Deep Propensity Network using Sparse Autoencoders

Shantanu Ghosh,[1] **Mattia Prosperi**[2]
[1]Computer and Information Science and Engineering
[2]Department of Epidemiology, College of Public Health and Health Professions & College of Medicine
University of Florida, Florida 32611, USA
{shantanughosh, m.prosperi}@ufl.edu

Abstract

Understanding causality is critical, especially in biomedical and social science; and the gold-standard solution is to perform randomized experiments, enacting interventional probabilities as opposed to conditional ones, but it is not often feasible due to operational or ethical constraints. On the other hand, drawing causal estimates from pre-existing, observational data is problematic, because datasets are often littered with underlying bias. Identifying the true causal effects however is important to evaluate what-if scenarios, i.e. counterfactuals. In fact, a prediction model based only on conditional probabilities, even with perfect accuracy, is neither guaranteed to estimate the correct causal effect, in terms of individual or average treatment effects (ITE, ATE), nor to calculate the correct counterfactuals. Propensity score matching (PSM) is a popular statistical approach for observational data that attempts to estimate the effect of an intervention, e.g. a medical treatment, by taking into account other factors that may bias the chance to undertake the intervention itself, e.g. social discrimination. PSM is usually implemented with logistic regression, but its performance can be limited due to linearity, high-dimensionality, and residual confounding in the feature space. Recently, deep counterfactual neural networks with propensity dropout (DCN-PD) have been introduced, enabling nonlinear PSM with advantage over classical methods in terms of treatment assignment error and ATE. In this work, we propose a deep propensity network using a sparse autoencoder (DPN-SA), a novel deep learning architecture for PSM to tackle the problem of high-dimensionality and residual confounding. DPN-SA takes advantage of a sparse autoencoder to reduce the dimension of the feature space. Tests performed on real-world observational data showed that the DPN-SA outperforms logistic regression-, LASSO-, and DCN-PD in estimating treatment assignment probability, ITE and ATE. The code is available under the MIT license on Github at: https://github.com/Shantanu48114860/

1 Introduction

In many research fields, such as biomedical and social sciences, preexisting (i.e. observational) data may contain underlying bias, arising in various steps of the data generation or collation process, for which datasets cannot be used seamlessly to draw causal claims (Prosperi et al. 2020). For instance, one could be interested in studying the effectiveness of certain medical treatments or interventions in the population, but the way in which people access the healthcare system could be different (e.g. due to social inequality, systemic racism). Therefore, due to such bias, the causal effects of the treatments could not be estimated properly. One solution would be to force the interventions to be non-discriminated, performing randomized controlled experiments or trials (RCT) (Sibbald and Roland 1998) (Concato, Shah, and Horwitz 2000). In RCT, individuals are assigned to different treatment (or control, no treatment) groups at random, regardless their background characteristics (i.e. the covariate or feature space). The mean difference between the observed treatment outcomes of two different groups is called the average treatment effect (ATE). Note that the individual treatment effect (ITE) is a missing data problem (Pearl, Glymour, and Jewell 2016) (Hernan and Robins 2019), because only one factual outcome can be observed (a person cannot be assigned to both treatment and control groups). Since RCT are not always feasible due to ethical or operational constraints, e.g. conducting RCT to assess the effect of smoking toward development of lung cancer, observational data are often used, and one must account for the possible types of underlying bias, including confounders and colliders (Greenland and Morgenstern 2001).

Propensity score matching (PSM) is a popular statistical approach for observational data that attempts to estimate the causal effect of a treatment variable with respect to an outcome, taking into account possible confounding bias from other pre-treatment characteristics (Rosenbaum and Rubin 1983) (Rosenbaum and Rubin 1984). The propensity score is a scalar estimate $\pi(\boldsymbol{x})$ representing the conditional probability of receiving a certain treatment $T = 1$, versus the control group or no treatment $T = 0$, given a set of measured pre-treatment covariates $\boldsymbol{X}$, denoted as

$$\pi(\boldsymbol{x}) = P(T = 1|\boldsymbol{X} = \boldsymbol{x}), \qquad (1)$$

Hence, PSM balances the pre-treatment potential confounders by achieving a quasi-randomization of the different treatment group assignments, allowing a better estimation of the treatment effect. However, PSM accounts only for measured (and measurable) covariates, and latent bias may remain after matching (Garrido and others 2014).

