Heterogeneous Treatment Effect Estimation through Deep Learning

Ran Chen* Hanzhong Liu†

Abstract

Estimating heterogeneous treatment effect is an important task in causal inference with wide application fields. It has also attracted increasing attention from machine learning community in recent years. In this work, we reinterpret the heterogeneous treatment effect estimation and propose ways to borrow strength from neural networks. We analyze the strengths and drawbacks of integrating neural networks into heterogeneous treatment effect estimation and clarify the aspects that need to be taken into consideration when designing a specific network. We proposed a specific network under our guidelines. In simulations, we show that our network performs better when the structure of data is complex, and reach a draw under the cases where other methods could be proved to be optimal.

Key Words: Heterogeneous Treatment Effect, Causal Inference, Machine Learning, Deep Learning, Neural Networks

1. Introduction

Estimating heterogeneous treatment effect is a task commonly encountered in economic, marketing, public policy, and personalized medicine. With the growing accessibility of data of various forms, it is possible to have more accurate heterogeneous treatment estimation of individuals based on the observable covariates.

1.1 Definitions and Problem Setting

Following the Neymann-Rubin Potential Outcome framework [6], we have two treatment conditions, $T_i = 1$ for treated and $T_i = 0$ for control, two corresponding potential outcomes, $Y_i(1)$ for treated and $Y_i(0)$ for control, and we can only observe one of the potential outcomes. Therefore, the observed data is (Y_i, X_i, T_i), where X_i is covariates. We assume our observed data come from a super probabiity distribution, that is $(Y_i, X_i, T_i) \overset{\text{iid}}{\sim} \mathbb{P}$.

Our goal is to estimate Individual Treatment Effect (ITE), $Y_i(1) - Y_i(0)$, as accurately as we can when we are given a new item with X_i observed. But this is not possible without strong assumptions [4] . A naive choice is to estimate the Average Treatment Effect (ATE) $\tau = \mathbb{E}(Y_i(1) - Y_i(0))$ as a substitute for $Y_i(1) - Y_i(0)$. However, this approach does not consider the individual properties given in X_i. Another way people often use is to estimate Conditional Average Treatment Effect (CATE) $E(Y_i(1) - Y_i(0)|X_i = x)$ as a substitute for $Y_i(1) - Y_i(0)$, using information in X_i. We use this more reasonable and widely used way in this work.

However, it should be noticed that X_i is what we observed, so that it does not necessarily carry enough pertinent information about $Y_i(1) - Y_i(0)$ due to two facts: 1, We cannot assert we can observe all the measures related to $Y_i(1) - Y_i(0)$; 2, X_i can be very noisy and noise does not tell us much about the $Y_i(1) - Y_i(0)$. Therefore, the distribution of $(Y_i(1) - Y_i(0)|X_i = x)$ could be heavy tail due to the noise, or only have a small concentration around the likely $Y_i(1) - Y_i(0)$ due to missing information. In these cases, which

*Department of Statistics, The Wharton School, University of Pennsylvania

†Center for Statistical Science and Department of Industrial Engineering, Tsinghua University, Beijing, 100084, China

are very likely, CATE is not a good substitute for Individual Treatment Effect, conditional mode may be a better choice. We will elaborate on this more later in a specific case, but for now, what's under consideration is CATE.

To make CATE estimable, we need some conditions:

- Unconfoundedness: $T \perp\!\!\!\perp (Y(1), Y(0))|X$,
- Strict Overlap: $\exists \epsilon \in (0, 1), P(T = 1|X) \in [\epsilon, 1 - \epsilon]$.

1.2 Related Works

Many endeavors have been made to borrow strength of machine learning methods for estimating various treatment effects. Two main streamlines are linear regression based methods and tree based methods.

Linear regression based methods include unadjusted regression, which is regression on treatment indicator; adjusted regression[3], which is regression on both treatment indicator and covariates; and adjusted regression with interaction term[5], which is regression on treatment indicator vector, covariates and interaction term of treatment indicator vector and covariates.

Tree based methods include Causal Tree [1], which integrated the notion of treatment effect into the tree building and fitting procedure and is designed to estimate the treatment effect; Causal Forest [7], which ensembles Causal Trees and can do both estimation and inference; and Meta-learners[4], a set of methods integrating treatment indicator into machine learning methods, mostly tree based methods.

Aforementioned work have also validated their methods theoretically under specific settings, mostly assume the truth is a regression model and enjoys Lipschitz conditions or certain degree of smoothness.

However, with the growing accessibility of data of various form, the structure of data could be complex. In target marketing, data includes user's profile photo, posts, demography information and purchase behaviors. In a medical setting, observable covariates are not restricted to blood pressure, hemogram, gender, image data like CT scan can also be collected. Therefore, data can be of very high dimension, can has complex structure, and can be very noisy. The simple models may fail to be representative in a lot of applications as mentioned above.

1.3 Reinterpretation of the problem

The problem of heterogeneous treatment effect estimation is defined on some model assumptions that captures the essentials of the real world problem, but with concessions and approximations. All the aforementioned methods are by nature a designation of a computation procedure, resulting in an output, no matter what model assumptions are. Therefore, the task becomes, coming up with a computation procedure that can result in an output "closer", as measured by mean squared error (MSE) or other criterion, to "truth" or its proxy, which is CATE in our problem setting, under the model assumptions we defined above. However, the model assumptions of the problem is too general for this goal to be fully achieved, a reasonable goal is to achieve good performances in a reasonable wide arrange of settings, among which, regression models with Lipschitz conditions are only a small part.

Some preferable characteristics of such methods are:

- Can fit different models well without knowing the model specification,

- Can achieve consistency when sample size grows to infinity,
- Computationally feasible.

The second is hard to achieve theoretically when the first is achieved empirically, as the setting could be complex enough to resist statistical and mathematical analysis. The third is also in conflict with the second one to some extent, when one goes for optimal convergence rate in some problems [2]. Therefore, in this work, we do not put any of the criterion as dominant, all the three are important.

2. Causal Networks

2.1 Integrating Neural Network into Heterogeneous Treatment Effect Estimation

Neural network has proved to be successful in image tasks and text tasks, due to its amazing expressiveness and the ability to deal with structured data, which motivates borrowing its strength into heterogeneous treatment effect estimation.

A working neural network has two basic elements: network structure and training paradigm.

The neural network structure is normally composed of input layer, several hidden layers and an output layer. Each Layer is composed of operators: linear transformation or piecewise linear transformation (max pooling), activation function, and batch normalizing (stabilize the input data in each layer). One can have an output with a neural network when parameters are fixed. With a given pre-specified loss function, the aim is to optimize the loss function, where the training paradigm comes into consideration. Common training paradigms are back propagation based training paradigm, Adam and SGD are popular ones.

2.1.1 Expressiveness of Neural Networks

Neural networks can express a wide range of relationships through a specific realization of parameters. For example, only a linear transformation operator is enough to express linear relationship, as could be expressed in linear regression based methods. When there are more layers with activation functions, whether linear or not, linear relationship can also be approximated as long as the activation function is differentiable on an interval, mean while, nonlinearity could also be detected as long as the activation function is second-order differentiable. Our exploratory simulation shows that two layer neural network with nonlinear activation function can detect nonlinearity (see appendix).

Though neural network output is a continuous function of input, the step function could be approximated by two Relu (a kind of activation function) with opposite direction. Forest structure with a given number of trees can also be approximated by adding an averaging layer (which is a linear transformation) in the end.

The expressiveness of neural network makes it possible to incorporate several methods in one network structure, the conventional model selection procedure is some what incorporated in the training procedure w.r.t. minimizing the loss function. The notion of model selection in neural network setting becomes selecting a neural network structure.

2.1.2 Computational Burden

A key issue in exploiting neural network based method is it's computational burden — it does not compute the exact solution and the computation time could be long.

Computation time primarily depends on the number of nodes and complexity of the function used in that node, so moderately large neural networks do not require much time for one updating procedure and it is linear to sample size. On the other hand, for random forest type methods, the computation time depends both on number of trees in the forest and the tree fitting procedure, both of which commonly grow with sample size. Besides, with the development of neural network in computer science, chips for computing neural networks are under development, in which situation the computation of basic operators becomes a single instruction like "==" and "!=". The problem of computational time is not an issue then.

What can not be offset by the recent development in computer hardware is that the problem is not exact solution. In our exploratory simulation, we find that with increasing iteration and access to new data, this problem can be allieviated. The problem exists, but to mild extent.

