AMSI VACATION RESEARCH SCHOLARSHIPS 2019–20

EXPLORE THE MATHEMATICAL SCIENCES THIS SUMMER

Study Causal Inference Techniques for Data-Driven Personalised Decision-Making

Zhou Dai

Supervised by Professor Jiuyong Li

University of South Australia

Vacation Research Scholarships are funded jointly by the Department of Education and Training and the Australian Mathematical Sciences Institute.

1 Abstract

With the rapid accumulation of big data, data-driven personalised decision making is becoming a reality in various areas, such as personalised online recommendation, precision medicine targeting specific patient subgroups and personalised teaching and learning. In the area of causal inference, heterogeneous treatment effect estimation has been studied extensively, with the goal of identifying the different effects of a treatment on different subpopulations or individuals. Recently machine learning techniques have been introduced for heterogeneous treatment effect estimation to deal with large and observational data, e.g. gene expression for identifying patient subgroups characterised by their distinct genetic features which possibly have led to the heterogeneous effects of a cancer treatment in the different subgroups. The existing machine learning techniques, however, are facing two major challenges: how to accurately identify subgroups from observational data and how to efficiently deal with large scale and multiple sources of data. This project aims to develop new machine learning and causal inference techniques to tackle the challenges. The outcome of the project can be applied to various application areas, e.g. medicine, particularly cancer treatment, business intelligence, and government policy making.

2 Statement of authorship

Under the direction of my academic supervisor I studied the Rubin causal model as well as some naive machine learning algorithm based papers from Athey Susan, Fredrik D. Johansson and David Sontag. I outlined the workflow of the propensity score matching and demonstrated the basic model in both causal inference and deep neural network so those who are interested in applying machine learning with causal inference will get a general idea about the framework. I also proved that the standardized mean difference formulas for continuous variable and dichotomous variable are exactly the same. In addition I redefined some notations and formulas to make it consistent throughout the paper.

3 Introduction

In data science normally we start the analysis on the data by looking at the relationship between variables. The scatter plot is one of many powerful tools that can be used to reveal such correlations. If there is strong correlation between two variables then we can construct ordinary regression model to represent such linear relationship and estimation could be made in the range of the variables, as it shows in figure 1. Although noticing such existence of a strong correlation between these two variables

it doesn't mean we could expect one of the variable to be modified when we manipulate on the other, i.e. correlation differs from causation.

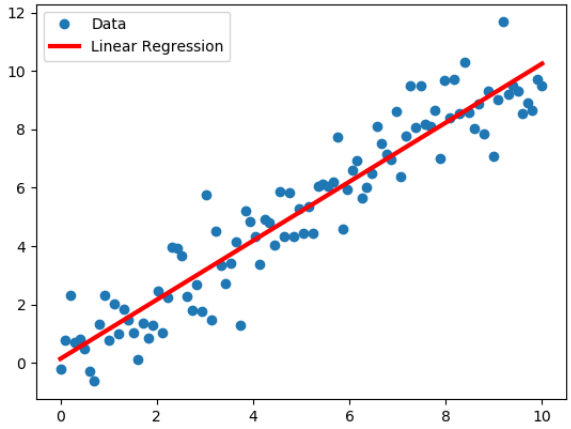

Figure 1: Correlation between two variables.

The classical application of causal inference is to estimate the overall treatment effect or the treatment effect on the population level, but if combined with advanced techniques in Machine Learning such as Deep Neural Network (DNN) better estimation could be achieved on the individual level, i.e. personalized decision making is possible. For instance, more than 50% of the breast cancer patients have received radiotherapy (RT) as treatment, although not all patients have benefited from the treatment as evidenced by distant metastatic spread and local recurrence. Prediction of individual responses will allow a stratified approach of applying the treatment, saving those unsuitable patients from the associated iatrogenesis[1].

4 Causal inference model

There are mainly two causal inference models, one is known as Pearl Graph Model or causal Bayesian networks or DAGs proposed by Judea Pearl in 2000. The other one is known as the Rubin causal model (RCM) or potential outcomes framework. The name "Rubin causal model" was first coined by Paul W. Holland. The potential outcomes framework was first proposed by Jerzy Neyman in his 1923 Master's thesis, though he discussed it only in the context of completely randomized experiments. Rubin extended it into a general framework for thinking about causation in both observational and experimental studies[3]. This paper focus on the Rubin causal model with DNN.

