Machine Learning for causal Inference on Observational Data

Author: Hernán E. BORRÉ

A thesis submitted for the degree of Master of Science in Artificial Intelligence

Supervisor: Dr. Spyros Samothrakis
School of Computer Science and Electronic Engineering
University of Essex

August 2018

Declaration of Authorship

I, Hernán E. BORRÉ, declare that this thesis titled, "Machine Learning for causal Inference on Observational Data" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:
Hernán Emilio Borré _____

Date:
28/08/2018 _____

"Thirty years ago, we used to ask: Can a computer simulate all processes of logic? The answer was yes, but the question was surely wrong. We should have asked: Can logic simulate all sequences of cause and effect? And the answer would have been no. "

Gregory Bateson, Mind and Nature

UNIVERSITY OF ESSEX

Abstract

Faculty Name

School of Computer Science and Electronic Engineering

Master of Science in Artificial Intelligence

Machine Learning for causal Inference on Observational Data

by Hernán E. BORRÉ

The established scientific way to make claims about cause and effect is to perform a Randomized Controlled Trial (RCT). However, although RCTs are the best way to determine causal effects, the chances to perform such rigorous scientific experiments is, most often, either impossible or unethical. The Average Treatment Effect (ATE) is usually the outcome of the RCT experiments and this outcome is ideally proof of an effect under the studied population, which hopefully extends to other individuals. In contrast, it is most common to find Observational Data, in which the data that has been collected might be heavily unbalanced for treatment assignments, or the patients covariates might come from completely different distributions. Nevertheless, the ultimate goal of causal effects is to find the specific Individual Treatment Effect (ITE) for each patient. Identifying the Individual Treatment Effect is a topic that has always been important in the field of causality, especially within the machine learning community.

Applications of such predictions are related with medicine, but can be extensively used in financial investments, advertisement placements, recommender systems for retail and social sciences, and beyond.

The ability to learn complex non-linear relationships of some machine learning algorithms have been trying to detect and predict policies, in which given the particular features of an individual (patient) the algorithms could determine whether or not to apply the treatment to them.

In this thesis, the ITE will be predicted using a benchmark semi synthetic-dataset which has been unbalanced. Assuming *strong ignorability*, alternative machine learning techniques that had not been tested in past publications will be applied to predict the ITE from observational data. The results obtained are compared with state-of-the-art outcomes; some of the algorithms applied in this work performed similarly to more complex, custom designed methods.

In addition, a full review of all recent literature in the machine learning applied to causal inference has been done.

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Spyros Samothrakis. He has been an essential pillar in the whole process of this dissertation and my career; giving me not only academic professional support but also encouraging me to make the most out of this process.

Second, I would like to thanks my parents, who had always been listening to my issues, stress periods and emotional crisis, and cheered me up every time I needed them. Without them, I would not be able to be here writing this dissertation by any chance. They made me believe that everything is possible in life if you try hard enough and you are a honest person. Infinitely grateful to them, forever.

Third, to my former university Professors from Universidad Tecnológica Nacional, Factual Regional Buenos Aires, Dr. Oscar Bruno, Dr. Alejandro Prince and Dra. María Florencia Pollo-Cataneo, for their recommendation letters, support through the year and enlightenment in my professional career giving me always the best advice I could always get.

Fourth, I would like to thanks Dr. Uri Shalit, who offered me immediate help and advise on this dissertation's topic and who also helped me on the full IHDP dataset collection metrics for benchmark comparisons.

Fifth, to all my classmates (some of them friends now) who spent with me countless hours discussing about our passion, making the world a better place through Machine Learning and Artificial Intelligence.

Last but not least, I would like to thanks the Government of Argentina and the Argentinian Ministry of Education for giving me the chance of coming to study to one of the best universities in the world throughout the *BEC.AR* scholarship.

Contents

List of Tables

List of Abbreviations

ML	**M**achine **L**earning
SVR	**S**upport **V**ector **R**egressor
RL	**R**einforcement **L**earning
NN	**N**eural **N**etworks
LR	**L**inear **R**egression
KNN	**K** **N**earest **N**eighbours
RCE	**R**andomized **C**ontrolled **E**xperiment
ITE	**I**ndividual **T**reatment **E**ffect
ATE	**A**varage **T**reatment **E**ffect
PEHE	**P**recision in **E**stimation of **H**eterogenous **E**ffects
CATE	**C**onditional **A**verage **T**reatment **E**ffect
RCM	**R**ubin **C**asual **M**odel

Chapter 1

Introduction

1.1 Motivation

Causality is often confused with correlation. Correlation does not imply causation. These inferences are often called *"spurious correlations"* and they often confuses the inference process in which humans make decisions.

A common definition of Causality is still not agreed by the scientific community nowadays.

The proven scientific way to make claims about cause and effect it is to perform what is called a Randomized Controlled Trial(RCT). In a Randomized Controlled Trial, a statistically representative portion of the population that will be participating of the experiment (trial), are exposed to a treatment(action), which could be either positive - apply the treatment - or neutral (control) -giving the patient a placebo or not treating the patient(*unit*) at all.

All these concepts are related to medical words since the field in which RCTs are applied the most, is medical trials. However, it is not the only industry in which these concept of dragging conclusions from a trial can be done. For example, it is widely used in social studies, but can also be applied to make decisions on buying, selling or holding a particular stock, or displaying an advertisement that generate more sales than the others in the advertisements industry.

Nevertheless, the Randomized Controlled Trials are the best way to detect causal effects, the possibility of perform such scientific rigorous experiments is, most of the times, either impossible or unethical. An example could be seem when trying to detect if driving while being under the effects of alcohol can affect (or not) the driver's skills. Another clear example of this is determining the causes of smoking in teenagers or young people in which, to perform a RCT would involve to take two groups of non-smoking teenagers, make half of the *units* smoke for several years - it depends on the experiment or the research question - and then determine if smoking in young people would cause something or not after that period. As the reader can infer, there are clearly ethical problems associated with performing full RCTs to determine causes and effects.

It is also important to let clear that, there are cases in which it would be impossible to analyze previously collected data without having in mind the RCT method. These cases are known as ***observational data***. Observational data is defined as information that can be obtained from previously collected situations in which a formal

randomized trial method has not been applied but it is still important to try to determine causes and effects from that data. This is the case in most organizations in the present since they would possibly be collecting massive amounts of data during the last decades but they could not or would not be able to establish a RCT process during collection of the metrics. Moreover, sometimes the data collection process happens in a non randomized controlled experiment. For example, a not so common disease that just affects a small percentage of the population might happen to appear within a wide range of people that makes the inference process difficult.

In the past, causal Inference methods have been a Statisticians only field. However, with recent advances of machine learning algorithms, more computer scientists and machine learning engineers have been trying to infer causes and relationships through traditional and new machine learning techniques.

The ability to learn complex non-linear relationships of some machine learning algorithms have been trying to detect and predict policies, in which given the particular features of an individual (patient) the algorithm could determine if to apply or not the treatment (action) to them. This concept is also known as Individual Treatment effect estimation or it could also be referred as Policy risk when predicting a binary class which is is to apply or not the treatment through the discovery of a certain threshold.

It is a matter of interest to be able to predict the individual ("customized") treatment effect because this would lead to better decisions(actions or treatments), specifically shaped for each person and not only relaying on the average of the whole studied population.