2.1.3 End to End Approach

Another characteristic coming along is that this is an end to end approach, this gives the method a huge space for accommodating to different data form. Data of text form, image form, traditional covariates could be considered together in one neural network. Though images data normally use CNN based neural network, text data normally use RNN based network, two kinds of network structure could be connected and merged into a big one and which is not the case for other methods.

2.1.4 Utilizing Treatment Group/ Control Group Information

A key task in integrating neural network in heterogeneous treatment effect estimation is figuering out how to use the information in treatment group and control group.

Treatment group and control group shares a part of information, as the data of the two are of the same form and the outcome depends not only on what treatment is given but also on which individual it is. Two groups also has separate information due to the different treatment, which is what we want to get. How to achieve this information sharing and information separation is a key question.

Diverter

Here we design a "diverter", which "diverts" the information flow of two groups (though our way of integrating treatment information does not restrict treatment indicator to be binary, it could be continuous):

- Compute according to first several layers of network with covariate X_i to be the input, getting $\mathbf{f}(X_i)$,
- Control Flow diverter:
  Diverting mechanism: $\mathbf{f_c}(X_i) = max(1 - sigmoid(max(0, \mathbf{f}(X_i) + T_i)), 0)$
  Keep going after diverting : $\tilde{\mathbf{f_c}}(X_i) = \mathbf{g_c} \circ \mathbf{f_c}(X_i)$
- Treatment Flow diverter:
  Diverting mechanism: $\mathbf{f_t}(X_i) = sigmoid(max(0, \mathbf{f}(X_i) + T_i - 1))$
  Keep going after diverting: $\tilde{\mathbf{f_t}}(X_i) = \mathbf{g_t} \circ \mathbf{f_t}(X_i)$
- Merge two flows:
  Adding the two flow up to be the output:
  $\hat{y}(X_i) = \tilde{\mathbf{f_c}}(X_i) + \tilde{\mathbf{f_t}}(X_i)$

Intuition of diverter

The intermediate output $\mathbf{f}(X_i)$ has reasonable range, adding the treatment indicator separate the range of control group and treatment group to some extent, so it is possible to extract the part with "mostly control group", and the part with "mostly treatment group", resulting in control flow diverter and treatment flow diverter. After the flow are diverted, they keep going on their computation or transformation. In the end, they are added up together, two flows merges.

With diverter, both information sharing and information separation are incorporated and the extent of information separation are learned through training, as we do not impose a hard separation.

2.2 Our Causal Network

Under the guidance we discussed above, we designed a simple neural network with a diverter, as shown in figure 1. The network is composed of a input block, a diverter, three processing block and an output block. This neural network is simpler than the simplest standard neural network, like Alexnet. Parameters in each convolution layer is only the convolution kernel, which is shared across different location of the image tensor.

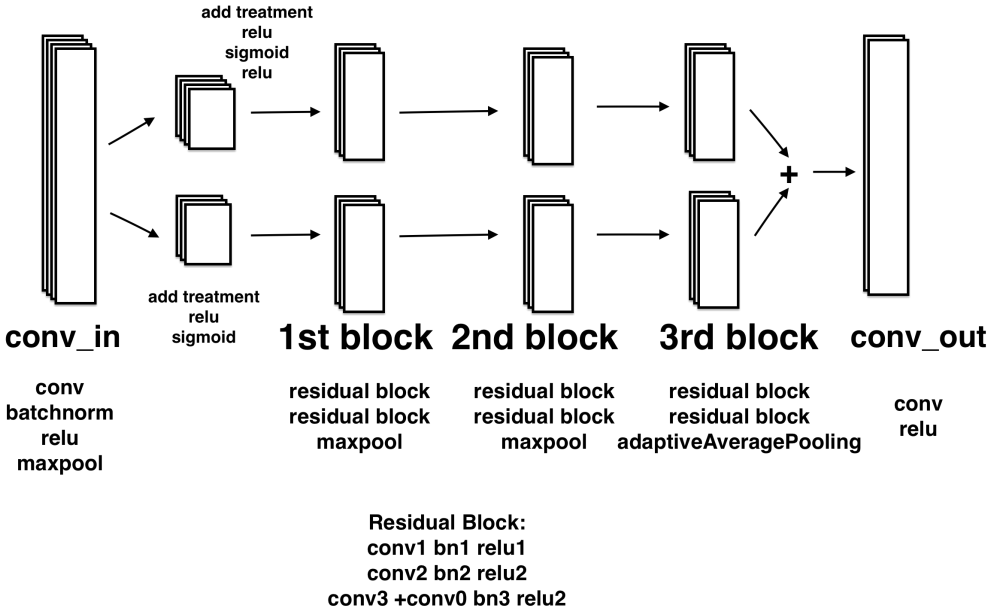

Figure 1: Causal Network

We use standard SDG for our training procedure. And we use L2 loss function of the outcome to train our network. Our philosophy here is to achieve good performance on CATE estimation through good performance on outcome estimation.

3. Simulation

In this section, we test Causal Network on various settings and compare it with other methods, namely adjusted regression, adjusted regression with interaction term, S-learner with standard random forest [4] and T-learn with standard random forest [4].

The first part is on image data with treatment effect being related to its topological structure, where we try to make the input similar to tumors. We show that our causal network is able to detect the treatment effect information incorporated into the image through topological structures, while all other methods totally fail. We also discussed a little bit on the criterion of CATE in the setting where X is noisy, which is very likely in reality.

The second part is devoted to the setting where data is generated according to the following schemes: linear, polynomial, given trees, and given neural networks, some of which provably favor linear regression based methods or tree based methods. The covariate is still image, which is of high dimension when seen as a vector, and signal to noise ratio is less than 10. Therefore, these tasks are very difficult. Our results show that all the methods do not perform well, and their performance are of the same order.

We also choose difference sample sizes for training data to see how the performance vary with sample size. Testing dataset is always the same one with sample size 10000, which is independent to training dataset.

3.1 Image

3.1.1 Noiseless Covariate

Simulation Setting

Data generating scheme is as follows:

- X is an 32 * 32 image containing a circle with radius R and center at the origin O,
- Radius $R \sim \text{Unif}(0, 16)$,
- Origin $O \sim \text{Unif}(0, 32) * (0, 32)$, and $O \perp\!\!\!\perp R$,
- For pixels inside the circle defined above, pixel=180,
- For pixels outside the circle defined above, pixel =0,
- Potential Outcomes $Y(0) \sim N(0, 1)$, $Y(1) \sim N(R, 1)$,
- $T \sim B(1, 0.5)$ independent of (X, Y(0),Y(1)).

Figure 2 illustrates how the covariates (images) look like. In this case, X contains the pertinent noiseless information about the treatment effect, and $\mathbb{E}(Y(1) - Y(0)|X) = R$.

We chose training sample size to be 2000,4000,6000,8000,10000, and test sample size to be 10000.

Results

For training sample size equal to 10000, we plot estimation versus true treatment effect on training dataset and test data set. The plots of Causal Net, S-learner, T-learner, adjusted regression with interaction term are shown in figure 3, figure 4, figure 5 and figure 6. Adjusted regression without interaction term's estimation is 8.040767. We can see that Causal Networks performs better than other methods, where other methods are basically predicting the mean.

When sample size varies, MSE for different methods on both training and testing set are shown in figure 7 (for adjusted regression with interaction term, collinearity exists). Since the true treatment effect is uniform on (0,16), it has variance $\frac{64}{3}$, which is about 21.3, and through checking the estimation versus true value plots for different methods on different training sample size, other methods' estimations are almost independent of the true treatment effect.

