Deep Propensity Network using a Sparse Autoencoder for Estimation of Treatment Effects

Shantanu Ghosh1, Jiang Bian, Ph.D.2, Yi Guo, Ph.D.2 Mattia Prosperi, Ph.D.3

1Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA; 2Department of Health Outcomes and Biomedical Informatics, College of Medicine, University of Florida, Gainesville, Florida, USA; 3Department of Epidemiology, College of Public Health and Health Professions & College of Medicine, University of Florida, Gainesville, Florida, USA

Abstract

*Understanding causality is critical, especially in biomedical and social science; and the gold-standard solution is to perform randomized controlled experiments, enacting interventional probabilities as opposed to conditional probabilities. However, randomized experiments are not always 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 a 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 typically 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 problems of high dimensionality and residual confounding. 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/*](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.[1] 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 or 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 (RCTs). [2,3] In a 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). Because of the randomization process, it leads to strong ignorability of individuals’ pre-treatment characteristics, and thus the causal effect of the treatment versus control can be evaluated.[4] 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[5,6] because only one factual outcome can be observed (a person cannot be assigned to both treatment and control groups).

Since RCTs are not always feasible due to ethical or operational constraints, e.g. conducting a RCT to ask individuals to smoke and then assess the effect of smoking toward the development of lung cancer, observational data are often used in attempts to draw these causal conclusions. Nevertheless, when using observational data, one must account for the possible types of underlying bias, including confounders and colliders.[7] 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.[4,8] The propensity score is a scalar estimate representing the conditional probability of receiving a certain treatment , versus the control group or no treatment , given a set of measured pre-treatment covariates denoted as

|  |  |  |
| --- | --- | --- |
|  |  | (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, traditional PSM approach accounts only for measured (and measurable) covariates, and latent bias may remain after matching.[9]

PSM has been implemented historically through logistic regression, which calculates the probability of treatment assignment given the pre-treatment covariates.[10] In presence of high-dimensional datasets, e.g., compiled from large electronic health record (EHR) databases,[11] different feature selection methods within PSM have been employed, such as the high-dimensional propensity score (hdPS)[12] or L1-logistic regression.[13] 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 difficult to explicitly define 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.[14–16] 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 in logistic regression-based PSM approach. A neural network can be built to provide the estimation of treatment group assignment probability (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),[17] where a feed-forward network with a set of shared layers is used to calculate the counterfactual outcomes, but being regularized by another network that calculates propensity scores and a dropout probability for training examples.[18] Through alternating training phases (treatment vs. control groups), both 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 estimates of 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[19] 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 original DCN-PD network layout.

2 Methodology

2.1 Problem Formulation

Let assume a population sample (independent and identically distributed) of individuals, given background set of pre-treatment covariates a treatment (binary, for simplicity) and a health outcome . Each subject is represented by the tuple {*Xi , Ti , Yi* }. Let *Yi0* and *Yi*1 the potential outcome for individual *i* under treatment *Ti* = 0and *Ti* = 1, respectively.[20,21] Given *Xi* = *x*, the ITE is defined as the difference in the mean potential outcomes for the individual under both treatments, conditional on the observed covariate vector

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

The ITE formulation as 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), the potential outcomes are independent of treatment conditional on background variables, i.e. .[5,22–24] Under the assumption of SITA, the ITE can then be calculated as

Further, under SITA and by averaging over the distribution of , the ATE can be calculated as

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

However, by assuming SITA, ITE and ATE can be calculated only with being the same in treatment groups, which becomes quickly unfeasible when the dimension of grows. PSM, through the conditional probability (see eq. 1) attempts at balancing the probability of receiving given . 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,[25] and propensity weighing[26] (on which the DCN-PD was based).

2.2 Proposed Approach

The DCN-PD method[17] offered a novel and effective way to calculate nonlinear propensity scores as well as to match individuals, providing better ITE and 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 databases.[27,28] Feature selection and shrinkage has been proven effective for PSM by logistic regression[12,13] and in the same way dimensionality reduction can be implemented in deep neural architectures.[29] 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).[30] In its simplest form, 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.[31] Autoencoders have been employed in several applications, from machine translation to drug discovery.[32,33] 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.[19,34] It has been also broadly applied in biomedical studies, including imaging and -omics datasets.[35–38]

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



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

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 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.