The ultimately motivation of this work is to be able to predict the Individual Treatment Effects for new patients with the previously collected data using alternative machine learning techniques to the ones used in past research efforts. Moreover, the compiling of a literature that can be understood by more computer scientist will be tried to present as well as running code of all the experiments performed will be released for others to build upon it.

1.2 Purpose and Research Question

The purpose of this dissertation is to predict the Individual Treatment Effect (ITE), the Average Treatment Effect (ATE) and the Precision in Estimation of Heterogeneous Effect (PEHE), for a widely adopted benchmark dataset in the field usually refer to this as IHDP (Infant Health Developed Program), which is a semi-synthetic dataset particularly unbalanced and created for the task of causal inference on observational data(Hill, 2011).

The research experiments are about trying alternative machine learning methods without adding extra complexity, custom error loss or custom metric functions while learning and predicting, would be able to obtain similar or even better results than the state-of-the-art metrics based on the exact same benchmark dataset.

1.3 Approach and Methodology

For causality analysis purposes, the Rubin Potential Outcomes Framework (*Causal Inference Using Potential Outcomes: Design, Modeling, Decisions*) and its notation will be used during this whole thesis. This model is also known as the Newman-Rubin causal model and it is an approach to statistical analysis of cause and effect based on the potential outcomes framework.

For the machine learning experiments, the latest version of the scikit-learn(*User guide: contents — scikit-learn 0.19.2 documentation*) python framework had been used. All their underlaying methods and default hyper-parameters had been used. Also, the mathematical notation of their documentation will be presented to describe each algorithm's functions and limitations.

The different algorithms had been tested with a benchmark *standard dataset* for causal inference from observational data, Infant Health and Development Program (IHDP), introduced by (Hill, 2011), as a semi synthetic-dataset, based on real features obtained from a real an observational study (Gross, 1993). Replications on this dataset had been created to get *10, 100 and 1,000 cases* to be able to train and predict the machine learning models on them and get the desired metric error results afterwards.

It is important to notice that, the testing method and the metrics used to determine the effectiveness of each algorithm, are different from the ones normally used to test machine learning algorithms and testing these algorithms within a causal framework differs substantially from the usual train/test paradigm of the machine learning field.

Since a synthetic-dataset has been used, the real Individual Treatment Effect is available to perform testing metrics. Therefore, the experiment results will consist of the performance of each algorithm based on Individual Treatment Effect, Average Treatment Effect (ATE) and Precision in Estimation of Heterogeneous Effect (PEHE). These three are the metrics displayed in the Experiments chapter for each algorithm trained. A detailed explanation of these formulas will be found in the next sections 2.0.3.

Also, it is important to notice that, the machine learning algorithms are trained just using the treatment applied (observed), the features (covariates in causal inference literature), and the observed outcome (usually known as Y or "Y factual"). After training, an completely unseen dataset during is used for **testing** purposes.

The already trained algorithm predicts just the Y *factual* based on the *unit* (also known as patient), features (covariates) for the both the cases the *unit* would have taken the treatment and likewise predicts the outcomes as if the *unit* would have taken the control treatment. Once both outcomes are predicted *("Y factual" and "Y counterfactual"*, the ATE, ITE and PEHE metrics are calculated. In addition, an average score and its deviation for each run of the *10, 100 and 1,000* replications of the IHDP had been run to evaluate these errors in bigger simulated scenarios.

The mathematical notation will be kept as minimum as possible to not confuse the reader with unnecessary information.

1.4 Scope and Limitation

In this document, the outcomes of the applied treatment to a patient will be only analyzed with respect two possible actions (binary treatment). Multi-valued treatments are not going to be covered in the experiments nor in the developed code but they can be easily extended to cover these cases.

A binary treatment is applied but its outcome value is **continuous**. Different to the most common used case of four possibles scenarios in which usually just two can be observed or measured. All the experiments and the code developed can be applied to discrete outputs but other machine learning techniques could be more suitable for this type of predictions (classification algorithms). Also, there are cases in which the task is to predict weather to apply or not the treatment to an *individual* (also known as *unit* or *patient*). To predict in this cases turns out more into a classification task in which a threshold on the interval confidence of predicting to affirmatively apply the treatment is usually set and validated through trial and error against several continuous values to determine what would be the one that predicts with best accuracy. This is call as Policy Risk on causal inference literature.

This case is more similar to real world scenarios where the data was observed and finally a decision on applying or not the treatment (*action*) has to be made in order to peruse a desired result.

In this work, the cases in which the dataset contains outcomes in a binary form to predict weather or not to apply the treatment will not be covered.

Chapter 2

Background

2.0.1 Rubin-Newman Causal Model

The *Rubin causal model* (RCM) (Rubin, 2005), also known as Rubin-Newman Potential Outcomes framework, is a extended statistical analysis frame to model *observational data* that Donald Rubin developed. He came up with the mentioned framework building it on top of the original Newman method that he developed in his 1923 masters thesis, extending it to non randomized controlled trials (observational data).

The Rubin-Newman potential outcomes framework consists in:

$$x_i \in \mathcal{X}$$

with an effectively applied treatment

$$t_i \in \{0, 1\}$$

The two possible potential outcomes are defined by

$$Y_0(x_i), Y_1(x_i) \in \mathcal{Y}$$

Of one of them (the one which actually happened), we can observe its *factual* outcome

$$y_i{}^F = t_i Y_1(x_i) + (1 - t_i) Y_0(x_i)$$

And let $(x_1, t_1, y_1{}^F), \ldots, (x_n, t_n, y_n{}^F)$ be a *unit* from the factual distribution.

Consequently, let $(x_1, 1 - t_1, y_1{}^{CF}), \ldots, (x_n, 1 - t_n, y_n{}^{CF}$ be the counterfactual sample.

Notice that all the factual outcomes y^F are known, whereas is never the case in any *unit* for the counterfactual outcomes y^{CF} (except for testing phase and just when the dataset is semi-synthetic or synthetic).

It will be used as interchangeable terms the expressions y^F or y_ft referring to factual observed outcomes, while y^{FC} or y_cft will be pointing to counterfactual outcomes.

2.0.2 The fundamental problem of causal analysis

The *fundamental problem of causal analysis* states that it is impossible, given a *unit* x and assigning either the treatment $t = 1$ or $t = 0$ to that *unit*, to observe the counter factual outcome $\mathbb{E}[Y_0|x, t = 1]$ *or* $\mathbb{E}[Y_1|x, t = 0]$ (what would have happen or what would have been the outcome if the other treatment would have been given to the *unit* x).

However, it is always possible to observe the outcome of the effectively applied treatment t, which is represented as $\mathbb{E}[Y_0|x, t = 0]$ *or* $\mathbb{E}[Y_1|x, t = 1]$ or in shorter terms, Y_0 or Y_1.

In this dissertation, the focus is on the case when the causal graph is simple and known to be of the form (Y_1, Y_0) <- x -> t , with no hidden confounders.

This problem can be extended, as most of the problems and applications discussed under the Rubin-Newman Potential Outcomes Frameworks in this dissertation, as a multi-treatment experiment. It is important to notice that the problem of not having access to the result of the counter factual outcome Y_cf is even worst when extending this problem to multi-treatment experiments since the missing values that matters for better Individual Treatment Effects are increased in the total order of possible treatments, except the one applied.