The settings of Rubin causal model is not complicated at all. Suppose we have several records of patients with two features or variables such as age and gender as it shows in figure 2. We use

(age, gender, treatment)	BP after medication
(40, F, 1)	$Y_1 = 140$
(40, M, 1)	$Y_1 = 145$
(65, F, 0)	$Y_0 = 170$
(65, M, 0)	$Y_0 = 175$
(70, F, 0)	$Y_0 = 165$

Figure 2: Patient records with features and outcomes.[5]

$x \in X$ to represent each patient with a set of features. There is also a special treatment variable which is denoted by T, in our case $T \in \{0, 1\}$ showing it is a binary variable. In other situations T could be multiple variable to represent $\{treatmentA, treatmentB, \cdots\}$. Finally there is an outcome variable denoted by Y. In our case we have two potential outcomes Y_0 and Y_1 according to the binary treatments respectively. After the treatment really happen one outcome becomes actual outcome and the other one becomes counterfactual outcome.

With these settings, we can estimate the individual treatment effect by $Y_1 - Y_0$ and the overall or average treatment effect by $\mathbb{E}[Y_1 - Y_0]$. But in practice it is only possible to have one of these potential outcomes observed, which is known as the fundamental problem in causal inference [4]. In other words, there are missing values in the outcome variable for each of the patient so it is not possible for us to calculate the treatment effect directly as it shows in figure 3.

Factual (observed) set		Counterfactual set	
(age, gender, treatment)	BP after medication	(age, gender, treatment)	BP after medication
(40, F, 1)	$Y_1 = 140$	(40, F, 0)	$Y_0 =?$
(40, M, 1)	$Y_1 = 145$	(40, M, 0)	$Y_0 =?$
(65, F, 0)	$Y_0 = 170$	(65, F, 1)	$Y_1 =?$
(65, M, 0)	$Y_0 = 175$	(65, M, 1)	$Y_1 =?$
(70, F, 0)	$Y_0 = 165$	(70, F, 1)	$Y_1 =?$

Figure 3: Fundamental problem of causal inference.[5]

An intuitive solution is to find a similar patient that share the same features, i.e. for a forty-year-old lady with treatment, we need another forty-year-old lady without treatment, then we can compare their outcomes to estimate the individual treatment effect. This process is known as matching and statistically we are trying to achieve two groups of samples that have similar distribution on some features x, as it shows in figure 4. Then the average treatment effect we are trying to measure can be

formulated as $\mathbb{E}[Y_1 - Y_0|X = x]$, i.e. the treatment effect conditioning on some features x.

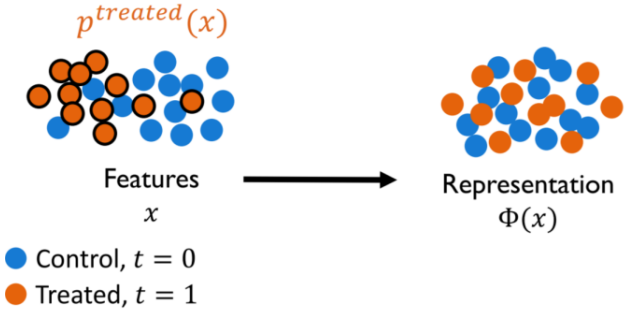

Figure 4: We achieve two balanced groups on some features through matching.[5]

It is worth noticing the difference between $\mathbb{E}(Y|T = 1)$ and $\mathbb{E}(Y_1)$ such that $\mathbb{E}(Y_1 - Y_0) \neq \mathbb{E}(Y|T = 1) - \mathbb{E}(Y|T = 0)$, in which $\mathbb{E}(Y|T = 1)$ can be recognized as the sub-population with treatment while $\mathbb{E}(Y_1)$ is the actual treatment outcomes on population level.