Let represent the population of $x(i)$ individuals i.i.d samples). Each is a dimensional vector, where represents the number of pre-treatment covariates, i.e. the number of features. The DPN-SA uses up to encoder and 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. As , the reconstructed output also needs to be . Supposing we set , the sample input covariate vector $x(i)$ is reconstructed in the forward propagation as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (4) |
|  |  | (5) |
|  |  | (6) |
|  |  | (7) |
|  |  | (8) |
|  |  |  |

where, are the activations of the encoder, decoder and the reconstructed input of layer for the sample, respectively. and are the weight matrices of encoder 1 and encoder 2, having sizes and , and are the weight matrices of decoder 1 and decoder 2 having sizes and . denotes size or number of neurons of layer and is the activation function ().

After reconstructing the a sample input feature , the objective function of the sparse autoencoder has to be minimized, as described in:[19]

|  |  |  |
| --- | --- | --- |
|  |  | (9) |
|  |  | (10) |

where is the regularization parameter, is the weight matrix corresponding to layer of the network, is the number of neurons in the hidden layer, is the weight of the sparsity penalty, is the Kullback-Leibler divergence, and the subscript is the Frobenius norm (equivalent to the squared norm of the weight matrix). The is minimized using backpropagation, for number of epochs.



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

The DPN-SA is trained in two phases. In the first phase the sparse autoencoder is trained and optimized ( epochs). The parameters () 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 Figure 2. The *softmax* classifier is trained for number of epochs. The final network gives the estimation of the propensity score . Algorithm 1 describes the two-phase procedures to obtain the final DPN-SA.

2.4 Experimental Setup

Datasets. We used the Infant Health and Development Program (IHDP) dataset, a multi-site, longitudinal, RCT 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 original IHDP dataset was resampled by throwing away a nonrandom subset of the treatment group (based on the ethnicity variable), thus inducing treatment imbalance, and then counterfactual outcomes were simulated using either a linear or nonlinear/nonparallel surface, thus knowing the true ITE/ATE.[39] The nonlinear surface dataset, which we chose as benchmark for this study, consisted of 747 subjects (139 in the treatment group and 608 controls), with 25 associated covariates, describing characteristics of the infants and their mothers (excluding the ethnicity).

|  |  |
| --- | --- |
|  | **Algorithm** **1** Training DPN-SA |
|  | **Input:** Batch *B* of random samples with assigned treatment T training set , number of epochs , learning rates for the sparse autoencoder (SA) and the *softmax* classifier, respectively. |
|  | **Output:**  consisting of propensity score for each sample *i*, where *i* ranges from to *N*. |
| 1: | **Procedure** DPN-SA training |
| 2: | Initialize of the SA. |
| 3: | **for** epochs = 1, 2, . . . **do** |
| 4: | **for** in |
| 5: | Compute using forward propagation. |
| 6: | Compute . |
| 7: | Compute Gradient: |
| 8: |  |
| 9: | **end for** |
| 10: | **end for** |
| 11: | Remove the decoder from the SA |
| 12: | Attach a *softmax* classifier to the last encoder |
| 13: | Initialize of the classifier |
| 14: | $ Empty |
| 15: | **for** epochs = 1, 2, . . . **do** |
| 16: | **for** in |
| 17: |  |
| 18: | Get the propensity score and add to |
| 19: | **end for** |
| 20: | **end for** |

In addition to IHDP, we also created a synthetic dataset with larger covariate space, first by doubling the original IHDP feature set through the creation of 25 random variables (shuffling the original ones), and then triplicating it with another set of 25 covariates partially correlated to the original ones (approx. ), using a Gaussian noise addition to each original variable . The factual and counterfactual outcomes matched those of IHDP.

DPN-SA and Other Comparison Methods. Four different configurations of the DPN-SA were tested --on both the 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 (), sparsity parameter (), and sparse penalty () to 0.0003, 0.8,

and 0.1, respectively, with a batch size of 32. The *softmax* classifier was ran for 50 epochs () 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 ran 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 and ATE in the same way as.[17] The error distributions were compared by means of a t-test with sample overlap adjustment.[40]

A close up of text on a white background

Description automatically generated

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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

3 Results

Figures 3 and 4 show the histogram, quantile-quantile, and scatterplots of the 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 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 ATE. The DPN-SA (25-20-10-20-25) with greedy stacked-current configuration showed the closest

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

resemblance to the test set ATE, followed by the end-to-end and the (25-10-25) configurations that were similar to the DCN-PD. The DPN-SA (25-1-25) significantly underestimated the test set ATE but was closer to the neural architectures than both logistic regression

and LASSO that exhibited much a lower ATE. Note that the IHDP surface is nonlinear by design, so any linear estimator would be biased.