2.0.3 Metrics for Causality

Three well-known metrics in the causality field are reported for each implemented machine learning technique applied.

The losses that will be reported are:

- ε_{ITE} : Error of Individual Treatment Effect - also known as the Conditional Average Treatment Effect (CATE) - and it is how well or bad perform the treatment on one particular *patient*

$$ITE(x) := \mathbb{E}[\mathbf{y}|\mathbf{X} = x, \mathbf{t} = 1] - E[\mathbf{y}|\mathbf{X} = x, \mathbf{t} = 0] = \mathbb{E}[Y_{x1} - Y_{x0}]$$

- ε_{ATE} : Error of Average Treatment Effect, as it name describes it, it represent the effect that the applied *treatment*, either $t = 0$ or $t = 1$ depending on the whole population effectively had. Note bold that, as an average, it can be not the best solution to treat a new *patient* with this treatment since its unique characteristics as a *unit* might make them experience wrong results or no results at all.

$$ATE := \mathbb{E}[ITE(x)] = \mathbb{E}[\delta] = \mathbb{E}[Y_1 - Y_0], \forall x \in \mathbf{X}$$

- $\varepsilon_{\text{PEHE}}$: Precision in Estimation of Heterogeneous Effect is used to measure the precision trade-off between the Individual Treatment Effect and the Average Treatment Effect. It is important to notice that this metric *relates the ATE and ITE predictions*, penalizing the predictions that had been predicted right for one measure but wrong or not that accurate for the other one.

$$PEHE(x) := \frac{1}{N}\sum_{i=1}^{N}((y_{i1} - y_{i0}) - (\hat{y}_{i1} - \hat{y}_{i0}))^2$$

2.0.4 Assumptions

To work on the results, three important assumptions under the Rubin-Newman causal Framework shall be made:

- **Consistency**: For each *unit*, just one of the two potential outcomes can be observed. Hence, if $t = 0$, then $y = Y_0$ will be the observed outcome or factual (y^F). However, if the applied treatment was $t = 1$, afterwards, $y = Y_1$ will be the available observed outcome or factual y^F.
- **Strong Ignorability**: Also known as *no unmeasured confounders*, this assumption can be stated by $(Y_1, Y_0) \perp\!\!\!\perp t|x$, and $0 < p(t = 1|x) < 1 \forall x$. It is important to notice that to be able to state this assumption, a domain knowledge expert would have to assess the dataset and therefore, determine if there are no unmeasured confounders. That is the case for the dataset implemented in this work.
- **Common Support**: This assumption states that for each *unit* $x \in \mathcal{X}$, there is a positive probability of being both treated ($t = 1$) and untreated ($t = 0$):

$$0 < P(t = 1|x) < 1$$

2.0.5 Definitions

In causal inference from observational data, several terms are used interchangeably and might confuse the reader.

This subsection *should be clear* before going further into this dissertation.

Some common synonyms are:

- **unit**: is the subject of the analysis, the one that will be applied the treatment.
  patient, individual, input, x_i, $x_i \in \mathcal{X}$
- **covariates**: all the collected (observed) variable that have a direct effect on the outcome.
  features(ML), x, $x \in \mathcal{X}$
- **treatment**: the possible different actions that can be applied to a *unit*. Usually binary, but can be multi-valued under the Rubin-Newman Potential Outcomes Framework.
  action, t, $t \in \{0, 1\}$, $t \in \{0, \ldots, N\}$
- **Outcome**: the measured result of applying a treatment t to a *unit* x
  observed outcome, result, factual, Y factual, $y_f = y^F$
- **Counterfactual**: what would have been the result if the opposite treatment to the effectively applied would have been applied to a *unit*
  unobserved outcome, y_cft, y_cf, Y^{CF}

2.0.6 Related Work

Potential outcomes are the framework to mathematically describe causality and counterfactuals (Rubin, 1978).

Causality from observational data can be clearly applicable to a wide range of industries, e.g. advertisement placement selection, health care systems, finance or even to improve education (*Recursive Partitioning for Heterogeneous Causal Effects **; Hoiles and Van Der Schaar, 2016; Bottou et al., 2013). In particular, counterfactual inference in observational studies has been a topic of interested study in economics, statistics, health care, pharmaceutical companies, epidemiology and sociology (*Causal Inference Using Potential Outcomes: Design, Modeling, Decisions*; Morgan and Winship, 2014; *Causal Inference Using Potential Outcomes: Design, Modeling, Decisions*; Chernozhukov et al., 2016), whereas in machine learning the attention has been caught not less than a decade ago (Lang, 1995; Bottou et al., 2013; Swaminathan and Joachims, 2015a). A lot of work in machine learning had been targeted for discovering the underlying causal graph from collected data (*Nonlinear causal discovery with additive noise models*; Maathuis et al., 2010; Triantafillou and Tsamardinos, 2015; Mooij et al., 2016).

causal inference for counterfactual predictions is usually grouped by: parametric, non-parametric and doubly robust methods.

For parametric methods causal inference the relationships within features and actions pairs and rewards by implementing one or more parameters, trying to specifically model the relations within context, outcomes and actions (treatments). In these methods, linear and logistic regression (Prentice, 1976; Gelman and Hill, 2007), random forests (Wager and Athey, 2015) and regression trees (Chipman, George, and McCulloch, 2010) had been used in the past to complete the task. For example, (Wager and Athey, 2017) estimates ITEs by causal Forests, but their asymptotic estimates in datasets with a large number of relevant features has limitations that needs to be addressed in future work.

In non-parametric approaches the counterfactual predictions are mostly calculated through a propensity score matching and re-weighting (Joachims and Swaminathan, 2016; Austin, 2011; Rosenbaum and Rubin, 1983; Rosenbaum, 2002). To perform doubly robust causality is done by merging parametric and non-parametric methods (Dudik, Langford, and Li, 2011; Jiang and Li, 2015).

Double robust methods, are known for merging the characteristics of both methods. A common example of this application would be propensity score weighted regression (Bang and Robins, 2005; Dudik, Langford, and Li, 2011). When the treatment assignment probability is known, this method models the problem particularly well, e.g. in off-policy evaluation or learning from bandits. However, in most of the cases in observational data, their efficiency drops dramatically (Kang and Schafer, 2007).

Machine learning for predicting Individual Treatment effects has been arisen a lot of interest during the last two years, through the development of custom metric functions -as long as the application of other techniques on causality- with special focus on unbalanced treatment application datasets. This refers to the sub area of causality, which is known as causal inference from *observational data*. Observational data is data that has been or is collected without the possibility of design and run a proper Randomized Controlled Trial. The creation of custom distance learning metrics and custom loss functions applied to Neural Networks had brought interesting

advances to the scientific community (Shalit, Johansson, and Sontag, 2017; "Learning Representations for Counterfactual Inference"). (Tian et al., 2014) modeled interactions between the treatment and the inputs (covariates), creating a relatively balanced method. Specifically, for estimation of Individual Treatment Effect, (Johansson, Shalit, and Sontag, 2016a; Shalit, Johansson, and Sontag, 2017; Alaa, Weisz, and Van Der Schaar, 2017) had made important contributions, whereas for Policy Optimization (Swaminathan and Joachims, 2015a; Swaminathan and Joachims, 2015b) can be consulted for their work. In Policy Optimization, the goal is to find a policy (threshold) that maximizes the factual outcome, or in other words, that takes the risk of predicting the action to the minimum.