PSM has been implemented historically through logistic regression, which calculates the probability of treatment assignment given the pre-treatment covariates (Kurth et al. 2005). In presence of high-dimensional datasets, compiled from large electronic health record (EHR) databases (Prosperi et al. 2018), different feature selection methods within PSM have been employed, such as the high-dimensional propensity score (hdPS) (Schneeweiss et al. 2009) or L1-logistic regression (Tian, Schuemie, and Suchard 2018). However, logistic regression is limited because it calculates a mere linear combination of input variables, thus it is not able to capture complex relationships between the pre-treatment covariates and the treatment assignment. This is particularly true in high-dimensional settings, where it is problematic to define explicitly variable-to-variable interactions, e.g. as higher-order terms in the logistic function, and computationally burdensome to scan all of them.

An artificial neural network is a universal approximator, and can smooth polynomial functions regardless of the order of the polynomial or the number of interaction terms (Barron 1994) (Bishop and others 1995) (Mhaskar 1996). In addition, it does not require *a priori* knowledge of what interactions and functional forms are likely to be relevant among covariates. Therefore, it is suited to overcome the issues of PSM by logistic regression. The network can provide estimation of probability of treatment group assignment (e.g. through *softmax*), and consequently of ATE and ITE.

Alaa, Weisz and van Der Schaar designed a multitask deep counterfactual network with propensity dropout (DCN-PD) (Alaa, Weisz, and Van Der Schaar 2017) which is a feed-forward network with a set of shared layers is used to calculate the counterfactual outcomes, being regularized by another network that calculates propensity scores and a dropout probability for training examples (Gal and Ghahramani 2016), through alternate training phases (treatment vs. control groups) where the shared and the outcome layers' weights are updated. The DCN-PD led to lower mean squared error (MSE) in treatment assignments and more accurate ATE than other classic PSM methods.

In this work, we propose a novel deep neural architecture –the deep propensity network using a sparse autoencoder (DPN-SA)– that addresses the problem of high-dimensional PSM, yet maintains the MSE advantage of the DCN-PD approach as compared to others. The DPN-SA estimates the propensity score using a sparse autoencoder (Ng and others 2011) which at the same time learns a nonlinear feature representation and reduces the dimensionality of the pre-treatment covariate space. The sparse autoencoder is learnt in an end-to-end manner, while the counterfactual prediction follows the DCN-PD layout.

2 Methodology

2.1 Problem Formulation

Let assume a population sample (independent and identically distributed) of N $(1...i...n)$ individuals, given background set of pre-treatment covariates $\boldsymbol{X}$, a treatment T (binary, for simplicity) and a health outcome Y. Each subject i is represented by the tuple $\{\boldsymbol{X}_i, T_i, Y_i\}$. Let Y_i^0 and Y_i^1 the potential outcome for individual i under treatment $T_i = 0$ and $T_i = 1$, respectively (Rubin 1974) (Rubin 2005). Given $\boldsymbol{X}_i = \boldsymbol{x}$, the ITE $\tau(\boldsymbol{x})$ is defined as the difference in the mean potential outcomes for the individual i under both treatments, conditional on the observed covariate vector $\boldsymbol{x}$

$$\tau(\boldsymbol{x}) = \mathbb{E}[Y_i^1 - Y_i^0 \mid \boldsymbol{X}_i = \boldsymbol{x}] \tag{2}$$

The ITE formulation as $\tau(\boldsymbol{x})$ –called the counterfactual framework– is usually incalculable in reality, since people cannot be assigned two different treatments at the same time. However, under the assumption of strongly ignorable treatment assignment (SITA), which means that the potential outcomes are independent of treatment conditional on background variables, i.e. $\{Y_i^1 Y_i^0\} \perp T \mid \boldsymbol{X}$ (Imbens 2000) (Lechner 2001) (Pearl, Glymour, and Jewell 2016) (Peters, Janzing, and Schölkopf 2017).

Under the assumption of SITA, the ITE can be calculated as $\tau(\boldsymbol{x}) = \mathbb{E}[Y^1 \mid T = 1, \boldsymbol{X} = \boldsymbol{x}] - \mathbb{E}[Y^0 \mid T = 0, \boldsymbol{X} = \boldsymbol{x}]$ $= \mathbb{E}[Y \mid T = 1, \boldsymbol{X} = \boldsymbol{x}] - \mathbb{E}[Y \mid T = 0, \boldsymbol{X} = \boldsymbol{x}]$.