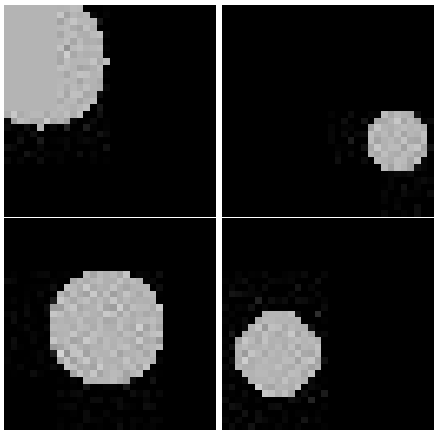

Figure 2: Noiseless Images

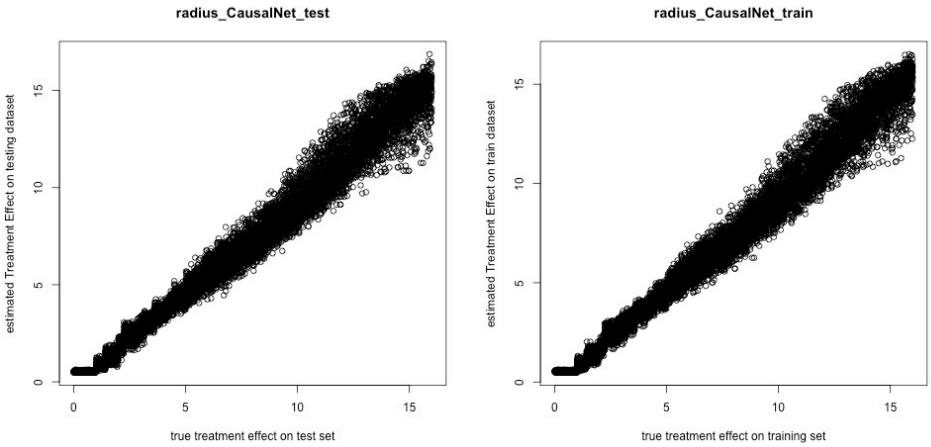

Figure 3: Causal Network on Noiseless data with training sample size 10000

3.1.2 Noisy Covariate

Simulation Setting

Covariates in the previous simulation does not have noise irrelevant to heterogeneous treatment effect. In the real settings however, images are polluted with information that is irrelevant to heterogeneous treatment effect. In this simulation, we add Guassian noise to see how different methods work.

However, we notice that $E(Y(1)-Y(0)|X)$ is not necessarily R in this situation, and it changes continuously with the variance of noise. In this situation, the only information that images contain about treatment effect is R, so when we are given a new individual, what we really what to get is still R. Our substitute for individual treatment effect, CATE, therefore is not a good substitute when the variance of noise is large. Reasons are as follows. Though for R far away from the truth and whatever origin, the probability is smaller than that of the true R and O, the conditional probability of a true R and absurd origin is also small. Since most origins are absurd, advantage of the true R over absurd R is decreased with respect to conditional probability, which is further decreased when conditional expectation is taken. In a very noisy setting, taking the conditional mode to be the substitute may be a better choice.

Detailed data generating scheme is as follows:

- X is an 32 * 32 image containing a circle with radius R and center at the origin O,

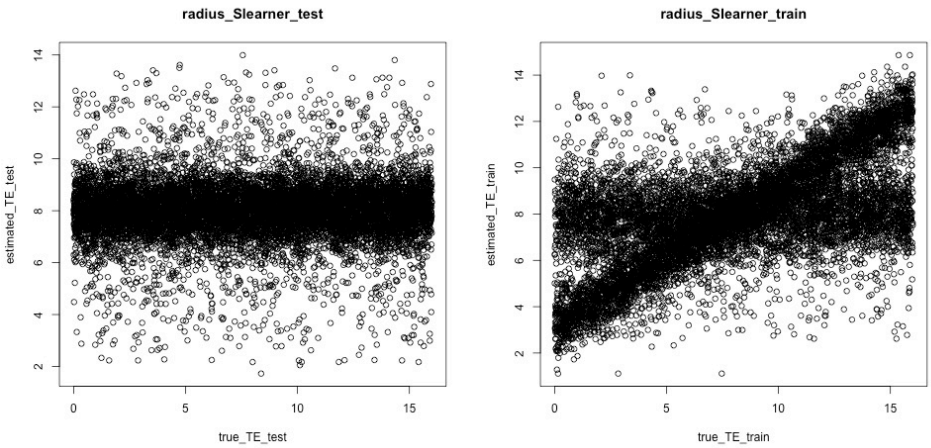

Figure 4: S learner on Noiseless data with training sample size 10000

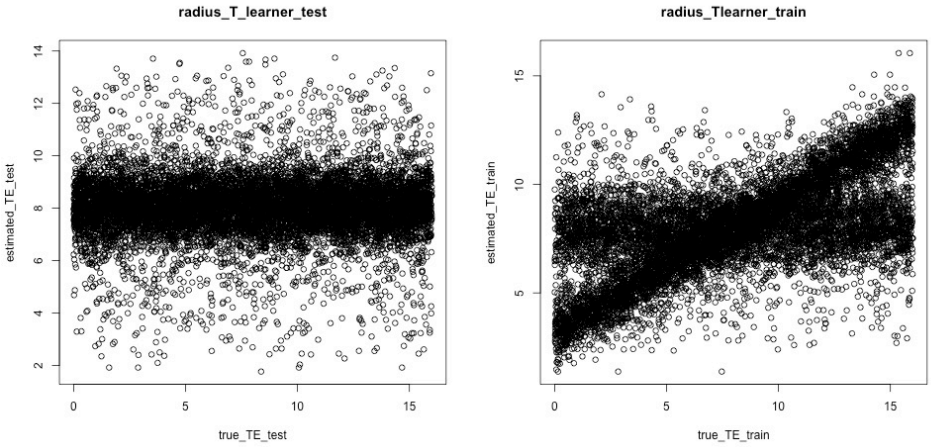

Figure 5: T learner on Noiseless data with training sample size 10000

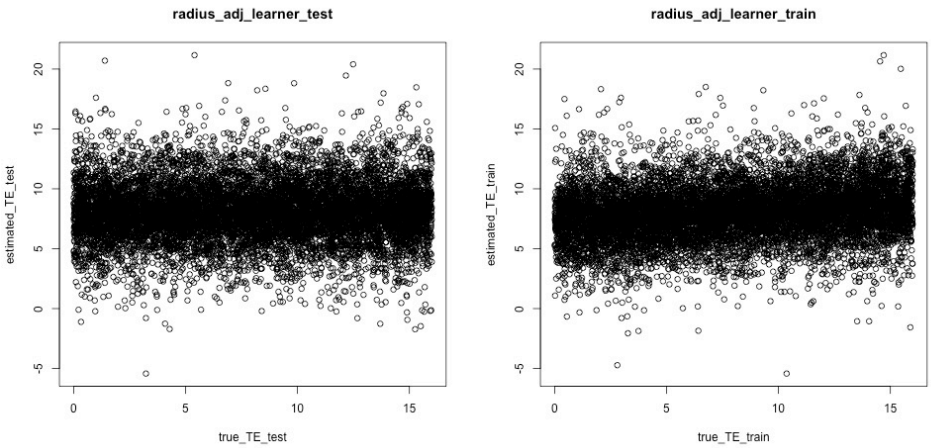

Figure 6: Adjusted regression with interaction term on Noiseless data with training sample size 10000