When comparing training times, the fastest methods were logistic regression and LASSO, followed by the DCN-PD. The DPN-SA takes more time to train than the DCN-PD, with the stacked configuration being slower than the end-to-end.

A screenshot of a cell phone

Description automatically generated

Figure 5: Violin plots of average treatment effect estimation

4 Discussion

The DPN-SA architecture conjugates the ability to calculate nonlinear propensity scores with dimension reduction and demonstrates advantage over other methods in treatment effect estimation. In the IHDP counterfactual datasets, the response surface was made nonlinear and nonparallel across treatment groups, thus the true ATE could not be estimated by means of a single linear model.

The DPN-SA architecture allows flexibility in configurations, from the simple 1-neuron akin to LASSO, to the single-layer PCA-like, to the multi-layer setup. All the DPN-SA multi-layer configurations gave ATE estimates close to the true ones, and even the simpler configurations yielded ATE better than the linear models. However, the advantage in MSE of DPN-SA over the DCN-PD, which also employs regularization, is small and would need to be assessed on larger and more diverse datasets. The DPN-SA might be preferable because of its latent space encoding that can be directly chosen and compared (e.g. linear vs. PCA vs. more complex nonlinear setup).

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,[41] 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, deep learning frameworks for propensity score estimation and treatment effect prediction are particularly suited for EHR-based studies, not only because such studies can include large sets of covariates, but also because there is high chance of complex heterogeneity in treatment assignments. In these cases, regularized linear propensity score methods, e.g. hdPS[12] or LASSO[13] would not be able to provide reliable estimates and yield biased ITE/ATE. The DPN-SA provides a valid, possibly improved, alternative to DCN-PD, and to more traditional PSM methods.

References

1 Prosperi M, Guo Y, Sperrin M, *et al.* Causal inference and counterfactual prediction in machine learning for actionable healthcare. *Nat Mach Intell* 2020;**2**:369–75. doi:10.1038/s42256-020-0197-y

2 Sibbald B, Roland M. 24 Herman JL. Histories of violence in an outpatient population: an exploratory study. 1994.

3 Concato J, Shah N, Horwitz RI. Randomized, controlled trials, observational studies, and the hierarchy of research designs. *N Engl J Med* 2000;**342**:1887–92. doi:10.1056/NEJM200006223422507

4 Rosenbaum PR, Rubin DB. The central role of the propensity score in observational studies for causal effects. 1083. https://academic.oup.com/biomet/article-abstract/70/1/41/240879 (accessed 21 Jul 2020).

5 Pearl J, Glymour M, Jewell NP. Counterfactuals and their applications. *Causal Inference Stat A Prim* 2016.

6 Hernán MA, Robins JM. Causal Inference Part1. *Causal Inference* Published Online First: 2010. doi:10.1177/1745691610383511

7 Greenland S, Morgenstern H. Confounding in Health Research. *Annu Rev Public Health* 2001;**22**:189–212. doi:10.1146/annurev.publhealth.22.1.189

8 Rosenbaum PR, Rubin DB. Reducing Bias in Observational Studies Using Subclassification on the Propensity Score. *J Am Stat Assoc* 1984;**79**:516. doi:10.2307/2288398

9 Garrido MM, Kelley AS, Paris J, *et al.* Methods for Constructing and Assessing Propensity Scores. doi:10.1111/1475-6773.12182

10 Kurth T, Walker AM, Glynn RJ, *et al.* Results of multivariable logistic regression, propensity matching, propensity adjustment, and propensity-based weighting under conditions of nonuniform effect. *Am J Epidemiol* Published Online First: 2006. doi:10.1093/aje/kwj047

11 Prosperi M, Min JS, Bian J, *et al.* Big data hurdles in precision medicine and precision public health. doi:10.1186/s12911-018-0719-2

12 Schneeweiss S, Rassen JA, Glynn RJ, *et al.* High-dimensional propensity score adjustment in studies of treatment effects using health care claims data. *Epidemiology* Published Online First: 2009. doi:10.1097/EDE.0b013e3181a663cc

13 Tian Y, Schuemie MJ, Suchard MA. Evaluating large-scale propensity score performance through real-world and synthetic data experiments. *Int J Epidemiol* 2018;**47**:2005–14. doi:10.1093/ije/dyy120