Further, under SITA and by averaging over the distribution of $\boldsymbol{X}$, the ATE τ_{01} can be calculated as

$$\tau_{01} = \mathbb{E}[\tau(\boldsymbol{X})] = \mathbb{E}[Y \mid T = 1] - \mathbb{E}[Y \mid T = 0] \tag{3}$$

However, by assuming SITA, ITE and ATE can be calculated only with $\boldsymbol{x}$ being the same in treatment groups, which becomes quickly unfeasible when the dimension of $\boldsymbol{x}$ grows. PSM, through the conditional probability $\pi(\boldsymbol{x})$ (see eq. 1) attempts at balancing the probability of receiving T given $\boldsymbol{X} = \boldsymbol{x}$.

Once propensity scores are obtained for a population sample, the individuals in the treatment group must be matched those in the control group to make sure that they are balanced with respect to the background covariates; this is a second problem for which a number of (approximate) solutions can be used, including *k*-nearest neighbor, Caliper matching (Austin 2014), and propensity weighing (Abadie and Imbens 2016) (on which the DCN-PD was based).

2.2 Proposed Approach

The DCN-PD method (Alaa, Weisz, and Van Der Schaar 2017) offered a novel and effective way to calculate nonlinear propensity scores as well as to match individuals, providing better ITE/ATE estimation over classical PSM methods. However, DCN-PD might be affected by curse of dimensionality, with associated residual confounding in study settings where the covariate space has potentially very high cardinality, e.g. in target trial designs that use EHR (Zhang et al. 2018) (Li et al. 2020). Feature selection and shrinkage has been proven effective for PSM by logistic regression (Schneeweiss et al. 2009) (Tian, Schuemie, and Suchard 2018), and in the same way dimensionality reduction can be implemented in deep neural architectures (Hinton and Salakhutdinov 2006). An autoencoder is a neural network that learns to copy its input to its output (encoder-decoder), but the input is coded into a lower dimension within the hidden layer(s) (Hinton and Zemel 1993). In its simplest form,

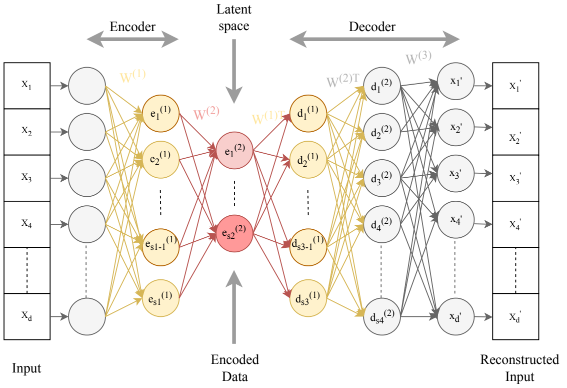

Figure 1: Architecture of a deep stacked sparse autoencoder with the sparsity constraint (brighter neurons are active) enforced in the 2^{nd} layer

i.e. a single layer with linear/sigmoid activations, the autoencoder is closely related to principal component analysis (PCA), while highly nonlinear codes can be achieved by augmenting the layer architecture, e.g. deep beliefs networks (Zhou et al. 2014). Autoencoders have been employed in several applications, from machine translation to drug discovery (Cho et al. 2014) (Zhavoronkov et al. 2019). The sparse autoencoder is an approach that includes extra units (more than inputs) in the hidden layer, but only a small number of them units are allowed to be activated depending on the input (Ng and others 2011) (Makhzani and Frey 2013). It has been also broadly applied in biomedical studies, including imaging and -omics datasets (Yu et al. 2017) (Mao et al. 2018) (Praveen et al. 2018) (Lemsara, Ouadfel, and Fröhlich 2020).

The DPN-SA that we propose here exploits a deep stacked sparse autoencoder to encode the covariate space $\boldsymbol{X}$ into a lower dimensional, nonlinear feature representation. After training, the decoder is replaced by a *softmax* classifier which calculates the probability of treatment T assignment, thus estimating the propensity score $\pi(\boldsymbol{X})$. The matching uses the propensity dropout component of the DCN-PD, also used for ITE/ATE calculation. Different training (end-to-end vs greedy stacked) procedures and layer (multiple vs single) architectures are tested on real and synthetic high-dimensional data, as detailed in the next sections.

2.3 Architecture and Training of DPN-SA

A deep stacked sparse autoencoder consists of multiple encoders and decoders to learn the identity function of the feature vectors, stacked up to each other, with the setup of sparsity constraints in the hidden layers, whose neurons can be activated or not depending on the input, as shown in Figure 1.