Before moving on to look at the matching methods there are several assumptions[4] of the Rubin causal model we need to consider:

- Stable unit treatment value assumptions (SUTVA), which states that units do not interfere with each other, the treatment assignment of one unit does not affect that outcome of another unit and the spillover or contagion are also terms for interference.
- Consistency: $Y = Y_T$ if $T = t$ for all t, in which t represents a specific treatment.
- Ignorability: $Y_0, Y_1 \perp\!\!\!\perp T|X$, which is stating among people with the same values of X, we can think of treatment T as being randomly assigned. For instance, suppose men is more prone to a treatment than women but for all men the possibility of getting such treatment is the same.
- Positivity: for every sef of values for X, treatment assignment was not deterministic: $P(T = t|X = x) > 0$ for all t and x, i.e. each treatment has at least one recipient.

Then with the assumptions we can calculate the conditioning treatment outcome

$$\begin{aligned}\mathbb{E}(Y|T = t, X = x) &= \mathbb{E}(Y^t|T = t, X = x) \text{ (consistency)} \\ &= \mathbb{E}(Y^t|X = x) \text{ (ignorability)}.\end{aligned}$$

And the overall treatment outcome

$$\mathbb{E}(Y^t) = \sum_x \mathbb{E}(Y|T = t, X = x)P(X = x).$$

5 Methods of matching

In tradition we have Randomized Controlled Trail (RCT) in which before the treatment we select random samples from the population and randomly (maybe by flipping coins) assign them to either treatment group or control group. By RCT treatment is exchangeable because all the samples in the two groups have balanced or similar distributions on all the relevant features due to the randomness. But in the observational data study in most cases samples have different distributions on some features. For instance men may be more prone to cancer so more prone to get treatment, i.e. gender is the confounder affecting both the treatment and the outcome.

One classical method to avoid this effect of confounding is called propensity score matching, which first generate a scaler variable known as the propensity score on different features containing the confounders and then match samples from treatment and control group based on the propensity score. The propensity score is defined as the probability to get treatment conditioning on some features

$$\pi := P(T = t|X = x).$$

The general process to run propensity score (PS) matching is as the following:

1. Run logistic regression on treatment group with $P(y|x)$.
2. Apply model to calculate PS for each sample in control group.
3. Check if covariates are balanced across treatment and control groups.
4. Match each sample in treatment group with the one in control group based on PS.
5. Verify that covariates are balanced across treatment and control group.
6. Improve the matching method and repeat matching if SMD is high (e.g. introducing caliper).
7. Estimate the conditioning treatment effect.

To access the balance before and after the matching between the treatment and control group we can apply the standardized mean difference formula[9], for continuous variable we have

$$\text{SMD} = \frac{\bar{x}_t - \bar{x}_c}{\sqrt{\frac{S_t^2 + S_c^2}{2}}},$$

and when the feature is binary or dichotomous variable we have

$$\text{SMD} = \frac{\hat{p}_t - \hat{p}_c}{\sqrt{\frac{\hat{p}_t(1-\hat{p}_t) + \hat{p}_c(1-\hat{p}_c)}{2}}},$$

where $\hat{p}_t, \hat{p}_c$ represent the proportion of samples showing '1' in the treatment and control group respectively. It is not hard to see

$$\bar{x}_t - \bar{x}_c = \hat{p}_t - \hat{p}_c$$

when x is dichotomous variable and

$$\begin{aligned} S_t^2 + S_c^2 &= \mathbb{E}[x_t^2] - (\mathbb{E}[x_t])^2 + \mathbb{E}[x_c^2] - (\mathbb{E}[x_c])^2 \\ &= \mathbb{E}[x_t] - (\mathbb{E}[x_t])^2 + \mathbb{E}[x_c] - (\mathbb{E}[x_c])^2 \quad \text{when } x \text{ is dichotomous} \\ &= \hat{p}_t - \hat{p}_t^2 + \hat{p}_c - \hat{p}_c^2 \\ &= \hat{p}_t(1-\hat{p}_t) + \hat{p}_c(1-\hat{p}_c), \end{aligned}$$

which means these two formulas are exactly the same. In regular if the calculated SMD is larger than 0.1 then we say the distribution between the treatment and control group are considered significantly different and vice versa.