Adopting machine learning methods to estimate the individual treatment effect had gained increasing interest in the past years, just to name a few (Wager and Athey, 2015), (Athey and Imbens, 2016), ("Learning Representations for Counterfactual Inference"), (Shalit and Sontag, 2016), (Shalit, Johansson, and Sontag, 2017). (Johansson, Shalit, and Sontag, 2016a) and (Shalit, Johansson, and Sontag, 2017) worked on learning balanced representations while using Neural Networks for both learn better predictions on the factual outcome and minimize the error loss between the factual and counterfactual representation of the unbalanced observational data. Specifically, in (Shalit, Johansson, and Sontag, 2017), the authors built their work based on (Johansson, Shalit, and Sontag, 2016a), focusing on the counterfactual error term, deriving a family of algorithms and metrics in the form of Integral Probability Metrics. In ITE prediction, other work was performed by implementing Gaussian processes (Alaa, Weisz, and Van Der Schaar, 2017) and decision trees in different approaches (Hill, 2011; *Recursive Partitioning for Heterogeneous Causal Effects* *; Wager and Athey, 2015).

Similarly, (Atan et al., 2016) faces the problem of learning from biased data and several features by performing feature selection while predicting among multiple possible actions (outcomes), being this more challenging but modeling closer to actual industry problems. The authors also remarks the difficulty of learning the relevant features leading to predict some actions while not taking them into account for others. The relevant feature selection learning was done by implementing a way of Online Contextual Multi-Armed Bandit (CMAB) likewise from (Tekin and Van Der Schaar, 2018), with some limitations due to the nature of the observational data. Also, (Joachims and Swaminathan, 2016) used IPS estimates and empirical Bernstein inequalities to learn counterfactual outcomes, although they do not worked with observational data and they do not identify individual important features to perform the task.

In terms of Policy Optimization methods, (Swaminathan and Joachims, 2015a) came up with a Counterfactual Risk Minimization (CRM) method in which they look to minimize the Inverse Propensity Score of the *units* by introducing an algorithm named 'POEM'. After that, (Atan, Zame, and Van Der Schaar, 2018) propose to address the selection bias by learning representations, working closely related to filed to domain adaptation bounds in (Ben-David et al., 2007; Blitzer, McDonald, and Pereira, 2006). Additional techniques on policy optimization were done by (Beygelzimer and Langford, 2008) in which the propensity scores need to be known, solving the selection bias through rejection measurements. The algorithm that (Atan, Zame, and Van Der Schaar, 2018) introduces is based on domain adaptation (DA) as in (Gan et al., 2016). More work in the DA techniques field was done by (Zhang et al., 2013; Daumé, 2009).

To conclude, in cause and effect analysis, Time Series data, is widely adopted for decision making support. The main challenge in the continuous time space is to properly gather feedback from the outcomes to help determine a future decision (treatment). (Robins, 1986) was the first to learn and optimize decisions throughout time accounting for the possible actions. Through the integration of action-value functions (Nahum-Shani et al., 2012), algorithms can learn rules to make decisions along time-steps. An estimator on structural nested models, was introduced by (Lok, 2008). Furthermore, (*Causal Reasoning from Longitudinal Data*) used Bayesian posterior predictive distributions to solve this time series causality task. Later on time, (*Reliable Decision Support using Counterfactual Models*) introduced the 'Counterfactual Gassian Process' to predict the counterfactual future progression of continuous-time trajectories under sequences of future actions, implementing a Reinforcement Learning approach (Sutton and Barto, 2017) with off-policy learning due to the nature of the observational data. Retrospective observational data is used for off-policy learning to estimate the best expected reward of a policy that is set before (Dudik, Langford, and Li, 2011; Swaminathan and Joachims, 2015a; Jiang and Li, 2015; Păduraru et al., 2012; Doroudi, Thomas, and Brunskill, 2017).

2.1 Machine Learning

In this section, the applied machine learning techniques using *scikit-learn* open source framework to perform the experiments will be described.

The vast majority of the actual available methods tested belong to **Generalized Linear Models** and they can be represented as a target or label value as a linear combination of the covariates (inputs).

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \ldots + w_p x_p \tag{2.1}$$

where the vector $w = (w_1, \ldots, w_p)$ represents the *coefficients* and w_0 is the *intercept*.

2.1.1 Ordinary Least Squares (Linear Regression)

In this model, the objective is to minimize the residual sum of squares between the observed dataset, and the predictions made on it.

Mathematically, it solves the problem of:

$$\min_w ||Xw - y||_2^2$$

The main limitation of this method is that if the features (covariates) have an approximate linear dependence, the model produces a high variance and therefore, it is more sensitive to random errors in the prediction. This limitation affects specially to data collected with out a design that was previously shaped in a experimental way.

2.1.2 Ridge Regression

Ridge regression accounts some of the limitations of the above mentioned Linear Regression method by penalizing the coefficient's size. There can be noticed the loss turns into a problem of minimizing the sum of the squares penalized:

$$\min_{w} ||Xw - y||_2{}^2 + \alpha||w||_2{}^2$$

It is worth mentioning, that the parameter $\alpha \geq 0$ is the one that takes into account the amount of robustness to collinearity that the trained model is going to have.

2.1.3 Support Vector Regressor

A Support Vector Regressor (SVR) method is an extension of the widely spread Support Vector Machines for classification in order to solve regression problems. During the training phase, the best possible solution is the one that gets less penalized in total by a loss function. The vectors will be the inputs that are either misclassified, classified within enough margin or the ones on the edged of the hyper-plane generated that splits the dataset for future predictions.

In particular, a SVR takes the training vectors $x_i \in \mathbb{R}^p$, $i = 1, \ldots, n$, and a vector $y \in \mathbb{R}^n \varepsilon$–SVR solves the following primal problem:

$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\zeta_i + \zeta_i^*)$$

where e is the vector of all ones, $C > 0$ is the upper bound, Q is an n by n positive semidefinite matrix, $Q_{ij} \equiv K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is the kernel. Here training vectors are implicitly mapped into a higher (maybe infinite) dimensional space by the function ϕ.

The decision function is:

$$\sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_i, x) + \rho$$

2.1.4 Bayesian Ridge

Bayesian Ridge Regression holds its robustness for ill-posed problems compared to Linear Regression.

This technique elaborates a probabilistic model formulated by a regression problem with parameter w of the general Bayesian Regression solver as a spherical Gaussian:

$$p(w|\lambda) = \mathcal{N}(w|0, \lambda^{-1} I_p)$$

The scikit-learn defaults are being used to train the model: $\alpha_1 = \alpha_2 = \lambda_1 = \lambda_2 = 10^{-6}$

In the fitting of the model process, the parameters $w, \alpha and \lambda$ are the one to be estimated together.

2.1.5 Lasso

Lasso Regression is a linear model but fitted with ℓ_1 prior as regularizer. Its objective is to minimize:

$$\min_w \frac{1}{2n_{samples}}||Xw - y||_2^2 + \alpha||w||_1$$

This method solves the *min* of the least-squares penalty with $\alpha||w||_1$ added, where α is a constant and $||w||_1$ is the ℓ_1-norm of the parameter vector.