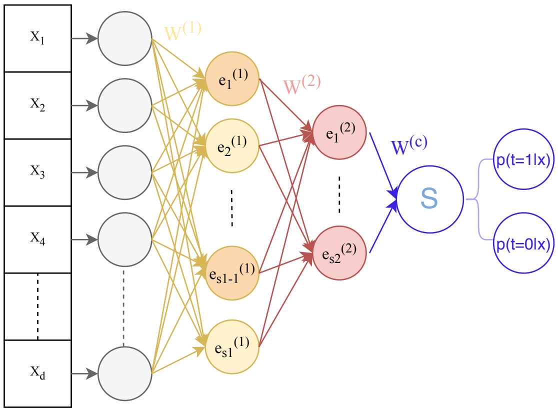

Figure 2: The deep propensity network, stacking the multilayer sparse autoencoder next to the *softmax* classifier

Let $\boldsymbol{X} = \{x(1), x(2), ..., x(N)\}$ represent the population of $x(i)$ individuals ($j = 1...N$ i.i.d samples). Each $x(i) \in \mathbb{R}^d$ is a d-dimensional vector, where d represents the number of pre-treatment covariates, i.e. the number of features. The DPN-SA uses up to l encoder and l decoder layers stacked together one after the other, with an additional linear layer at the end of the last decoder. The activation function of each encoder and decoder is the hyperbolic tangent layer*tanh*. As $x(i) \in \mathbb{R}^d$, the reconstructed output also needs to be $x'(i) \in \mathbb{R}^d$. Supposing we set $l = 2$, the sample input covariate vector $x(i)$ is reconstructed in the forward propagation as follows:

$$e^{(1)}(i) = f(W^{(1)}x(i) + b^{(1)}) \tag{4}$$

$$e^{(2)}(i) = f(W^{(2)}e^{(1)}(i) + b^{(2)}) \tag{5}$$

$$d^{(1)}(i) = f(W^{(2)T}e^{(2)}(i) + b^{(3)}) \tag{6}$$

$$d^{(2)}(i) = f(W^{(1)T}d^{(1)}(i) + b^{(4)}) \tag{7}$$

$$x'(i) = W^{(3)}d^{(2)}(i) + b^{(5)} \tag{8}$$

where, $e^{(l)}(i)$, $d^{(l)}(i)$, $x'(i)$ are the activations of the encoder, decoder and the reconstructed input of l^{th} layer for the i^{th} sample, respectively. $W^{(l)}$ and $W^{(2)}$ are the weight matrices of encoder 1 and encoder 2, having sizes $s1$ x d and $s2$ x $s1$, $W^{(2)T}$ and $W^{(1)T}$ are the weight matrices of decoder 1 and decoder 2 having sizes $s1$ x $s2$ and d x $s1$. $s(l)$ denotes size or number of neurons of l^{th} layer and *f(.)* is the activation function (*tanh*).

After reconstructing the a sample input feature $x(i)$, the objective function of the sparse autoencoder $J_{sparse}(W, b)$ has to be minimized, as described in (Ng and others 2011):

$$\begin{aligned} J(W, b) = &\left[\frac{1}{2N}\sum_{i=1}^{N}\left\|x'_{W,b}(i) - x(i)\right\|^2\right] \\ &+ \frac{\lambda}{2N}\sum_{l=1}^{L}\left\|W^{(l)}\right\|_F^2 \end{aligned} \tag{9}$$

$$J_{sparse}(W, b) = J(W, b) + \beta\sum_{j=1}^{s2} KL(\rho||\rho_j) \tag{10}$$

where λ is the regularization parameter, $W^{(l)}$ is the weight matrix corresponding to l^{th} layer of the network, s2 is the number of neurons in the 2^{nd} hidden layer, β is the weight of the sparsity penalty, KL is the Kullback-Leibler divergence, and the subscript F is the Frobenius norm (equivalent to the squared norm of the weight matrix). The $J_{sparse}(W, b)$ is minimized using backpropagation, for K_{sa} number of epochs.

The DPN-SA is trained in two phases. In the first phase the sparse autoencoder is trained and optimized (K_{sa} epochs). The parameters (W, b) are updated by the Adam optimizer in each iteration. After the sparse autoencoder has learnt the latent representation of the covariates, the decoder part is removed and a *softmax* classifier is attached to the end of the last encoder layer, as shown in 2. The *softmax* classifier is trained for K_c number of epochs. The final network gives the estimation of the propensity score $\pi(\boldsymbol{x})$. Algorithm 1 describes the two-phase procedures to obtain the final DPN-SA.

Algorithm 1 Training DPN-SA

Input: Batch *B* of random samples with assigned treatment *T*, training set X_{train}, number of epochs K_s, K_c, learning rates lr_{SA}, lr_c for the sparse autoencoder (SA) and the *softmax* classifier, respectively.

