purpose of this change was to test the performance of Baseline Hydranet for delivering a plug-in estimator that is doubly robust. As it can be seen, this estimator fails for smaller dataset sizes, due to a well-reported phenomenon of finite-sample instability. This fact motivates the development and implementation of Targeted Regularization-equipped Hydranet. Finally, experiments with varying number of confounders show similar results, with Hydranet outperforming the alternatives.

[2] https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRegressor.html



(a) In sample (b) Out sample

Figure 2. Errors w.r.t. bias size

Table 1. Errors of the different algorithms w.r.t. bias size

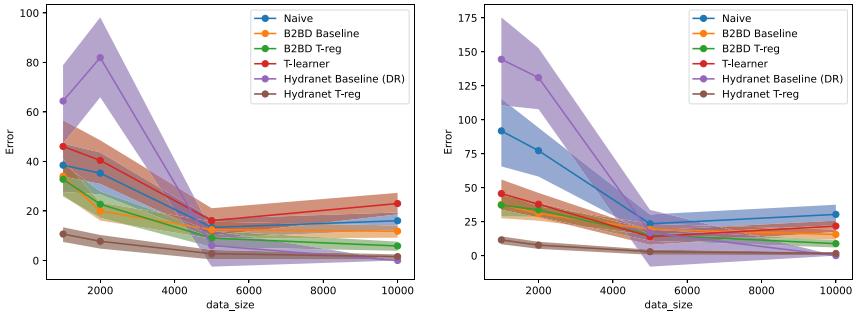
Bias	5		10		30	
	In-Sample	Out-Sample	In-Sample	Out-Sample	In-Sample	Out-Sample
Naive	48.83 ± 16.04	24.47 ± 5.27	70.16 ± 20.39	31.91 ± 9.67	69.46 ± 18.28	50.84 ± 9.82
B2BD base.	17.18 ± 3.74	15.19 ± 2.61	17.47 ± 4.25	16.55 ± 3.5	39.12 ± 8.91	25.7 ± 4.81
B2BD t-reg.	14.48 ± 5.38	14.48 ± 4.7	17.56 ± 4.22	17.56 ± 3.49	23.78 ± 9.35	23.78 ± 4.7
T-learn	26.44 ± 5.45	26.49 ± 5.44	33.09 ± 7.8	33.23 ± 7.82	57.48 ± 11.17	58.93 ± 12.48
Hydranet base.	8.08 ± 2.41	8.01 ± 2.83	7.48 ± 2.56	6.64 ± 2.55	9.88 ± 2.13	8.46 ± 2.09
Hydranet t-reg.	9.75 ± 2.6	9.75 ± 2.74	6.34 ± 2.81	6.34 ± 2.46	8.64 ± 2.77	8.64 ± 2.87

Table 2. Errors of the different algorithms w.r.t. dataset size

Data Size	2000		5000		10000	
	In-Sample	Out-Sample	In-Sample	Out-Sample	In-Sample	Out-Sample
Naive	77.23 ± 17.96	35.29 ± 8.46	23.45 ± 7.5	13.35 ± 3.93	30.34 ± 7.51	16.07 ± 3.5
B2BD base.	30.97 ± 5.3	19.95 ± 3.65	19.39 ± 4.2	12.46 ± 3.43	15.59 ± 3.73	11.83 ± 2.46
B2BD t-reg.	22.74 ± 5.11	22.74 ± 5.35	9.04 ± 4.15	9.04 ± 3.28	5.87 ± 2.63	5.87 ± 1.84
T-learn	37.85 ± 9.01	40.44 ± 8.81	13.95 ± 5.34	16.11 ± 5.52	21.76 ± 4.15	23.01 ± 4.34
Hydranet base.	130.98 ± 22.48	81.93 ± 16.15	16.83 ± 20.88	6.04 ± 7.29	2.43 ± 1.19	2.94 ± 1.10
Hydranet t-reg.	7.76 ± 2.61	7.76 ± 2.76	2.74 ± 1.9	2.74 ± 1.91	1.57 ± 1.07	1.57 ± 1.04

5.2. IHDP Data Experiments

Table 3 shows the error of the different algorithms with the IHDP dataset. Similarly as with synthetic data, Hydranet (both baseline and targeted regularization) outperform the comparators. The targeted regularization algorithm has a slightly bigger error than the baseline algorithm, but the difference is considered negligible. These results prove the efficacy of Hydranet with semi-synthetic data, showing its suitability for real-world scenarios.