There are also other methods based on propensity score[9], the whole list is shown as the following:

- Propensity Score Matching
- Stratification on the PS
- Inverse Probability of Treatment Weighting Using PS (IPTW)
- Covariate Adjustment Using the Propensity Score

In the first step of generating the propensity score for each sample based on combination of diverse features, instead of logistic regression there are also studies that apply machine learning methods such as random forest[7], Bayesian Additive Regression Tree(BART) or even Deep Neural Network[5]. The rest section of this paper will focus on how we apply an adapted version of DNN with the Rubin causal model. We start by looking at what is Artificial Neural Network (ANN).

6 Artificial Neural Network

A simplest neural network containing only one neuron is also known as **perceptron**. As it shows in figure 5, we have several features $x \in X$ as the input. Then the linear combination of all the features can be denoted by Σ with $w \in W$ as the coefficients. Finally the linear combination Σ is fed into an activate function to get an output.

There are many options for the activation function[10], in tradition we use the sigmoid function since the output could be always between 0 and 1 as it shows in figure 6. Thus the composite function is

$$\frac{1}{1+e^{-X^TW}},$$

where X^TW represents the linear combination of features X with coefficients W. By applying this simple model, we can obtain a probability like number (which is between 0 and 1) for each sample

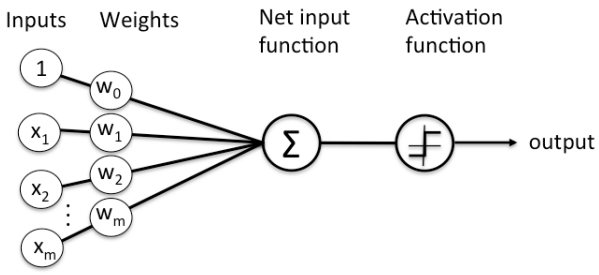

Figure 5: A simple neural network with only one neuron.

with several features, which can be recognized as the possibility for each sample to be within a specific class y, i.e. $P(y|X = x)$. If we decide the threshold to be 0.5 then this simple model can be used as a binary classifier.

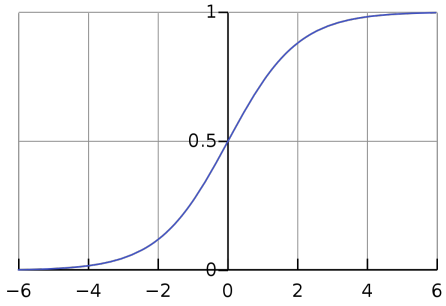

Figure 6: Sigmoid function as activation function.[10]

To deal with more complicated task such as multi-classification we need more complicated neural network which can be obtained by stacking many simple perception together, as it shows in figure 7. We still have three layers with the input layer at the bottom and output layer at the top. The middle layer is known as the hidden layer, if there are more than two hidden layers then the whole neural network is known as the Deep Neural Network (DNN).

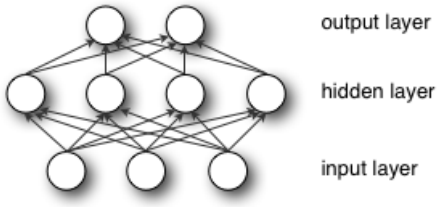

Figure 7: More complicated neural network.

For each layer we have

$$a^i = \sigma(\mathbf{w}^i \cdot a^{i-1} + b^i),$$

where the superscript i indicates which layer, a^i is the output of layer i, σ is the chosen activation function and b is the bias we add for each layer, i.e. each layer's input is the output of the bottom layer (if we put input layer at bottom as figure 7 shows). Also $a^0 = X$ which is simply the vector of input features.