Output: PS_{out} consisting of propensity score for each sample *i*, where i ranges from 1 to *N*.

```
procedure DPN-SA TRAINING
    Initialise $W^{(l)}, b^{(l)}$ of the SA.
    for $epochs = 1, 2, \dots K_s$ do
        for $batches = 1, 2, \dots B$ in $X_{batch}$ do
            Compute $x'_{batch}$ using forward propagation.
            Compute $J_{sparse}$.
            Compute Gradient: $\nabla(J_{sparse})$
            $(W^{(l)}, b^{(l)}) \leftarrow$ Adam$(X_{batch}, W^{(l)}, b^{(l)})$
        end for
    end for
    Remove the decoder from the SA
    Attach a softmax classifier to the last encoder
    Initialize $W^{(c)}, b^{(c)}$ of the classifier
    $PS_{out} \leftarrow$ Empty
    for $epochs = 1, 2, \dots K_c$ do
        for $batches = 1, 2, \dots B$ in $X_{batch}$ do
            $(W^{(l)}, b^{(l)}, W^{(c)}, b^{(c)}) \leftarrow$ Adam$(X_{batch}, W^{(l)}, b^{(l)}, W^{(c)}, b^{(c)})$
            Get the propensity score $\pi(T|X_{batch})$ and add to $PS_{out}$
        end for
    end for
end procedure
```

2.4 Experimental Setup

Datasets. We used the Infant Health and Development Program (IHDP) dataset, a multi-site, longitudinal, randomized, clinical trial designed to evaluate the efficacy of comprehensive early intervention in enhancing the outcomes of low birth weight, prematurely born infants in the United States. The dataset consists of 747 subjects (139 in the treatment group and 608 controls), with 25 associated covariates, describing characteristics of the infants and their mothers. Here we adopt the simulated outcome completion, used as a benchmark in several causal inference methodology and modelling studies, including (Hill 2011) and (Alaa, Weisz, and Van Der Schaar 2017).

In addition to IHDP, we also created a synthetic data set with larger covariate space, first by doubling the original feature set through the creation of 25 random variables (shuffling the original ones), and then triplicating it by creating another set of 25 covariates partially correlated to the original ones (apprx. ρ=0.4), using a Gaussian noise addition $\mathcal{N}(0, 2\sigma(\boldsymbol{x}_k)))$ to each original variable $\boldsymbol{x}_k$.

DPN-SA and Other Comparison Methods. Four different configurations of the DPN-SA were tested –on both real and synthetic datasets: (i) 25-20-10-20-25; (ii) 25-10-25, which is similar to PCA; (iii) 25-1-25, which is similar to regularized logistic regression. For all four, both end-to-end and stacked greedy training were executed. In the greedy layer wise training, we employed two strategies - 1) we trained only the newly added layer while keeping the other layers fixed and stacked them one after the other; 2) we trained the newly added layer along with the previous layers and stacked them one after the other. The learning rates for the sparse autoencoder and for the *softmax* were 0.001 and 0.01, respectively. We set weight decay (λ), sparsity parameter (ρ), and sparse penalty (β) to 0.0003, 0.8, and 0.1, respectively, with a batch size of 32. The *softmax* classifier was ran for 50 epochs (K_c) with a batch size of 32. The DPN-SA was implemented using the *Pytorch* framework (https://pytorch.org/).

In addition to the DPN-SA, we run and compared: (i) the original DCN-PD, (ii) a standard logistic regression, and (iii) logistic regression with LASSO regularization.

Testing. The models were trained on 80% of the data and validated on the remaining 20% , repeating the procedure for 100 times, calculating MSE, ITE/ATE in the same way as (Alaa, Weisz, and Van Der Schaar 2017). The error distributions were compared by means of a t-test with sample overlap adjustment (Nadeau and Bengio 2003).

3 Results

Figures 3 and 4 show the histogram, quantile-quantile, and scatterplots of the $\pi(\boldsymbol{X})$ distribution (stratified by treatment group) of the PDN-SA and the other models (real dataset). The propensity scores among all DPN-SA were well-correlated, similarly to the correlation between DPN-SA and DCN-PD. Logistic regression and LASSO were highly correlated. The propensity scores of the DPN-SA exhibited a more polarized distribution toward the extremes as compared to DCN-PD, but not as marked as logistic regression and LASSO.