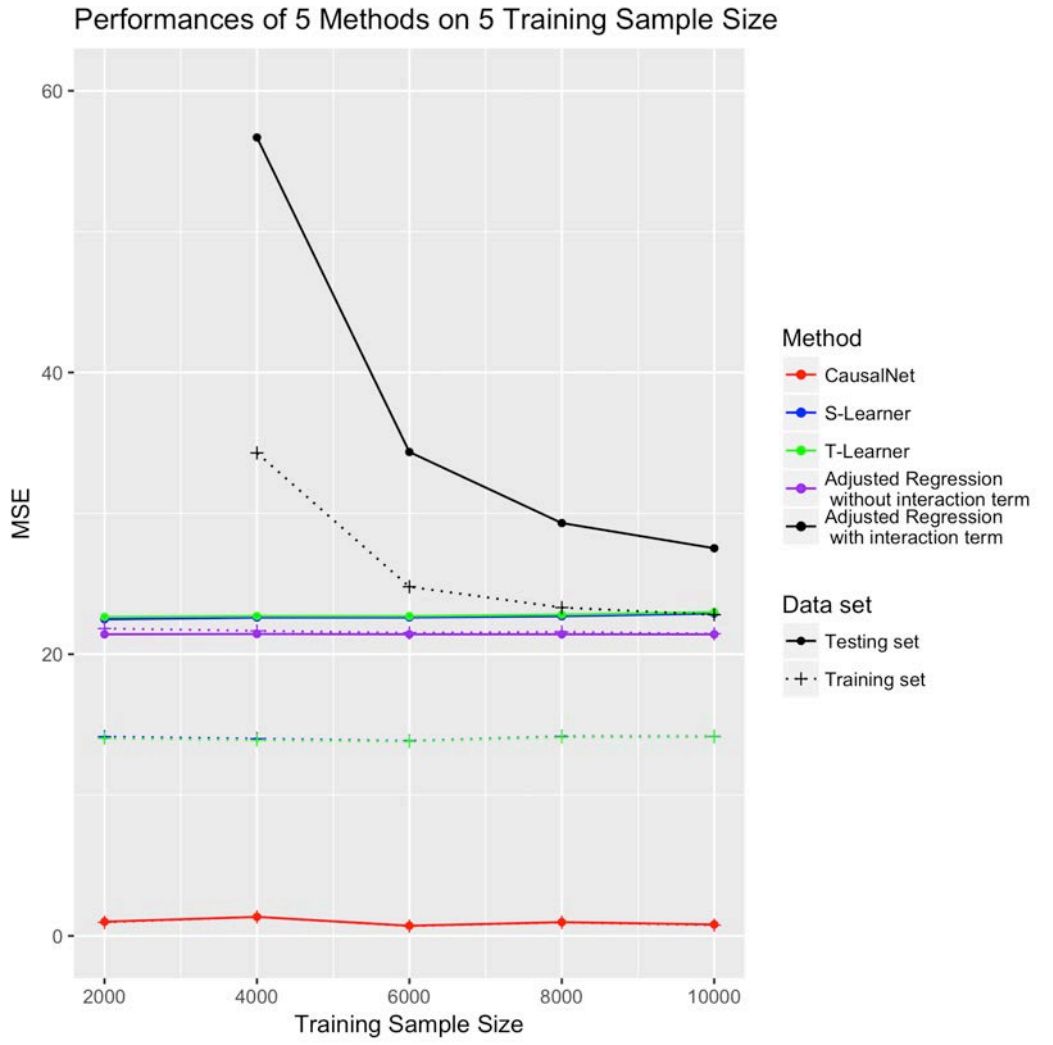

Figure 7: Performance of 5 Methods on Noiseless Data Set with 5 Training Sample Size

- Radius $R \sim \text{Unif}(0, 16)$,
- Origin $O \sim \text{Unif}(0, 32) * (0, 32)$, and $O \perp\!\!\!\perp R$,
- For pixels inside the circle defined above, pixels $\sim N(180, 64)$, truncated to [0,255] and rounded,
- For pixels outside the circle defined above, pixels $\sim N(0, 64)$, truncated to [0,255] and rounded,
- Potential Outcomes $Y(0) \sim N(0, 0.4)$, $Y(1) \sim N(R, 0.4)$,
- $T \sim B(1, 0.5)$ independent of (X, Y(0),Y(1)).

Figure 8 is an illustration of noisy images (covariates). Since the CATE is hard to compute exact in this setting, we take R to be our substitute for treatment effect, and do not change the names in our figures.

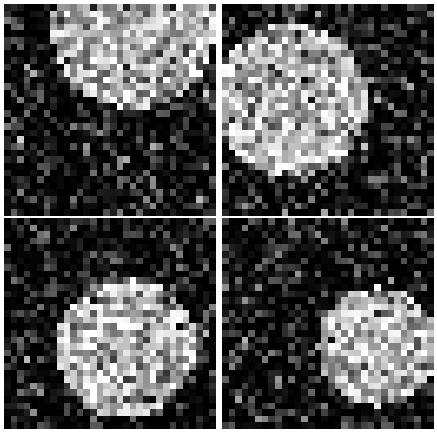

Figure 8: Noisy Images

Results

Figures 9 - 12 show the estimation versus the substitute of CATE (radius) of Causal Network, S-learner, T-learner, and adjusted regression with interaction term, for sample size 10000. Adjusted regression without interaction term estimates CATE as 8.032603. Figure 13 shows the performance of 5 methods on noisy data when sample size varies. The behavior on noisy data is similar to that of noiseless data.

3.2 Simple Relations

Since the linear regression based methods and tree based methods have been under intensive studied and has theoretical guarantees under simple model settings, we also want to test how the methods perform when the data generation procedure follows those simple models, or are likely to be preferable to linear regression base methods or tree based methods. We find four kinds of setting worth testing: data comes from a linear model, data comes from a polynomial model, data comes from two trees (one for treatment group and one for control group), data comes from two neural networks (one for treatment group and one for control group).

The covariates we use here are noisy images mentioned above, we generate outcomes according to different simple relations.

Simulation Setting

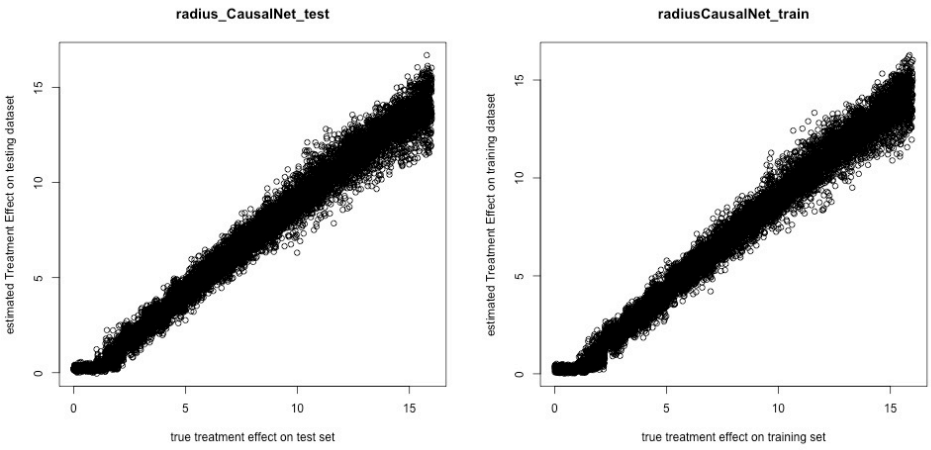

Figure 9: Causal Network on Noisy data with CATE substitution and training sample size 10000

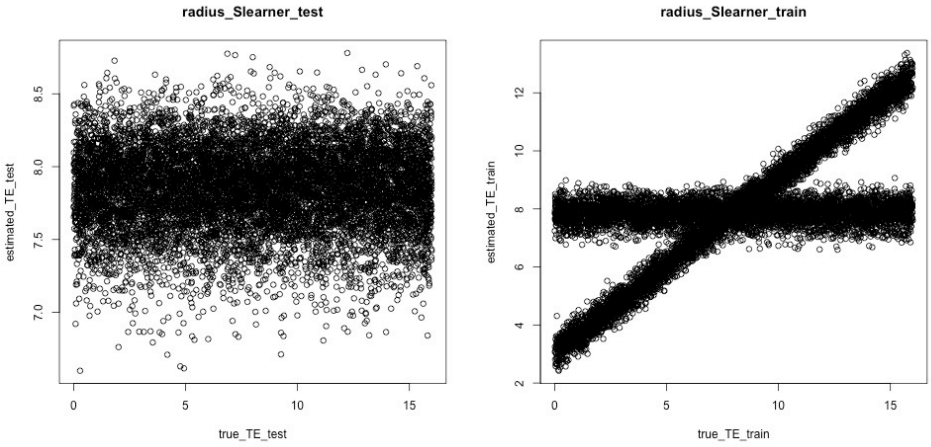

Figure 10: S-learner on Noisy data with CATE substitution and training sample size 10000

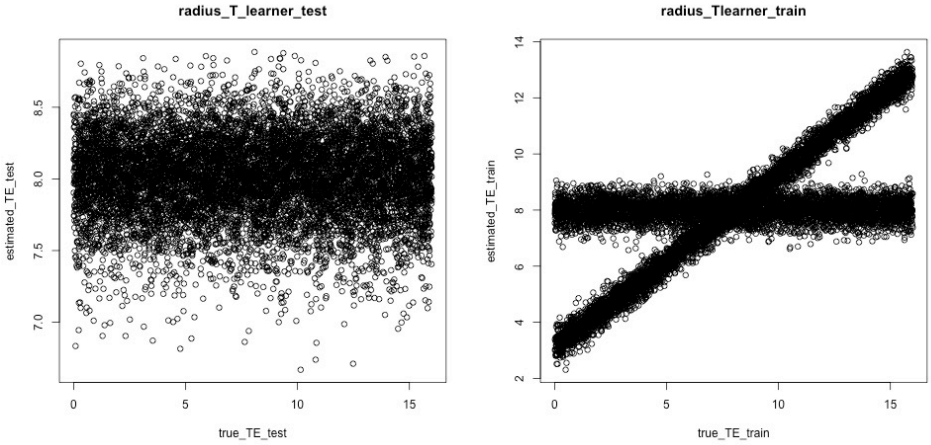

Figure 11: T-learner on Noisy data with CATE substitution and training sample size 10000