It is important to notice that this algorithm retrieves sparse models, which may be helpful to perform feature selection.

2.1.6 Lasso Lars

This model is trained with the Least Angle Regression (Lars). The L1 regularization is applied.

The objective function is determined by:

$$\frac{1}{2n_{samples}}||y - Xw||_2^2 + \alpha||w||_1$$

2.1.7 ARD Regression

Although this method is similar to Bayesian Ridge, it may lead to sparser weights w.

It drops the assumption of Gaussian being spherical, making it to elliptical.

Mathematically:

$$p(w|\lambda) = \mathcal{N}(w|0, A^{-1}),$$

with diag $(A) = \lambda = \{\lambda_1, ..., \lambda_p\}$.

2.1.8 Passive Aggressive Regressor

Suitable for large scale learning, they do not require a learning rate but it requires a regularization parameter c.

It can be used with two different loss functions. PA-I or *epsilon intensive* or PA-II, also known as *squared epsilon intensive*.

2.1.9 Theil Sen Regressor

It is specially suited for multi-variate outliers, but its efficiency decreases dramatically when it tries solve a high-dimensionality problem. When this happens, this method becomes similar to a Linear Regression with Ordinary Least Squares in high dimension.

2.1.10 K-Neighbors Regressor

In this algorithm, the target is predicted by n nearest neighbors used during the training phase. It is important to notice that n is defined by the user and it will affect positively or negatively the obtained results of the predictions.

2.1.11 Logistic Regression

Logistic Regression is mostly used for classification problems but it can be used for more than one class predictions using the *log* function.

This *scikit-learn* implementation can fit binary, One-vs-Rest, or multinomial logistic regression with optional L2 or L1 regularization.

Several solvers and regularizations were applied to the datasets and the results will be discussed in the Experiments section.

Chapter 3

Methodology

In this chapter, what was tried to be achieved will be detailed. In addition, the methods that were used are going to be explained, as well as any other necessary information related to help the reader to understand the flow of the later covered experiments.

In addition, how the dataset that had being used will be displayed, closing with a section about other possible datasets that can be applied and possible limitations to the ones used in this dissertation.

Finally, a whole coverage of the Dataset used to perform the experiments will be detailed.

3.1 Dataset

Datasets for testing causal inference on observational data extracted from real life scenarios are difficult to obtain.

On the one hand, the whole point of the current project - and at some extent - of the last efforts in machine learning applied to causality are to make mostly accurate predictions on a set of *units*, *patients* or *inputs* (in the Machine Learning vocabulary) that had been collected without the chance of previously design a carefully planned Randomized Controlled Trial. Since the nature of the already collected or *observational data* has not been randomized properly, neither it comes from the same probability distribution. Also, the amount of *units* which received the treatment versus the amount of them who did not receive the treatment could potentially differ substantially.

On the other hand, some experiments can not be designed and executed under Randomized Controlled Trial conditions since they are unethical or impossible to perform. For example, designing a experiment to test if driving while under the effects of alcohol is dangerous for the driver or the pedestrians tested against a control treatment which, in this case, will be driving without alcohol consumption is completely unethical to perform for clear reasons.

To solve these limitation when working on causal effects on observational data, synthetic, semi-synthetic or toy datasets are created by the researchers in order to establish a good starting point and benchmark framework to try, test or develop better algorithms that are able to make more accurate predictions surpassing the state-of-the-art results.

Lastly, it is important to notice that there are two different kinds of predictions for causal inference. One is the most common one to obtain, in which, the counter factual outcomes could not be recorded because of the nature of the experiment (and being this the *fundamental problem of causal analysis*). In this cases, a Policy risk function π is designed to apply or not the treatment t depending on a certain threshold θ. The less possible errors when predicting the application of the active treatment or control, are the main goal when iterating over different values of the threshold variable for the dataset trained.

3.2 IHDP dataset

The Infant Health and Development Programa (IHDP) (Gross, 1993) was a Randomized Controlled Trial (RCT) hold in the United States across multiple sites applying control and treatment to reduce the developmental and health problems of low birth weight of premature infants. On the one hand, the treated group received visits to their homes, integration at a dedicated child development center, in addition to a pediatric follow-up, which can be described as *high-quality child care*. On the other hand, the control group only received the pediatric follow-up.

However, (Hill, 2011) presented a semi-synthetic (also could be mention in this work and in the field as semi-simulated) dataset that derived directly from the original IHDP RCT (Gross, 1993) mentioned in the above paragraph. In (Hill, 2011) some continuous and binary covariates from the this real life RCT were selected. Making use of these covariates, (Hill, 2011) created a simulated outcome and generates non-parametric simulated outcomes for the whole population of the trial. In the dataset, 25 covariates of the whole study where taken for this dataset creation. Consequently, the author introduces an artificial imbalance on the control and treatment *individuals* by **removing a subset of the treated population**. Finally, the dataset comprises of 747 subjects (*units* or *inputs*) from which **608** had not been applied the treatment (control) and **139** treated. As it can be clearly noticed, the dataset end up being quite unbalanced, especially for learning and predicting effects of the treatment $\mathbf{t = 0} or \mathbf{t = 1}$based on the generalization task that an algorithm can perform.

Along with the covariates for each *unit*, it can be observed the simulated causal information. This is the effectively applied treatment ($t = 0 or t = 1$, the observed outcome ($Y_f t$), the counter-factual outcome ($Y_c f t$) and the average outcomes with noise $mu0$ and $mu1$

In this dissertation, 100 and 1000 replications of the original (Hill, 2011) dataset were used to evaluation and hyperparameter selection, all with the log-linear response surface implemented as setting "B" in the NPCI package (Dorie, 2016). The 100 and 1000 replications were downloaded from (Johansson, 2017 (accessed July 19, 2018)) and are the exact same files used in (Shalit, Johansson, and Sontag, 2017; Louizos et al., 2017) which are the state-of-the-art baseline that was chosen to compare in the experiments of the present work.

This dataset is nowadays a strong **benchmark** framework for analysis the predictions results of a new machine learning technique applied to causal inference on observational data.

3.3 Other articles metrics

It is worth to mention other published articles evaluating ITE, ATE and PEHE errors for the reader to look further on them if intended. The results of the papers mentioned in this section had been collected using the same initial dataset from (Hill, 2011) but with slightly different methods for replications, different number of runs or not specifying how many replication were used.

In (Johansson, Shalit, and Sontag, 2016b) they run the IHDP dataset(Hill, 2011), on 100 replication experiments in order to perform hyperparameter tuning and 1000 replications for evaluation. All these replications were created using the NPCI package (Dorie, 2016) while selecting the log-linear response surface implemented as setting “B” in the mentioned tool. These results are not shown in this dissertation since the response surface chosen differs from the state-of-the-art results and papers published on the following years (Louizos et al., 2017; Shalit, Johansson, and Sontag, 2017). In (Johansson, Shalit, and Sontag, 2016b), to implement the **BART** results, they were based on Bayesian Additive Regression Trees (Chipman, George, and McCulloch, 2010) applying a non-linear regression model, following the implementation given in the *BayesTree Rpackage*.