Table 1 shows the MSE results for all models (real and synthetic data). On the real dataset, the DPN-SA configured with the 25-20-10-20-25 layer stacking, trained in an end-to-end manner, yielded the best performance in terms of MSE, with an improvement of 6% over the DCN-PD, and 64% over the LASSO logistic regression. The one-neuron DPN-SA 25-1-25 configuration was better than LASSO, but worse

Dataset	# Covariates	Model	MSE (SD)	Bengio's p-value	Raw p-value
Real	25	**DPN-SA (25-20-10-) end-to-end**	**2.09 (0.22)**	Ref.	Ref.
		DPN-SA (25-20-10-) greedy st.	2.1 (0.2)	0.396	0.599
		DPN-SA (25-20-10-) greedy st. curr.	2.11 (0.2)	0.391	0.350
		DPN-SA (25-10-)	2.14 (0.22)	0.374	0.078
		DPN-SA (25-1-)	2.52 (0.37)	0.061	1.69e-16
		DCN-PD	2.22 (0.21)	0.295	1.54e-4
		Logistic Regression	6.02 (1.19)	0	1.01e-55
		LASSO Logistic Regression	5.89 (1.22)	0	6.03e-53
Real + Synthetic	75	DPN-SA (25-20-10-) end-to-end	3.14 (0.52)	0.096	1.13e-13
		DPN-SA (25-20-10-) greedy st.	2.7 (0.37)	0.397	0.689
		DPN-SA (25-20-10-) greedy st. curr.	**2.69 (0.35)**	Ref.	Ref.
		DPN-SA (25-10-)	3.52 (0.53)	0.02	2.74e-22
		DPN-SA (25-1-)	3.59 (0.54)	0.009	8.83e-26
		DCN-PD	2.71 (0.41)	0.396	0.648
		Logistic Regression	7.08 (1.27)	2.69e-10	1.56e-59
		LASSO Logistic Regression	7.35 (1.12)	1.36e-12	3.34e-65

Table 1: Performance of the models against IHDP dataset

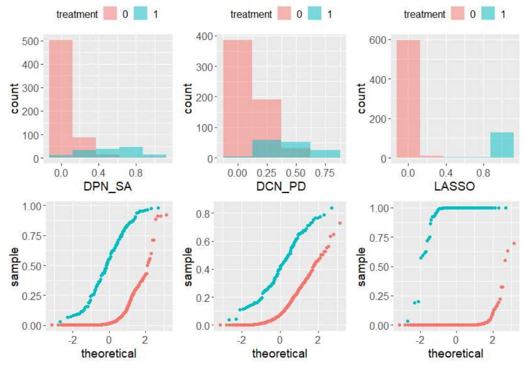

Figure 3: Histograms and quantile-quantile plots of the propensity score distributions (stratified by treatment group) for the DPN-SA, DCN-PD, and LASSO.

than all neural network-based classifiers, while the PCA-like DPN-SA 25-10-25 had performance comparable to all other networks. On the synthetic dataset, performance of all models decreased due to the artificial noise addition, but the regularization/sparsity constraints demonstrated to be robust against such noise and the additional correlated variables. The DPN-SA (25-20-10-20-25) with greedy stacked-current layer training was the best model, but only 1% better than the DCN-PD. Instead, the end-to-end training did not work as well as in the real dataset case. Overall, due to the limited sample size, the null hypothesis of no difference could not be rejected at the 0.05 significance level for DPN-SA architectures and DCN-PD (except for the simple 25-1-25 configuration).

Figure 5 shows the violin plots of each model's ATE estimation (real dataset), compared to the test set crude ATE. The DPN-SA (except the 25-1-25 configuration) showed

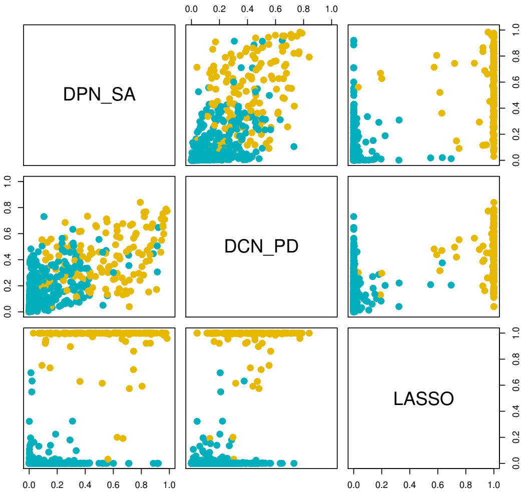

Figure 4: Scatterplots of the propensity score distributions (stratified by treatment group) comparing the DPN-SA, DCN-PD, and LASSO.

similar distributions to those yielded by the DCN-PD (with the end-to-end being the closest). Instead, logistic regression and LASSO were substantially different, exhibiting much a lower ATE.