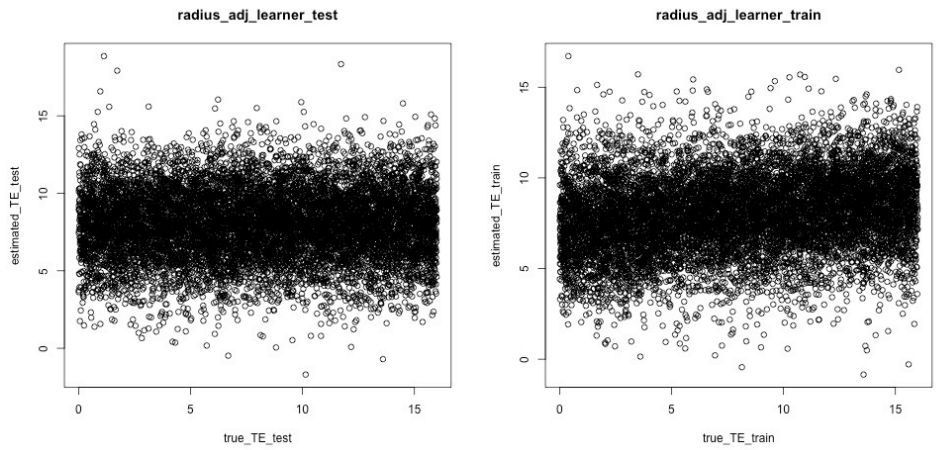

Figure 12: Adjusted regression with interaction term on Noisy Data with CATE substitution and training sample size 10000

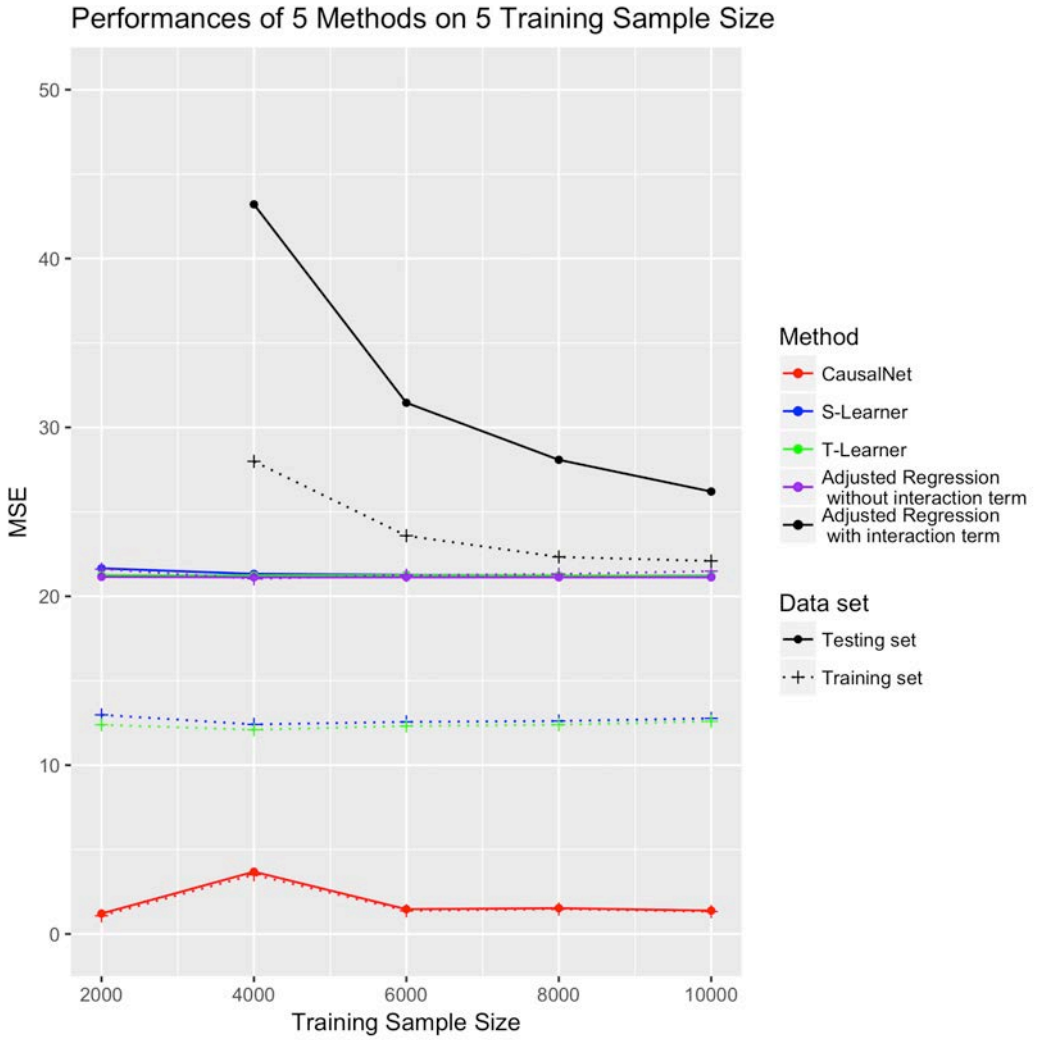

Figure 13: Performance of 5 Methods on Noisy Data with CATE substitution and 5 Training Sample Size

For all simple relations, data comes from two regression model and the experiment is randomized and balanced. More precisely, it follows the following model.

- $Y(0) = f_0(X) + \epsilon_0$,
- $Y(1) = f_1(X) + \epsilon_1$,
- $\epsilon_0 \sim N(0, \sigma^2)$, $\epsilon_1 \sim N(0, \sigma^2)$,
- ϵ_0, ϵ_1 and X are mutually independent,
- X obeys the same distribution as of noisy data in previous subsection,
- $T \sim B(1, 0.5)$, $T \perp\!\!\!\perp (X, Y(1), Y(0))$.

For linear data generator, $f_t(X) = X\beta_1 + tX\beta_2$, where (β_1, β_2) are generated from $N(0, 10)$, with seed 5, and only keep first 20 components of β_1 and β_2 to be nonzero. $\sigma = \sqrt{mean(Y_i^2)/10}$, where Y_i are data we generated with sample size 10000.

For polynomial data generator, $f_t(X) = X\beta_3 + X^T D_1 X + (X\beta_4 + X^T D_2 X)t$, where D_1 and D_2 are diagonal matrix. All components of $\beta_3, \beta_4, diag(D_1), diag(D_2)$ are generated independently from $N(5, 50)$. $\sigma = \sqrt{mean(Y_i^2)/10}$, where Y_i are the data we generated with sample size 10000.

For tree based data generator, f_1 and f_2 are random forests trained by mnist dataset, with responses being 1 to 9 and interchanging responses with same remainders when divided by 5 . $\sigma = 1.1$, which is approximately $\sqrt{mean(Y_i^2)/10}$, where Y_i are the data we generated with sample size 10000.

For neural network based data generator, f_1 and f_2 are pretrained VGG net and pretrained Alexnet respectively. $\sigma = \sqrt{mean(Y_i^2)/10}$, where Y_i are the data we generated with sample size 10000.

Results

For all the figures shown below, y axis is MSE divided by the variance of CATE of test dataset. Therefore, the method is only valid when it is less than 1, or it is meaningless to say the method could detect heterogeneity. When it is larger than 1, fine-grid comparison scale is not reasonable, going to scales as rough as the order it is of may be a better choice.

Linear

Figure 14 shows the performances of 5 methods, with different training sample size, on linearly generated data. Missing points stand for value being larger than the scope. In this setting, linear regression based methods could be theoretically proved to be optimal. However, tree based methods perform better here, which may due to the sparsity imposed during data generation. Causal Net do not perform well in this setting, but it is of the same order with the best one, while the best one is no better than always guessing the average.

Polynomial

Figure 15 shows the performances of 5 methods, with different training sample size, on polynomially generated data. Missing points stand for value being larger than the scope. All the methods are well above one, indicating bad performances. They are also of the same order except adjusted regression with interaction term.