To figure out what is the best values for the coefficients W and bias b we need a cost function. The traditional cost function which is popular from least square fitting is

$$C = \frac{1}{2}\sum(\hat{y} - y)^2,$$

by which we are trying to minimize the difference or error between $\hat{y}$ the estimation from the model and y the known label from the data. But this cost function doesn't ensure the existence of the optimal solution when we apply algorithm like gradient descent to solve for W and b, so the logistic regression cost function is introduced instead

$$\begin{aligned} C &= -\frac{1}{m}\sum\left[y\log\hat{y} + (1-y)\log(1-\hat{y})\right] \\ &= \frac{1}{m}\sum L(\hat{y}, y), \end{aligned}$$

in which this term

$$L(\hat{y}, y) = -[y\log\hat{y} + (1-y)\log(1-\hat{y})]$$

is known as the loss function for each sample and m is the number of samples. This cost function works because when $y = 1$ the loss function becomes $-\log\hat{y}$ and we try to minimize it i.e. to maximize $\hat{y}$. If we use sigmoid function as the activation function then

$$\hat{y} = \frac{1}{1 + e^{-(w^T x + b)}}$$

which always ranges in $(0, 1)$, so $\hat{y} = 1$ which is what we expected. Similarly when $y = 0$ we have $\hat{y} = 0$. By the logistic regression cost function it is guaranteed that when we try to minimize it the problem would be convex i.e. the global minimum coincides with the local minimum, so it is guaranteed we could reach the optimal point.

We could derive the cost function by maximum likelihood method. For each sample what we need is when $y = 1$ our model gives $\hat{y} = P(y|x)$ while when $y = 0$ we have $1 - \hat{y} = P(y|x)$, which could be combined in one equation

$$P(y|x) = \hat{y}^y(1-\hat{y})^{(1-y)}.$$

Taking log on both side gives

$$\begin{aligned}\log P(y|x) &= y \log \hat{y} + (1-y)\log(1-\hat{y}) \\ &= -L(\hat{y}, y).\end{aligned}$$

When there are m samples we have

$$\begin{aligned}\log P(y|x_1 x_2 \cdots x_m) &= \log \prod_{i=1}^{m} P(y^i|x^i) \\ &= \sum_{i=1}^{m} \log P(y^i|x^i) \\ &= -\sum L(\hat{y}, y).\end{aligned}$$

To maximize on it is the same to minimize on

$$\frac{1}{m}\sum L(\hat{y}, y)$$

which is exactly the defined cost function.

One classical algorithm to minimize on the cost function is gradient descent. Suppose $\hat{y} = \frac{1}{1+e^{-z}}$ where $z = w_1 x_1 + w_2 x_2 + b$ and the loss function is $L = -[y \log \hat{y} + (1-y)\log(1-\hat{y})]$. Then for one sample case we start by a random point (w_0, b_0) and update (w, b) by

$$\begin{aligned}w_1 &= w_1 - \alpha \frac{\partial L}{\partial w_1}, \\ w_2 &= w_2 - \alpha \frac{\partial L}{\partial w_2}, \\ b &= b - \alpha \frac{\partial L}{\partial b},\end{aligned}$$

where

$$\begin{aligned}\frac{\partial L}{\partial w_1} &= x_1(\hat{y} - y), \\ \frac{\partial L}{\partial w_2} &= x_2(\hat{y} - y), \\ \frac{\partial L}{\partial b} &= \hat{y} - y,\end{aligned}$$

and $\alpha \in \mathbb{R}$ is the learning step. Notice that if the learning step is too large then we might skip the optimal point and never reach it, while if the learning step is too small then it would take long to reach the optimal point.

7 DNN for causal inference [5]

To construct the Deep Neural Network for causal inference we need to consider two things:

1. The model should learn two balanced groups that have similar distributions on all covariates.
2. The error between the model estimation and the factual outcome from the data should be minimized.

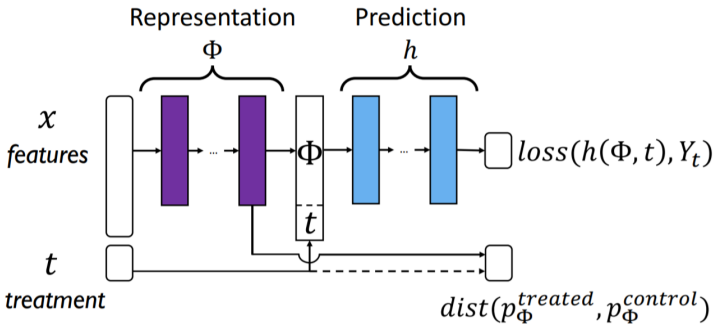

Figure 8: Causal inference with Deep Neural Network.[5]

Figure 8 shows the workflow of the adapted DNN for causal inference. We first feed samples with features $x \in X$ into the first few hidden layers in purple which will do representation learning by minimizing the difference of the distributions between the treatment and control groups. Then the outputted representation combined with the treatment variable t would be fed into the posterior part of hidden layers in blue to construct the model by minimizing the error between the estimation of the model and the factual outcome.