In a recent publication, from the Proceeding of the 10th International Conference on Educational Data Mining (*Estimating Individual Treatment Effect from Educational Studies with Residual Counterfactual Networks*), referenced and run the experiments on the IHDP dataset (Hill, 2011). However, the authors *do not explicit the amount of replications* used to gather the metrics, *neither they express* if a log-linear "A" or "B" or any other method that was used to simulate the semi-synthetic dataset. Consequently, the results obtained by them can not be compared to this dissertation results and they are not shown on this work.

Chapter 4

Experiments

A series of runs with replications of the IHDP dataset were performed to ultimately predict all the factual y^F and counterfactual y^{CF} outcomes for every single *unit*. Subsequently, those values are inputed into the programming code produced by (Louizos et al., 2017) in which the ϵITE,ϵATE and $\epsilon PEHE$ errors are calculated to evaluate the performance of the applied machine learning methods. In this work, it is of particular interest correctly predicting the *Individual Treatment Effect*(ITE) that accounts for identifying the best possible *action* or *treatment* to a given *unit* x with its unique covariates (features).

This is a challenging goal since, as the reader might have clear by this point, is that neither the counterfactual outcome y^{CF} nor the average treatment effect with noise $mu0$, $mu1$ can be used at all to train the regressor models. Insted, these three values, as long with the factual outcome y^F are used to obtain the ϵITE,ϵATE and $\epsilon PEHE$ errors.

The experiments were run on 10, 100 and 1,000 replications, both **within-sample** and **out-of-sample**. The 10, 100 and 1,000 replications were downloaded from (Johansson, 2017 (accessed July 19, 2018)) and they are the same used to produce the results in (Shalit, Johansson, and Sontag, 2017; Louizos et al., 2017) from which the tables with their state-of-the-art errors will be also displayed in this section so the reader can compare with the outcomes of this work.

It is important to clarify and define here what **within-sample** and **out-of-sample** stands for. The definition given by (Shalit, Johansson, and Sontag, 2017) in its publication, being the same technique later followed by (Louizos et al., 2017) to perform, compare and show their results.

Within-sample: this test refers to all the errors (ITE, ATE nd PEHE) made by the predictions of the already trained model against the training and validation (if any) dataset. **Note bold** here, that this is not a trivial task since the model has already been trained with and unbalanced dataset (different number of samples in which treatments $t = 0$ and $t = 1$ was applied and observed) in which it is only known one treatment applied and the factual outcome of that treatment applied to an individual $x \in \lhd$. The other problem to overcome, is that in practice the population who received treatment $t = 1$ and the population who received $t = 0$ might come from completely different probability distributions. All these are common problems of observational data and they were mention in the **??**.

Out-of-sample: These predictions are made on a completely unseen, out of training or validation phase with new *units*. In this case, it is naturally harder to make predictions since the inputs might come from even different probability distributions from the training phase (already potentially unbalanced). The experiment procedure is the same, predictions for $t = 0$ and $t = 1$ are made for each single *unit*(input) of the testing dataset to later determine the errors ITE, ATE and PEHE.

Once the model is trained, it predicts for the each one of the inputs (*units*) using the treatment value $t = 0$ and consequently they predictions are made setting all the values of the treatment to $t = 1$. The subtraction between this two predictions for each input is known as the ITE and will ultimately define if the *patient* would be benefited or not by applying the treatment. Mathematically, it is represented by $\mathbb{E}[Y_1 - Y_0|x]$.

The machine learning algorithms implemented in python programming code by (*User guide: contents — scikit-learn 0.19.2 documentation*) were run with the default hyperparameters to obtain the above mentioned metrics that are finally shown in the tables displayed in this chapter. The hyperparameter tunning was done with 100 replications following the same methodology of the compared methods in the previously mentioned publications.

4.1 Machine learning methods applied to IHDP dataset

First, traditional, out of the shelf machine learning methods, were applied to the 10 replications IHDP dataset.

Their medians and variances across the 10 Replications for within-sample run are displayed in the Table 4.1, whereas in Table 4.2, it can be observed the **out-of-sample** errors (the lower the better) for 10 Replication of the IHDP dataset with the same algorithms.

TABLE 4.1: IHDP 10 replications with traditional machine learning algorithms - **Within sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
Support Vector Regressor (SVG)	2.62 ± 1.14	0.94 ± 0.35	2.73 ± 1.23
BayesianRidge	3.90 ± 1.99	0.97 ± 0.67	4.80 ± 2.80
LassoLars	4.76 ± 1.25	4.67 ± 0.57	7.40 ± 2.55
Lasso	4.76 ± 1.25	4.67 ± 0.57	7.40 ± 2.55
ARDRegression	3.92 ± 2.01	0.97 ± 0.74	4.80 ± 2.81
PassiveAggressiveRegressor	4.39 ± 2.09	1.54 ± 1.07	4.97 ± 2.92
TheilSenRegressor	3.93 ± 1.99	0.89 ± 0.63	4.78 ± 2.79
BaggingRegressor	5.14 ± 1.67	3.57 ± 0.47	6.27 ± 2.31
KNeighboursRegressor	5.14 ± 1.67	3.57 ± 0.47	6.27 ± 2.31
LinearRegression	3.92 ± 2.01	0.89 ± 0.65	4.79 ± 2.79

In the next experiment, Tables 4.4 and 4.4 show 100 Replications of the IHDP dataset were taking into account both for within-sample and out-of-sample respectively. In this case, it is remarkable that **the split in between training and testing was perform only over the training dataset** randomly. The intention was to prove if the results

TABLE 4.2: IHDP 10 replications with traditional machine learning algorithms - **Out-of-sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
Support Vector Regressor (SVG)	2.26 ± 0.63	1.24 ± 0.62	2.34 ± 0.98
BayesianRidge	3.54 ± 1.67	1.82 ± 1.35	4.13 ± 2.23
LassoLars	4.30 ± 0.89	5.48 ± 1.25	6.95 ± 2.06
Lasso	4.30 ± 0.89	5.48 ± 1.25	6.95 ± 2.06
ARDRegression	3.57 ± 1.70	1.83 ± 1.41	4.14 ± 2.27
PassiveAggressiveRegressor	4.19 ± 1.94	2.38 ± 1.75	4.45 ± 2.49
TheilSenRegressor	3.62 ± 1.68	1.76 ± 1.30	4.08 ± 2.21
BaggingRegressor	4.27 ± 1.18	3.92 ± 0.95	5.63 ± 1.81
KNeighboursRegressor	4.27 ± 1.18	3.92 ± 0.95	5.63 ± 1.81
LinearRegression	3.58 ± 1.70	1.77 ± 1.33	4.09 ± 2.22

obtained for the following experiments with the datasets already split into train and test, downloaded from (Johansson, 2017 (accessed July 19, 2018)), are similar or differ dramatically.