(a) In sample (b) Out sample

Figure 3. Errors w.r.t. dataset size

Table 3. Performance with IHDP dataset

	Out-Sample	In-Sample
Naive	14.81 ± 0.97	17.51 ± 2.07
B2BD base.	22.75 ± 1.92	22.74 ± 2.38
B2BD t-reg.	22.22 ± 1.9	22.81 ± 2.68
T-learn	13.53 ± 1.22	13.7 ± 1.2
Hydranet base.	3.09 ± 0.56	3.03 ± 0.69
Hydranet t-reg.	3.82 ± 0.88	3.8 ± 0.87

6. Discussion

In this work, we have generalized a top-performing, neural network-based algorithm for ATE estimation from a binary treatment setting to a n-valued treatment setting. We have developed and implemented synthetic and semi-synthetic DGPs for algorithmic benchmarking purposes, and we have developed comparator algorithms for evaluating the performance of Hydranet. We show that Hydranet (both baseline and targeted regularization) performs well under different bias sizes, dataset sizes, and number of confounders, and we provide both theoretical and empirical evidence of the motivation for developing targeted regularization-equipped Hydranet. In addition, we show the good performance of the algorithm with semi-synthetic data. The direct generalizability of neural network-based algorithms for ATE estimation from binary settings to n-valued treatment settings is a common claim in the literature, but we show that it has its own challenges and that the behavior of the algorithms in each particular scenario requires its own interpretation. As far as we know, this paper is opening ground on the proposal of benchmarking results for neural network-based ATE estimation in multivalued treatment scenarios.

The main limitations of this work are twofold: on one hand, we have been forced to construct the competitor algorithms of Hydranet ourselves, due to the scarcity of

benchmarking data in the literature. On the other hand, we have only tested this algorithm for a 5-valued treatment scenario. It is a line of future work to adapt the algorithm and perform experiments for k-valued scenarios. Finally, we have run into some instabilities during neural network training in some of the experiments. It is also a line of future work to correct them and ensure the stability of the training process.

Acknowledgments

This work was supported by Doctorat Industrial funded by Generalitat de Catalunya [DI-2020-18] and by project CI-SUSTAIN funded by the Spanish Ministry of Science and Innovation [PID2019-104156GB-I00]. Borja Velasco-Regúlez is a PhD Student of the doctoral program in Computer Science at the Universitat Autonoma de Barcelona.

References

[1] Hernán MA, Robins JM. Causal Inference: What If;.
[2] Yoon J, Jordon J, Van Der Schaar M. GANITE: Estimation of individualized treatment effects using generative adversarial nets. In: International Conference on Learning Representations; 2018. .
[3] Johansson FD, Shalit U, Sontag D. Learning Representations for Counterfactual Inference. arXiv:160503661 [cs, stat]. 2018 Jun. ArXiv: 1605.03661. Available from: http://arxiv.org/abs/1605.03661.
[4] Nair N, Gurumoorthy KS, Mandalapu D. Individual Treatment Effect Estimation Through Controlled Neural Network Training in Two Stages. arXiv; 2022. Number: arXiv:2201.08559 arXiv:2201.08559 [cs]. Available from: http://arxiv.org/abs/2201.08559.
[5] Lin A, Merchant A, Sarkar SK, D'Amour A. Universal Causal Evaluation Engine: An API for empirically evaluating causal inference models:9.
[6] Shimoni Y, Yanover C, Karavani E, Goldschmnidt Y. Benchmarking Framework for Performance-Evaluation of Causal Inference Analysis. arXiv:180205046 [cs, stat]. 2018 Mar. ArXiv: 1802.05046. Available from: http://arxiv.org/abs/1802.05046.
[7] Cattaneo MD, Drukker DM, Holland AD. Estimation of Multivalued Treatment Effects under Conditional Independence. The Stata Journal: Promoting communications on statistics and Stata. 2013 Sep;13(3):407-50. Available from: http://journals.sagepub.com/doi/10.1177/1536867X1301300301.
[8] Li F, Li F. Propensity score weighting for causal inference with multiple treatments. The Annals of Applied Statistics. 2019 Dec;13(4):2389-415. Publisher: Institute of Mathematical Statistics. Available from: https://projecteuclid.org/journals/annals-of-applied-statistics/volume-13/issue-4/Propensity-score-weighting-for-causal-inference-with-multiple-treatments/10.1214/19-AOAS1282.full.
[9] Shi C, Blei DM, Veitch V. Adapting Neural Networks for the Estimation of Treatment Effects. arXiv:190602120 [cs, stat]. 2019 Oct. ArXiv: 1906.02120. Available from: http://arxiv.org/abs/1906.02120.
[10] Velasco B, Cerquides J, Arcos JL. Hydranet: A Neural Network for the estimation of Multi-valued Treatment Effects. In: NeurIPS 2022 Workshop on Causality for Real-world Impact; 2022. Available from: https://openreview.net/forum?id=sJChORLuPHK.
[11] Pearl J, Madelyn Glymour,, Nicholas P Jewell. Causal Inference in Statistics. A Primer. John Wiley and Sons Ltd, United States; 2016.
[12] Cattaneo MD. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. Journal of Econometrics. 2010 Apr;155(2):138-54. Available from: https://linkinghub.elsevier.com/retrieve/pii/S030440760900236X.
[13] Kaddour J, Zhu Y, Liu Q, Kusner MJ, Silva R. Causal Effect Inference for Structured Treatments.
[14] Enhancing the Outcomes of Low-Birth-Weight, Premature Infants: A Multisite, Randomized Trial. JAMA. 1990 Jun;263(22):3035-42. _eprint:

https://jamanetwork.com/journals/jama/articlepdf/382131/jama_263_22_030.pdf. Available from: https://doi.org/10.1001/jama.1990.03440220059030.