Tree Based Data Generator

Figure 16 shows the performances of 5 methods, with different training sample size, on data generated by tree based data generator. Missing points stand for value being larger than the scope. This setting is expected to be favorable for trees, though tree based methods mysteriously have drastically higher training error than test error. Though both training

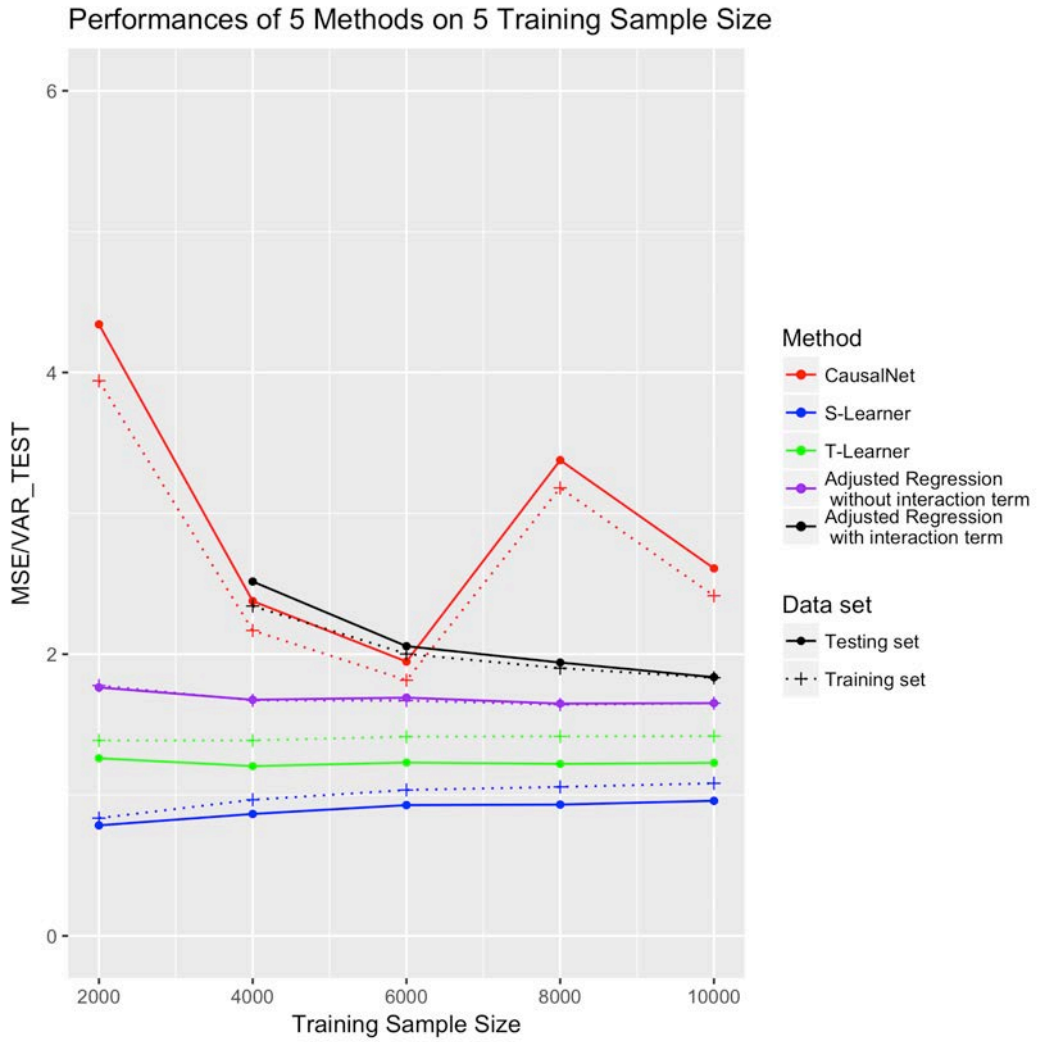

Figure 14: Performance of 5 Methods on Linear Data Generator with 5 Training Sample Size

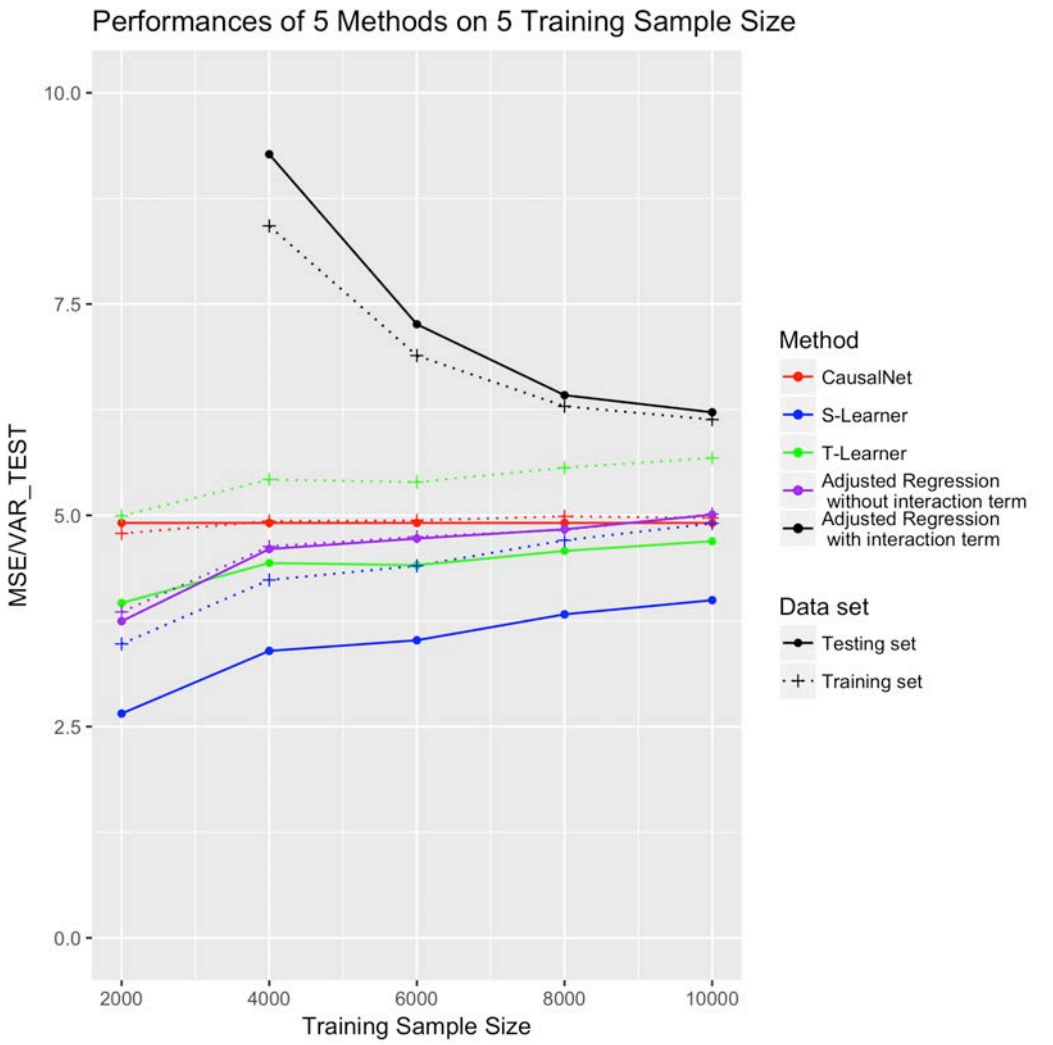

Figure 15: Performance of 5 Methods on Polynomial Data Generator with 5 Training Sample Size