Same as the general Deep Neural Network model we need a cost function

$$C = \frac{1}{n}\sum_{i=1}^{n} \left| h(\Phi(x_i), t_i) - y_i^F \right| \tag{1}$$

$$+ \frac{\gamma}{n}\sum_{i=1}^{n} \left| h(\Phi(x_i)), 1 - t_i) - y_{j(i)}^F \right| \tag{2}$$

$$+ \alpha \text{disc}_{\mathcal{H}}(\hat{P}_\Phi^F, \hat{P}_\Phi^{CF}), \tag{3}$$

where $h \in \mathcal{H}$ is our hypothesised function, Φ is the representation function that gives balanced distribution on the input features $x \in X$ between treatment and control group, y_i^F is the factual outcomes of samples under treatment t_i, $y_{j(i)}^F$ is the factual outcomes of samples under $(1 - t_i)$ and also closest to samples under treatment $(1 - t_i)$ but without observed outcomes, $\text{disc}_{\mathcal{H}}$ is a discrepancy function to measure the distance between two distributions and finally $\hat{P}_\Phi^F, \hat{P}_\Phi^{CF}$ represents the distribution of the factual outcomes and the counterfactual outcomes on representation Φ respectively.

The first term simply try to minimize the error between the estimation of the model and the factual outcomes from the data for samples with treatment t_i. The second term evaluates similar error but for the samples with treatment $1 - t_i$. The last term $\text{disc}_{\mathcal{H}}$ is the only term contains unobserved or counterfactual information.

Let us look at the last term in more details. The discrepancy function $\text{disc}_{\mathcal{H}}$ is to measure the difference between two distributions which are denoted by $\hat{P}_{\Phi}^{F}, \hat{P}_{\Phi}^{CF}$ respectively. This is exactly the concept of domain adaptation[11]. Let χ denote the feature or input space and $\mathcal{Y}$ be the label or output space. We have a source domain denoted by (Q, f_Q) where Q is the distribution over χ, f_Q is the corresponding labelling function. Similarly we have a target domain denoted by P, f_P where P is the distribution over χ and f_P be the corresponding labelling function, i.e. the source domain and the target domains share the same feature space and label space but with different distribution on χ. Normally f_Q can be learnt if $\mathcal{Y}$ is known in the source domain, just as we know $\hat{P}_{\Phi}^{F}$.

We denote a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ which could be the squared error widely used in regression. For any two functions $h, h' : \chi \to \mathcal{Y}$ and any distribution D over χ then we have

$$\mathcal{L}_D(h, h') = \mathbb{E}_{x \sim D}[L(h(x), h'(x))]$$

as the expected loss of $h(x)$ and $h'(x)$. The domain adaptation problem is regarding selecting a hypothesis function $h \in \mathcal{H}$ by minimizing this expected loss according to the target distribution P.

Fix $h \in H$ the difference of expected losses can be denoted as

$$\mid \mathcal{L}_P(f_P, h) - \mathcal{L}_Q(f_P, h) \mid .$$

To measure the difference between two distributions we will take supremum of this quantity. Then we can define the discrepancy function: given a hypothesis set $\mathcal{H}$ and loss function L the discrepancy $disc$ between two distributions P and Q over χ is defined by

$$disc(P, Q) = \max_{h, h' \in \mathcal{H}} \left| \mathcal{L}_P(h', h) - \mathcal{L}_Q(h', h) \right| .$$

This is not in a metric space since we have $disc(P, Q) = 0$ for $P \neq Q$, but it does satisfy the triangle inequality for any loss function L and is symmetric as well. It is also better than the general

$L_1(P,Q)$ distance since

$$\begin{aligned} disc(P,Q) &= \max_{h,h'\in\mathcal{H}} \left| \int_{\chi} (p(x)-q(x))L(h'(x),h(x))dx \right| \\ &\leq \max_{h,h'\in\mathcal{H}} \int_{\chi} \left| (p(x)-q(x))L(h'(x),h(x)) \right| dx \\ &\leq M \int_{\chi} |p(x)-q(x)dx| \\ &= ML_1(P,Q), \end{aligned}$$

when the loss function L is bounded by M and $P,\ Q$ are continuous with density functions $p,\ q$.