14 Barron AR. Approximation and Estimation Bounds for Artificial Neural Networks. *Mach Learn* Published Online First: 1994. doi:10.1023/A:1022650905902

15 Bishop CM. *Pattern recognition and machine learning / Christopher M. Bishop.* 2006. doi:10.1117/1.2819119

16 Mhaskar HN. Neural Networks for Optimal Approximation of Smooth and Analytic Functions. *Neural Comput* Published Online First: 1996. doi:10.1162/neco.1996.8.1.164

17 Alaa AM, Weisz M, van der Schaar M. Deep Counterfactual Networks with Propensity-Dropout. Published Online First: 2017.http://arxiv.org/abs/1706.05966

18 Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: *33rd International Conference on Machine Learning, ICML 2016*. 2016.

19 Ng A. CS294A Lecture notes Sparse autoencoder.

20 Rubin DB. Estimating causal effects of treatments in randomized and nonrandomized studies. *J Educ Psychol* 1974;**66**:688–701. doi:10.1037/h0037350

21 Rubin DB. Causal inference using potential outcomes: Design, modeling, decisions. *J Am Stat Assoc* 2005;**100**:322–31. doi:10.1198/016214504000001880

22 Imbens GW. The role of the propensity score in estimating dose-response functions. *Biometrika* Published Online First: 2000. doi:10.1093/biomet/87.3.706

23 Lechner M. Identification and estimation of causal effects of multiple treatments under the conditional independence assumption. 2001. doi:10.1007/978-3-642-57615-7\_3

24 Janzing D, Schölkopf B. Elements of Causal Inference Foundations and Learning Algorithms.

25 Austin PC. A comparison of 12 algorithms for matching on the propensity score. *Stat Med* 2014;**33**:1057–69. doi:10.1002/sim.6004

26 Abadie A, Imbens GW. Matching on the Estimated Propensity Score. *Econometrica* 2016;**84**:781–807. doi:10.3982/ecta11293

27 Zhang H, He Z, He X, *et al.* Computable Eligibility Criteria through Ontology-driven Data Access: A Case Study of Hepatitis C Virus Trials. *AMIA . Annu Symp proceedings AMIA Symp* 2018.

28 Li Q, He Z, Guo Y, *et al.* 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.

29 Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science (80- )* Published Online First: 2006. doi:10.1126/science.1127647

30 Hinton GE, Zemel RS. Autoencoders, Minimum Description Length and Helmholtz free Energy. *Adv Neural Inf Process Syst* Published Online First: 1994. doi:10.1021/jp906511z

31 Zhou Y, Arpit D, Nwogu I, *et al.* Is Joint Training Better for Deep Auto-Encoders?

32 Cho K, van Merrienboer B, Bahdanau D, *et al.* On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. 2015. doi:10.3115/v1/w14-4012

33 Zhavoronkov A, Ivanenkov YA, Aliper A, *et al.* Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat Biotechnol* Published Online First: 2019. doi:10.1038/s41587-019-0224-x

34 Makhzani A, Frey B. k-Sparse autoencoders. In: *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*. 2014.

35 Yu J, Hong C, Rui Y, *et al.* Multitask Autoencoder Model for Recovering Human Poses. *IEEE Trans Ind Electron* Published Online First: 2018. doi:10.1109/TIE.2017.2739691

36 Mao K, Tang R, Wang X, *et al.* Feature Representation Using Deep Autoencoder for Lung Nodule Image Classification. *Complexity* Published Online First: 2018. doi:10.1155/2018/3078374

37 Praveen GB, Agrawal A, Sundaram P, *et al.* Ischemic stroke lesion segmentation using stacked sparse autoencoder. *Comput Biol Med* Published Online First: 2018. doi:10.1016/j.compbiomed.2018.05.027

38 Lemsara A, Ouadfel S, Fröhlich H. PathME: Pathway based multi-modal sparse autoencoders for clustering of patient-level multi-omics data. *BMC Bioinformatics* Published Online First: 2020. doi:10.1186/s12859-020-3465-2

39 Hill JL. Bayesian nonparametric modeling for causal inference. *J Comput Graph Stat* Published Online First: 2011. doi:10.1198/jcgs.2010.08162

40 Nadeau C, Bengio Y. Inference for the Generalization Error.

41 Yoon J, Jordon J, Van Der Schaar M. GANITE: ESTIMATION OF INDIVIDUALIZED TREAT-MENT EFFECTS USING GENERATIVE ADVERSARIAL NETS.