When comparing training and prediction times, the fastest methods are logistic regression and LASSO in a obvious way. The models with autoencoders have taken a relatively larger time to train than normal neural network and logistic regression models. This is because in autoencoder models, first the dimensionality of the covariates have been reduced and then the classifier has been trained. The greedy layer-wise trained autoencoders took the longest time as we were training the autoencoder layer by layer and then stacked to-

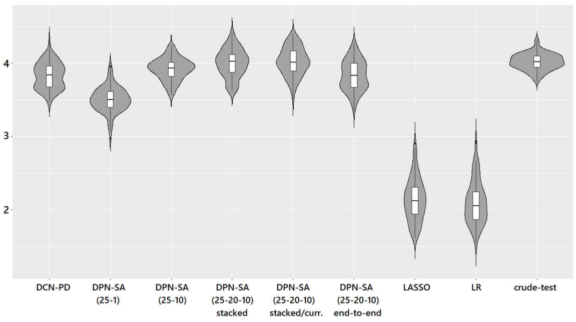

Figure 5: Violin plots of average treatment effect estimation.

gether one after.

4 Discussion

The DPN-SA architecture conjugates the ability to calculate nonlinear propensity scores with dimension reduction, and demonstrates advantage over other methods in MSE and treatment effect estimation. The multi-layer deep architecture also improves over less complex autoencoder configurations that are akin to LASSO and PCA. The advantage of DPN-SA over the DCN-PD, which also employs regularization, however is small and would need to be assessed on larger and more diverse datasets.

This work has some limitations. First, the choice of a *softmax* classifier as a replacement to the decoder is relatively simplistic, nonetheless, it provided lower MSE and lower-variance ATE. Other solutions for embedding the sparse autoencoder within the DCN-PD framework or within alternative approaches, such as ITE estimation with generalized adversarial networks (Yoon, Jordon, and van der Schaar 2018), could be devised. Another limitation is that the benchmark dataset has a limited sample size, and therefore the differences in average performance between models are subject to uncertainty. Finally, the distribution of propensity scores among treatment and control groups is often highly dependent on the dataset, and can be highly imbalanced, and therefore the results obtained with one experimental data set are not assured to be reproducible with others.

In conclusion, the DPN-SA provides a valid, possibly improved, alternative to DCN-PD and to more traditional PSM methods.

References

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

Alaa, A. M.; Weisz, M.; and Van Der Schaar, M. 2017. Deep counterfactual networks with propensity-dropout. *arXiv preprint arXiv:1706.05966*.

Austin, P. C. 2014. A comparison of 12 algorithms for matching on the propensity score. *Statistics in Medicine* 33:1057–1069.

Barron, A. R. 1994. Approximation and estimation bounds for artificial neural networks. *Machine learning* 14(1):115–133.

Bishop, C. M., et al. 1995. *Neural networks for pattern recognition*. Oxford university press.

Cho, K.; van Merrienboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the properties of neural machine translation: Encoder-decoder approaches.

Concato, J.; Shah, N.; and Horwitz, R. I. 2000. Randomized, controlled trials, observational studies, and the hierarchy of research designs. *New England journal of medicine* 342(25):1887–1892.

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

Garrido, M., et al. 2014. Methods for constructing and assessing propensity scores. *Health Services Research* 49(5):1701—20.

Greenland, S., and Morgenstern, H. 2001. Confounding in health research. *Annual review of public health* 22(1):189–212.

Hernan, M., and Robins, J. 2019. *Causal Inference*. Chapman & Hall/CRC Monographs on Statistics & Applied Probab. Taylor & Francis.

Hill, J. L. 2011. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics* 20(1):217–240.

Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *science* 313(5786):504–507.

Hinton, G. E., and Zemel, R. S. 1993. Autoencoders, minimum description length and helmholtz free energy. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, 3–10. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Imbens, G. W. 2000. The role of the propensity score in estimating dose-response functions. *Biometrika* 87(3):706–710.

Kurth, T.; Walker, A. M.; Glynn, R. J.; Chan, K. A.; Gaziano, J. M.; Berger, K.; and Robins, J. M. 2005. Results of Multivariable Logistic Regression, Propensity Matching, Propensity Adjustment, and Propensity-based Weighting under Conditions of Nonuniform Effect. *American Journal of Epidemiology* 163(3):262–270.

Lechner, M. 2001. Identification and estimation of causal effects of multiple treatments under the conditional independence assumption. In *Econometric evaluation of labour market policies*. Springer. 43–58.

Lemsara, A.; Ouadfel, S.; and Fröhlich, H. 2020. PathME: pathway based multi-modal sparse autoencoders for clustering of patient-level multi-omics data. *BMC Bioinformatics* 21(1):146.