[15] Dorie V, Hill J, Shalit U, Scott M, Cervone D. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. arXiv:170702641 [stat]. 2018 Jul. ArXiv: 1707.02641. Available from: http://arxiv.org/abs/1707.02641.

[16] Künzel SR, Sekhon JS, Bickel PJ, Yu B. Metalearners for estimating heterogeneous treatment effects using machine learning. Proceedings of the National Academy of Sciences. 2019;116(10):4156-65. _eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1804597116. Available from: https://www.pnas.org/doi/abs/10.1073/pnas.1804597116.

A. Supplementary material

A.1. Hydranet

We want to prove that

$$\left.\frac{\partial \bar{F}}{\partial \varepsilon_t}\right|_{\hat{\varepsilon}_t} = \frac{1}{N}\sum_i \varphi_{i,t}, \qquad \forall\, t \text{ in}[0,k] \tag{10}$$

Proof. On one hand, using equations (3), (4) and (5) we get

$$\begin{aligned}
\left.\frac{\partial \bar{F}}{\partial \varepsilon_t}\right|_{\hat{\theta},\hat{\varepsilon}_t} &= \left.\frac{\partial}{\partial \varepsilon_t}\left(\hat{R}(\theta) + \beta \frac{1}{N}\sum_i \gamma_i(y_i, t_i, x_i; \theta, \varepsilon)\right)\right|_{\hat{\theta},\hat{\varepsilon}_t} \\
&= \left.\frac{\beta}{N}\sum_i \frac{\partial}{\partial \varepsilon_t}\gamma_i(\theta,\varepsilon)\right|_{\hat{\theta},\hat{\varepsilon}_t} \\
&= \left.\frac{2\beta}{N}\sum_i (y_i - \bar{Q}_i(\theta,\varepsilon))\frac{\partial \bar{Q}_i(\theta,\varepsilon)}{\partial \varepsilon_t}\right|_{\hat{\theta},\hat{\varepsilon}_t} \\
&= \left.\frac{2\beta}{N}\sum_i \left[(y_i - \bar{Q}_i(\theta,\varepsilon))\left(\frac{\mathbf{I}(T=t)}{g_t^{nn}(\theta)} - \frac{\mathbf{I}(T=0)}{g_0^{nn}(\theta)}\right)\right]\right|_{\hat{\theta},\hat{\varepsilon}_t} \\
&= \frac{2\beta}{N}\sum_i \left[(y_i - \hat{Q}(t,x_i)\left(\frac{\mathbf{I}(T=t)}{\hat{g}_t} - \frac{\mathbf{I}(T=0)}{\hat{g}_0}\right)\right] \text{(evaluate at } \hat{\theta},\hat{\varepsilon}) \\
&= \frac{2\beta}{N}\sum_i \left(\hat{Q}(t,x_i) - \hat{Q}(0,x_i)\right) - \frac{\beta}{N}\sum_i \left(\hat{Q}(t,x_i) - \hat{Q}(0,x_i)\right) + \\
&\frac{\beta}{N}\sum_i \left[(y_i - \hat{Q}(t,x_i))\left(\frac{\mathbf{I}(T=t)}{\hat{g}_t} - \frac{\mathbf{I}(T=0)}{\hat{g}_0}\right)\right] \text{(add and subtract term)} \\
&= \frac{2\beta}{N}\sum_i \left[\hat{Q}(t,x_i) - \hat{Q}(0,x_i) + (y_i - \hat{Q}(t,x_i))\left(\frac{\mathbf{I}(T=t)}{\hat{g}_t} - \frac{\mathbf{I}(T=0)}{\hat{g}_0}\right) - \hat{\psi}_t\right] \text{(group sums)}
\end{aligned}$$

On the other hand, by substituting the definition of the efficient influence curves (8) in the set of non-parametric estimation equations (9), multiplying by β and particularizing at $\hat{Q}, \hat{g}, \hat{\psi}$ (the functions modelled by the neural network at the optimal point of the parameter space), we obtain an expression equal to the one in the last line of the proof. Thus, the non-parametric estimation equations (9) are satisfied, and the proof is complete. □ □