TABLE 4.3: IHDP 100 replications - **Within sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
Support Vector Regressor (SVG)	3.17 ± 0.40	0.82 ± 0.09	3.30 ± 0.42
BayesianRidge	4.49 ± 0.60	0.86 ± 0.16	5.65 ± 0.83
LassoLars	4.76 ± 0.36	4.57 ± 0.17	7.90 ± 0.77
Lasso	4.76 ± 0.36	4.57 ± 0.17	7.90 ± 0.77
ARDRegression	4.49 ± 0.60	0.81 ± 0.16	5.64 ± 0.83
PassiveAggressiveRegressor	5.49 ± 0.75	0.83 ± 0.14	5.66 ± 0.83
TheilSenRegressor	4.45 ± 0.59	0.79 ± 0.15	5.63 ± 0.83
BaggingRegressor	5.35 ± 0.49	3.46 ± 0.14	6.78 ± 0.70
KNeighboursRegressor	5.35 ± 0.49	3.46 ± 0.14	6.78 ± 0.70
LinearRegression	4.53 ± 0.60	0.79 ± 0.16	5.63 ± 0.83

TABLE 4.4: IHDP 100 replications - **Out-of-sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
Support Vector Regressor (SVG)	2.79 ± 0.27	0.86 ± 0.13	3.25 ± 0.42
BayesianRidge	4.27 ± 0.54	1.02 ± 0.26	5.37 ± 0.78
LassoLars	4.75 ± 0.39	4.51 ± 0.23	7.57 ± 0.71
Lasso	4.75 ± 0.39	4.51 ± 0.23	7.57 ± 0.71
ARDRegression	4.27 ± 0.54	1.00 ± 0.26	5.36 ± 0.78
PassiveAggressiveRegressor	5.28 ± 0.69	1.00 ± 0.21	5.36 ± 0.77
TheilSenRegressor	4.24 ± 0.53	0.99 ± 0.25	5.35 ± 0.78
BaggingRegressor	4.93 ± 0.43	3.19 ± 0.18	6.23 ± 0.63
KNeighboursRegressor	4.93 ± 0.43	3.19 ± 0.18	6.23 ± 0.63
LinearRegression	4.31 ± 0.55	0.99 ± 0.26	5.36 ± 0.79

Consequently, in Table 4.5 and Table 4.6 it can be observed, the results for the already split in training and test obtained from (Johansson, 2017 (accessed July 19, 2018)) which accounts for the exact same dataset used in (Louizos et al., 2017; Shalit, Johansson, and Sontag, 2017). As mentioned in this thesis, the hyperparameter tunning was performed on this number of replications, if any.

TABLE 4.5: IHDP 100 replications already split dataset - **Within sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
Support Vector Regressor (SVG)	3.05 ± 0.38	0.76 ± 0.08	3.17 ± 0.40
BayesianRidge	4.44 ± 0.58	0.80 ± 0.14	5.61 ± 0.83
LassoLars	4.76 ± 0.36	4.55 ± 0.17	7.88 ± 0.76
Lasso	4.76 ± 0.36	4.55 ± 0.17	7.88 ± 0.76
ARDRegression	4.45 ± 0.59	0.77 ± 0.15	5.61 ± 0.83
PassiveAggressiveRegressor	5.03 ± 0.62	0.83 ± 0.13	5.63 ± 0.82
TheilSenRegressor	4.40 ± 0.57	0.72 ± 0.13	5.60 ± 0.82
BaggingRegressor	5.31 ± 0.48	3.41 ± 0.14	6.72 ± 0.69
KNeighboursRegressor	5.31 ± 0.48	3.41 ± 0.14	6.72 ± 0.69
LinearRegression	4.48 ± 0.59	0.75 ± 0.14	5.60 ± 0.82

TABLE 4.6: IHDP 100 replications already split dataset - **Out-of-sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
Support Vector Regressor (SVG)	2.84 ± 0.28	0.73 ± 0.07	3.49 ± 0.49
BayesianRidge	4.41 ± 0.57	0.81 ± 0.11	5.74 ± 0.89
LassoLars	4.65 ± 0.34	4.31 ± 0.14	7.96 ± 0.82
Lasso	4.65 ± 0.34	4.31 ± 0.14	7.96 ± 0.82
ARDRegression	4.42 ± 0.58	0.78 ± 0.11	5.73 ± 0.89
PassiveAggressiveRegressor	4.95 ± 0.59	1.01 ± 0.17	5.78 ± 0.89
TheilSenRegressor	4.38 ± 0.56	0.85 ± 0.13	5.74 ± 0.89
BaggingRegressor	4.95 ± 0.46	2.98 ± 0.10	6.65 ± 0.75
KNeighboursRegressor	4.95 ± 0.46	2.98 ± 0.10	6.65 ± 0.75
LinearRegression	4.45 ± 0.58	0.78 ± 0.11	5.73 ± 0.89

With 1,000 replications, it can be compared, both within sample and out-of-sample with the results obtained in (Shalit, Johansson, and Sontag, 2017; Louizos et al., 2017). The same semi-synthetic dataset IHDP by (Hill, 2011) with *log-linear response setting "A"* generated using the code from (Dorie, 2016) was used to perform both type of measures.

It can be observed four different tables for the 1,000 replications. In pairs, two of them (Tables 4.7, 4.8) were obtained without normalization of the *input features* (covariates), the other couple was obtained by scaling from $[0, 1]$ using the *MinMaxScaler()* from the scikit-learn library. The results improved, not significantly, but enough for keeping the scaling as the presented final result of the methods in following section.

TABLE 4.7: IHDP 100 replications - No scaling - **Within sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
Support Vector Regressor (SVG)	3.09 ± 0.12	0.69 ± 0.02	3.21 ± 0.13
BayesianRidge	4.59 ± 0.19	0.78 ± 0.04	5.81 ± 0.26
LassoLars	4.65 ± 0.10	4.40 ± 0.04	7.91 ± 0.24
Lasso	4.65 ± 0.10	4.40 ± 0.04	7.91 ± 0.24
ARDRegression	4.59 ± 0.19	0.76 ± 0.04	5.80 ± 0.26
PassiveAggressiveRegressor	5.41 ± 0.22	0.90 ± 0.05	5.85 ± 0.26
TheilSenRegressor	4.55 ± 0.18	0.70 ± 0.03	5.79 ± 0.26
BaggingRegressor	5.31 ± 0.15	3.28 ± 0.04	6.76 ± 0.21
KNeighboursRegressor	5.31 ± 0.15	3.28 ± 0.04	6.76 ± 0.21
LinearRegression	4.63 ± 0.19	0.73 ± 0.04	5.80 ± 0.26

TABLE 4.8: IHDP 1000 replications - No Scaling - **Out-of-sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
Support Vector Regressor (SVG)	2.81 ± 0.09	0.78 ± 0.03	3.37 ± 0.14
BayesianRidge	4.57 ± 0.19	0.98 ± 0.05	5.79 ± 0.26
LassoLars	4.66 ± 0.11	4.41 ± 0.05	7.90 ± 0.24
Lasso	4.66 ± 0.11	4.41 ± 0.05	7.90 ± 0.24
ARDRegression	4.58 ± 0.19	0.96 ± 0.05	5.78 ± 0.26
PassiveAggressiveRegressor	5.42 ± 0.22	1.13 ± 0.07	5.83 ± 0.27
TheilSenRegressor	4.54 ± 0.19	0.95 ± 0.05	5.78 ± 0.26
BaggingRegressor	4.95 ± 0.14	3.09 ± 0.05	6.54 ± 0.22
KNeighboursRegressor	4.95 ± 0.14	3.09 ± 0.05	6.54 ± 0.22
LinearRegression	4.61 ± 0.19	0.94 ± 0.05	5.78 ± 0.26