Li, Q.; He, Z.; Guo, Y.; Zhang, H.; George, T. J.; Hogan, W.; Charness, N.; and Bian, J. 2020. Assessing the Validity of a a priori Patient-Trial Generalizability Score using Real-world Data from a Large Clinical Data Research Network: A Colorectal Cancer Clinical Trial Case Study. *AMIA ... Annual Symposium proceedings. AMIA Symposium* 2019:1101–1110.

Makhzani, A., and Frey, B. 2013. k-sparse autoencoders.

Mao, K.; Tang, R.; Wang, X.; Zhang, W.; and Wu, H. 2018. Feature representation using deep autoencoder for lung nodule image classification. *Complexity* 2018.

Mhaskar, H. N. 1996. Neural networks for optimal approximation of smooth and analytic functions. *Neural computation* 8(1):164–177.

Nadeau, C., and Bengio, Y. 2003. Inference for the generalization error. *Mach. Learn.* 52(3):239–281.

Ng, A., et al. 2011. Sparse autoencoder. *CS294A Lecture notes* 72(2011):1–19.

Pearl, J.; Glymour, M.; and Jewell, N. 2016. *Causal Inference in Statistics: A Primer*. Wiley.

Peters, J.; Janzing, D.; and Schölkopf, B. 2017. *Elements of causal inference*. The MIT Press.

Praveen, G.; Agrawal, A.; Sundaram, P.; and Sardesai, S. 2018. Ischemic stroke lesion segmentation using stacked sparse autoencoder. *Computers in biology and medicine* 99:38–52.

Prosperi, M.; Min, J. S.; Bian, J.; and Modave, F. 2018. Big data hurdles in precision medicine and precision public health. *BMC Medical Informatics and Decision Making* 18:139.

Prosperi, M.; Guo, Y.; Sperrin, M.; Koopman, J. S.; Min, J. S.; He, X.; Rich, S.; Wang, M.; Buchan, I. E.; and Bian, J. 2020. Causal inference and counterfactual prediction in machine learning for actionable healthcare. *Nature Machine Intelligence* 2:369–375.

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

Rosenbaum, P. R., and Rubin, D. B. 1984. Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American statistical Association* 79(387):516–524.

Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology* 66(5):688–701.

Rubin, D. B. 2005. Causal inference using potential outcomes. *Journal of the American Statistical Association* 100(469):322–331.

Schneeweiss, S.; Rassen, J. A.; Glynn, R. J.; Avorn, J.; Mogun, H.; and Brookhart, M. A. 2009. High-dimensional propensity score adjustment in studies of treatment effects using health care claims data. *Epidemiology (Cambridge, Mass.)* 20(4):512.

Sibbald, B., and Roland, M. 1998. Understanding controlled trials: Why are randomised controlled trials important? *BMJ* 316(7126):201.

Tian, Y.; Schuemie, M. J.; and Suchard, M. A. 2018. Evaluating large-scale propensity score performance through real-world and synthetic data experiments. *International journal of epidemiology* 47(6):2005–2014.

Yoon, J.; Jordon, J.; and van der Schaar, M. 2018. GANITE: estimation of individualized treatment effects using generative adversarial nets. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Yu, J.; Hong, C.; Rui, Y.; and Tao, D. 2017. Multitask autoencoder model for recovering human poses. *IEEE Transactions on Industrial Electronics* 65(6):5060–5068.

Zhang, H.; He, Z.; He, X.; Guo, Y.; Nelson, D. R.; Modave, F.; Wu, Y.; Hogan, W.; Prosperi, M.; and Bian, J. 2018. Computable Eligibility Criteria through Ontology-driven Data Access: A Case Study of Hepatitis C Virus Trials. *AMIA ... Annual Symposium proceedings. AMIA Symposium* 2018:1601–1610.

Zhavoronkov, A.; Ivanenkov, Y. A.; Aliper, A.; Veselov, M. S.; Aladinskiy, V. A.; Aladinskaya, A. V.; Terentiev, V. A.; Polykovskiy, D. A.; Kuznetsov, M. D.; Asadulaev, A.; Volkov, Y.; Zholus, A.; Shayakhmetov, R. R.; Zhebrak, A.; Minaeva, L. I.; Zagribelnyy, B. A.; Lee, L. H.; Soll, R.; Madge, D.; Xing, L.; Guo, T.; and Aspuru-Guzik, A. 2019. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nature Biotechnology* 37(9):1038–1040.

Zhou, Y.; Arpit, D.; Nwogu, I.; and Govindaraju, V. 2014. Is joint training better for deep auto-encoders?