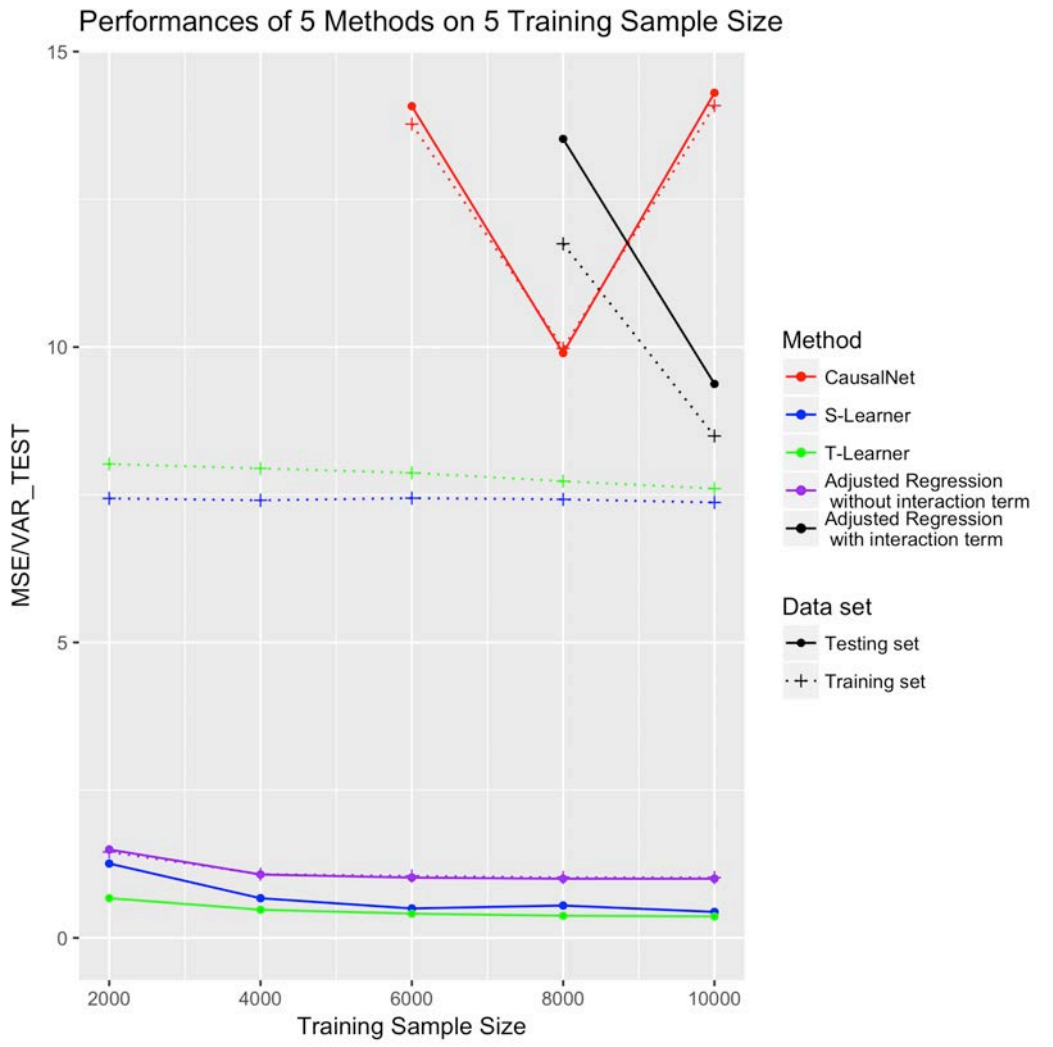

Figure 16: Performance of 5 Methods on Tree Based Data Generator with 5 Training Sample Size

error and test error of tree based methods are better than others, we can see that training error is of the same order of other methods and test error is also above 1. All the methods perform bad in this setting.

Neural Network Based Data Generator

Figure 16 shows the performances of 5 methods, with different training sample size, on data generated by neural network based data generator. The missing points stand for value being larger than the scope. This setting is expected to favor neural network, however, since both VGG network and Alexnet are much more complicated and involved, causal network does not include those models, despite of the fact that all of them are neural networks. In this setting, all the methods performs unsatisfactory— they are all above one, and of the same order, except the adjusted regression with interaction term.

4. Discussion

In this paper, we reinterpreted heterogeneous treatment effect estimation, discussed both the criterion for heterogeneous treatment effect estimation methods and the justifiability of

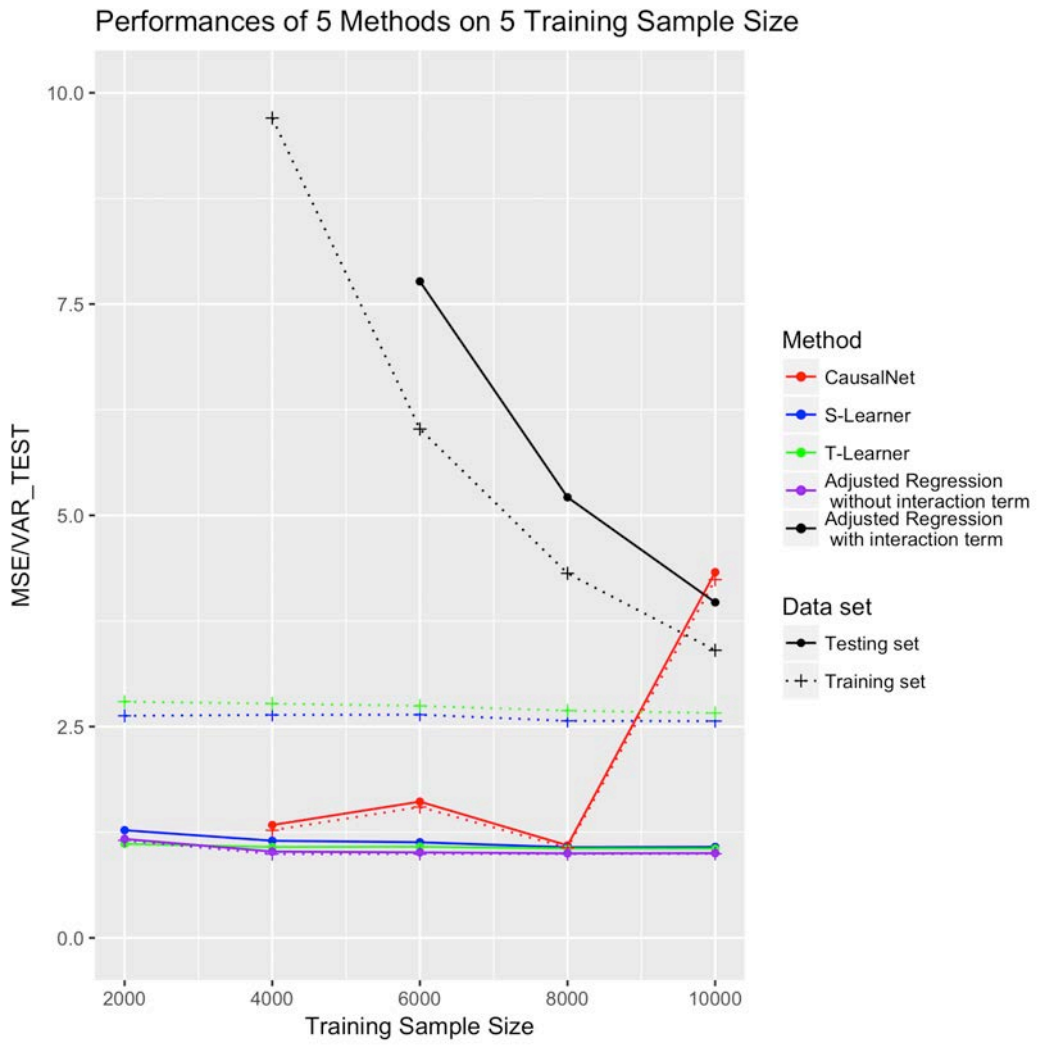

Figure 17: Performance of 5 Methods on Neural Network Data Generator with 5 Training Sample Size

using CATE to quantify heterogeneous treatment effect.

We analyzed both the advantages and issues needing consideration of integrating neural network into heterogeneous treatment effect estimation. It wins in its expressiveness, ability of dealing with data of various forms (e.g. structured data like image and text), ease of combining different networks into one, short computational time for moderately large network, computational support on hardware level, and the ability for smart use of both shared and separate information of control and treatment group (our diverter mechanism). It losses in issues of its optimization, the accuracy and convergence behavior (optimization wise convergence) of which is not yet well understood and involves tuning (optimization wise), despite of the empirical success and wide applicable range of default off the shelf parameter setting.

We proposed diverter mechanism to automatically enable both information share and separation in treatment and control group, and in our diverter mechanism, treatment indicator is not by nature restricted to 0 and 1, it can take continuous value, like dose. We give a specific configuration of causal network. We tested our network on simulated image data, both in setting where the image (covariates) does not have completely irrelevant noise, and in setting where the image (covariate) has completely irrelevant noise, thus we can also see where CATE stops to be a good quantification of heterogeneous treatment effect. In both settings, causal network performs much better than other methods. We tested causal net along with linear regression based methods and tree based methods on simulated data generated by linear data generator, polynomial data generator, tree based data generator and neural network based data generator, with the covariate being the images mentioned above, in order to see how causal network behaves in settings where linear regression based method, tree based methods, neural net based methods, S-learner scheme and T-learner scheme are expected to do better. But the settings themselves are hard problem due to high dimensionality of images, the nonstandard distribution of covariate and topological-free relationship between the responses and images. We found that all the methods behave badly on these settings, and approximately equally bad except adjusted regression with interaction term being extremely bad, especially when training sample size is small.