8 Conclusion

We have looked at one of the causal inference model known as the Rubin Causal Model or the Potential Outcome Framework, which indicates that to estimate the treatment effect on individual level we need both the outcome with treatment and outcome without treatment when all the variables except the treatment variable keep unchanged. The fundamental problem in causal inference is that only one of the potential outcomes could be observed, i.e. we only have the labels for the factual outcome but not the counterfactual one. In tradition even groups that have same distributions on features can be obtained automatically through Randomized Control Trial. For observational data, several methods have been studied on the same purpose. One classical method is Propensity Score Matching which map the features from $\mathbb{R}^n$ space to a scaler number so samples from treatment group could be matched with samples from control group on the closest propensity score. There are also several advanced methods based on propensity score such as Stratification, Inverse Probability of Treatment Weighting and Covariate Adjustment. The second half of this paper focuses on the application of the Deep Neural Network to causal inference. We start by introduction on perceptron the simplest one-neuron network and move on to more complicated multi-hidden-layer neuron network or Deep Neuron Network. We define cost function by minimizing which to derive the coefficients $\mathbf{w}$ and $\mathbf{b}$. The adapted version of DNN are used to realize causal inference on the individual level. The special cost function consists of three parts: the error on the treatment outcomes, the error on the outcomes without treatment and the distance between two distributions.

9 Acknowledgements

I wish to acknowledge Professor Jiuyong Li for his valuable advice and AMSI for funding this project.

References

[1] Weijia Zhang, Thuc Duy Le, Lin Liu, Zhi-Hua Zhou, Jiuyong Li, Mining, *heterogeneous causal effects for personalized cancer treatment*, Bioinformatics, Volume 33, Issue 15, 01 August 2017, Pages 2372–2378, https://doi.org/10.1093/bioinformatics/btx174

[2] Judea Pearl, *Graphical Models for Probabilistic and Causal Reasoning*, University of California, Los Angeles, 2014

[3] Wikipedia contributors, *Rubin causal model.* Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 11 Feb. 2020. Web. 20 Feb. 2020.

[4] Guido W. Imbens, Donald B. Rubin, *CAUSAL INFERENCE for Statistics, Social, and Biomedical Sciences An Introduction*, Cambridge University Press, 2014

[5] Fredrik D. Johansson,Uri Shalit, David Sontag, *Learning representations for counterfactual inference*, NIPS 2016 Deep Learning Symposium, December 2016

[6] Athey S, Imbens G. *Recursive partitioning for heterogeneous causal effects.* Proc Natl Acad Sci U S A. 2016;113(27):7353–7360. doi:10.1073/pnas.1510489113

[7] Stefan Wager & Susan Athey (2018), *Estimation and Inference of Heterogeneous Treatment Effects using Random Forests*, Journal of the American Statistical Association, 113:523, 1228-1242, DOI: 10.1080/01621459.2017.1319839

[8] Wikipedia contributors. *Propensity score matching.* Wikipedia, The Free Encyclopedia, 13 Feb. 2020. Web. 23 Feb. 2020.

[9] Austin PC. *An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies.* Multivariate Behav Res. 2011;46(3):399–424. doi 10.1080/00273171.2011.568786

[10] Wikipedia contributors. *Activation function.* Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 21 Feb. 2020. Web. 23 Feb. 2020.

[11] Corinna Cortes and Mehryar Mohri, *Domain Adaptation and Sample Bias Correction Theory and Algorithm for Regression* Theoretical Computer Science, Volume 519, 2014, Pages 103-126, ISSN 0304-3975.