Therefore, causal networks shows huge advantage in image data setting, where the topological structure is related to heterogeneous treatment effect. And for hard situations, our specific causal network does not save the day in all the cases, and does not ruins the day either.

Our specific causal network configuration is very simple and is CNN based, but it already shows huge advantage in heterogeneous treatment effect estimation with informative image data. It is promising to combine the strength of other time proved network structures (e.g. RNN) and expressiveness of networks (incorporating time proved non neural network based methods or structures) to fully borrow the strength of neural networks into heterogeneous treatment effect estimation with various form of data. Given the flexibility of integrating treatment information into the neural network, exploration of continuous treatment indicator or indicator of different forms is promising. In the end, however, it is always important to balance the representation capacity, the network structure complexity and computational complexity, and this should always be kept in mind when designing networks.

5. Acknowledgment

Part of the work is part of the first author's undergrad thesis at Tsinghua University, the first author wants to thank Professor Michael Zhu, for his machine learning seminar held in Tsinghua University; Karl Kumbier and Professor Bin Yu, for introducing the first author

to state of art machine learning methods during her visit to Berkeley. The first author also wants to thank the stat department of Wharton School, for supporting her PhD study.

References

[1] Susan Athey and Guido W Imbens. Machine learning methods for estimating heterogeneous causal effects. *stat*, 1050(5), 2015.

[2] T. Tony Cai, Tengyuan Liang, and Alexander Rakhlin. Computational and statistical boundaries for submatrix localization in a large noisy matrix. *Ann. Statist.*, 45(4):1403–1430, 08 2017.

[3] David A Freedman. On regression adjustments to experimental data. *Advances in Applied Mathematics*, 40(2):180–193, 2008.

[4] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Meta-learners for estimating heterogeneous treatment effects using machine learning. *arXiv preprint arXiv:1706.03461*, 2017.

[5] Winston Lin. Agnostic notes on regression adjustments to experimental data: Reexamining freedmans critique. *Ann. Appl. Stat.*, 7(1):295–318, 03 2013.

[6] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

[7] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, (just-accepted), 2017.

6. Appendix

Following are some exploratory simulations to explore neural networks' characteristics on simplified toy networks.

6.1 Linear Neural Network on Linear Data generator

Network: One layer linear neural network with linear activation function

Data Generator: Linear Data Generator

Purpose: separate out computational issue

Result: figure 18 shows the computational issue affects accuracy, but when sample size gets larger and iteration number gets larger, it is alleviated.

6.2 Linear with Non Linear Activation Function

Network: One layer linear neural network with sigmoid activation function

Data Generator: Linear Data Generator

Two optimizing initializing points

Purpose: separate out issue of choosing initializing points

Result: figure 19 shows different initial points affects the accuracy in the first several iterations but in the long run, is not a big issue.

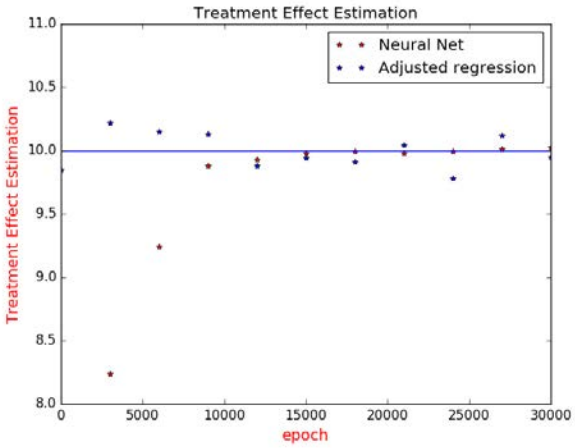

Figure 18: Computing accuracy

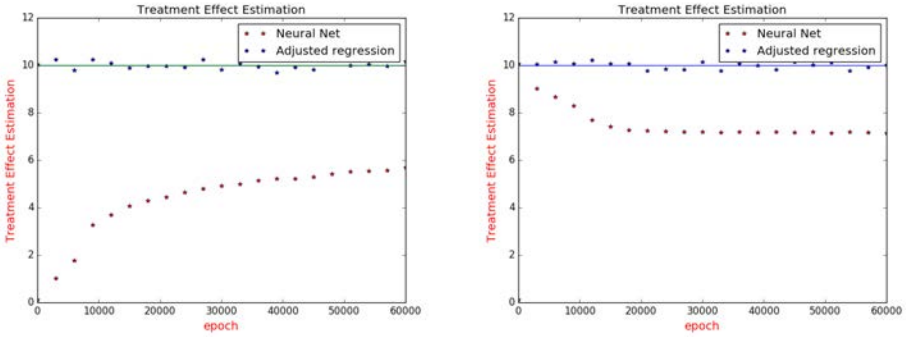

Figure 19: Influence of starting points

6.3 Ability of detecting nonlinearity

Network: Two layer nonlinear activation function

Data Generators: linear (figure 21), polynomial with small variance (figure 22) ,polynomial with large variance (figure 23)

Purpose: Explore the behavior of deeper neural network on linear data ; Explore the ability of detecting nonlinearity

Results: Two layer nonlinear activation function can detect nonlinearity when linearity increase, though perform a bit worse than linear regression based method in linear model, in which setting the linear regression based method is provably optimal.

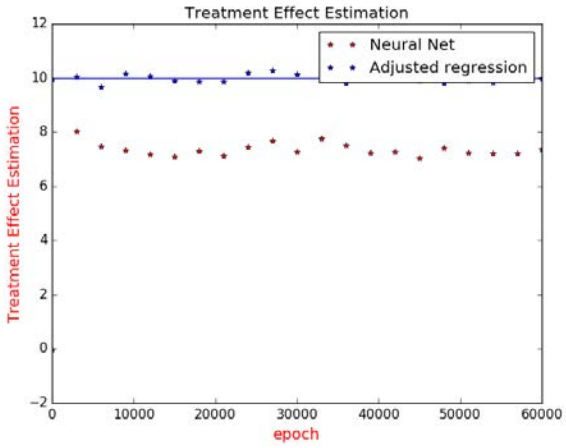

Figure 20: Two layer nonlinear NN with Linear Data generator

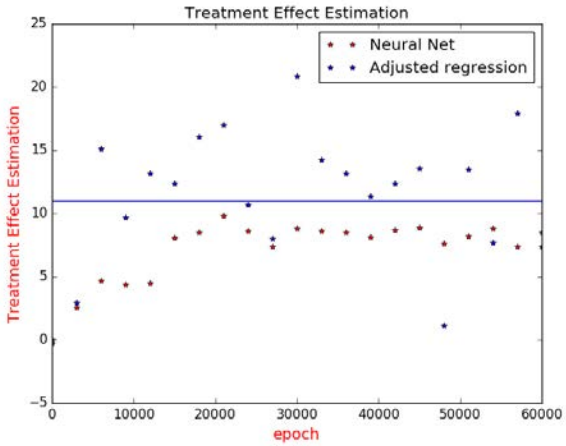

Figure 21: Two layer nonlinear NN with Nonlinear Data generator, Small Variance

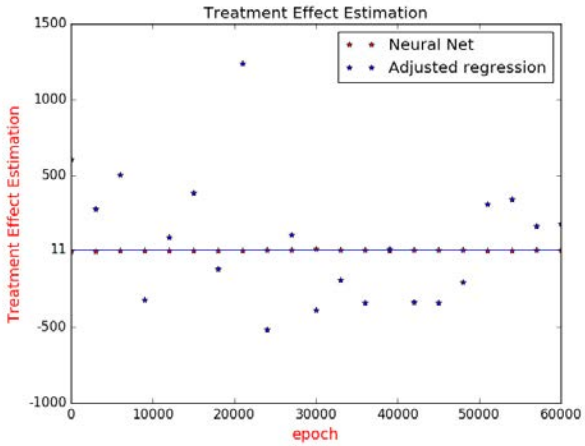

Figure 22: Two layer nonlinear NN with Nonlinear Data Generator, Large Variance