TABLE 4.9: IHDP 100 replications - Scaled - **Within sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
Support Vector Regressor (SVG)	2.38 ± 0.08	0.33 ± 0.02	2.77 ± 0.12
BayesianRidge	4.58 ± 0.19	0.73 ± 0.04	5.80 ± 0.26
LassoLars	4.65 ± 0.10	4.40 ± 0.04	7.91 ± 0.24
Lasso	4.65 ± 0.10	4.40 ± 0.04	7.91 ± 0.24
ARDRegression	4.59 ± 0.19	0.76 ± 0.04	5.80 ± 0.26
PassiveAggressiveRegressor	5.47 ± 0.22	1.02 ± 0.06	5.88 ± 0.26
TheilSenRegressor	4.68 ± 0.19	0.69 ± 0.03	5.79 ± 0.26
BaggingRegressor	4.77 ± 0.13	2.67 ± 0.03	6.37 ± 0.21
KNeighboursRegressor	4.77 ± 0.13	2.67 ± 0.03	6.37 ± 0.21
RANSACRegressor	4.93 ± 0.20	1.64 ± 0.09	6.09 ± 0.26
HuberRegressor	4.44 ± 0.18	0.67 ± 0.03	5.79 ± 0.25
ElasticNet	4.65 ± 0.10	4.40 ± 0.04	7.91 ± 0.24
LinearRegression	4.63 ± 0.19	0.73 ± 0.04	5.80 ± 0.26

TABLE 4.10: IHDP 1000 replications - No Scaling - **Out-of-sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
Support Vector Regressor (SVG)	2.44 ± 0.08	0.45 ± 0.03	2.81 ± 0.13
BayesianRidge	4.55 ± 0.19	0.95 ± 0.05	5.78 ± 0.26
LassoLars	4.66 ± 0.11	4.41 ± 0.05	7.90 ± 0.24
Lasso	4.66 ± 0.11	4.41 ± 0.05	7.90 ± 0.24
ARDRegression	4.58 ± 0.19	0.96 ± 0.05	5.78 ± 0.26
PassiveAggressiveRegressor	5.44 ± 0.22	1.18 ± 0.07	5.87 ± 0.26
TheilSenRegressor	4.68 ± 0.19	0.95 ± 0.05	5.78 ± 0.26
BaggingRegressor	4.46 ± 0.13	2.33 ± 0.04	6.12 ± 0.22
KNeighboursRegressor	4.46 ± 0.13	2.33 ± 0.04	6.12 ± 0.22
RANSACRegressor	4.91 ± 0.20	1.73 ± 0.09	6.06 ± 0.27
HuberRegressor	4.44 ± 0.18	0.92 ± 0.05	5.77 ± 0.26
ElasticNet	4.66 ± 0.11	4.41 ± 0.05	7.90 ± 0.24
LinearRegression	4.61 ± 0.19	0.94 ± 0.05	5.78 ± 0.26

Consequently, Logistic Regression with multi-class as multinomial predictor has been applied. The performance is way bellow the regressors, being the main reason that, when encoding the target values to assign them a probability, these are not the same that are needed to be predicted. Also when decoding the predictions, precision is lost. The l2 norm has been used with two different solvers: *newton-cg* and *lbfgs*. This results are displayed in Table 4.11 and Table 4.12.

TABLE 4.11: IHDP 100 replications logistic regressions - **Within sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
LogisticRegression - L2 (NEWTON-CG)	7.77 ± 0.76	4.40 ± 0.17	7.77 ± 0.76
LogisticRegression - L2 (lbfgs)	7.77 ± 0.76	4.40 ± 0.17	7.77 ± 0.76

TABLE 4.12: IHDP 100 replications logistic regressions - **Out-of-sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
LogisticRegression - L2 (NEWTON-CG)	5.90 ± 0.57	2.41 ± 0.11	7.21 ± 0.85
LogisticRegression - L2 (lbfgs)	5.90 ± 0.57	2.41 ± 0.11	7.21 ± 0.85

From all these tables, the method which obtained the best results, was consistently the Support Vector Regressor. Therefore, a few runs of hyper-parameters tunning were done. The errors observed were even smaller, so the final hyper-parameters selected for this dataset were: Radial Basis Function (rbf), C=1e3 and gamma=0.01. The selection was performed within sample and out-of-sample but for 100 replications of the dataset, this is the same method the authors (Shalit, Johansson, and Sontag, 2017; Johansson, Shalit, and Sontag, 2016a; Louizos et al., 2017) state to use

for their own hyper-parameter selection. In Table 4.13 and Table 4.14 the results of running SVR hyperparameter selection with the final results shown.

TABLE 4.13: IHDP 100 replications SVR Hyper-parameters tunning - **Within sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
SVR-rbf-1e3-g0.1	3.17 ± 0.40	0.82 ± 0.09	3.30 ± 0.42
SVR-rbf-1e3-g0.05	2.71 ± 0.34	0.45 ± 0.07	2.78 ± 0.36
SVR-rbf-1e3-g0.01	2.35 ± 0.29	0.24 ± 0.03	2.32 ± 0.31
SVR-rbf-1e3-g0.001	3.65 ± 0.45	0.52 ± 0.09	4.51 ± 0.65
SVR-rbf-1e3-g0.0001	4.28 ± 0.55	0.76 ± 0.11	5.61 ± 0.82
SVR-rbf-1e3-g0.00001	4.25 ± 0.52	1.49 ± 0.10	5.97 ± 0.81
SVR-rbf-1e10-g0.1	3.17 ± 0.40	0.82 ± 0.09	3.30 ± 0.42
SVR-rbf-1e20-g0.1	3.17 ± 0.40	0.82 ± 0.09	3.30 ± 0.42
SVR-rbf-1e30-g0.1	3.17 ± 0.40	0.82 ± 0.09	3.30 ± 0.42
SVR-poly-1e3-degree2	2.50 ± 0.29	0.28 ± 0.03	2.53 ± 0.30
SVR-poly-1e3-degree1	2.50 ± 0.29	0.28 ± 0.03	2.53 ± 0.30
SVR-poly-1e3-degree4	2.50 ± 0.29	0.28 ± 0.03	2.53 ± 0.30
SVR-poly-1e10-degree2	2.99 ± 0.34	0.41 ± 0.06	3.00 ± 0.39

TABLE 4.14: IHDP 100 replications SVR Hyper-parameters tunning - **Out-of-sample**

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
SVR-rbf-1e3-g0.1	2.79 ± 0.27	0.86 ± 0.13	3.25 ± 0.42
SVR-rbf-1e3-g0.05	2.66 ± 0.24	0.53 ± 0.10	2.71 ± 0.35
SVR-rbf-1e3-g0.01	2.50 ± 0.23	0.31 ± 0.05	2.26 ± 0.31
SVR-rbf-1e3-g0.001	3.45 ± 0.40	0.77 ± 0.16	4.23 ± 0.62
SVR-rbf-1e3-g0.0001	4.09 ± 0.50	0.96 ± 0.21	5.31 ± 0.77
SVR-rbf-1e3-g0.00001	4.05 ± 0.47	1.59 ± 0.18	5.65 ± 0.75
SVR-rbf-1e10-g0.1	2.79 ± 0.27	0.86 ± 0.13	3.25 ± 0.42
SVR-rbf-1e20-g0.1	2.79 ± 0.27	0.86 ± 0.13	3.25 ± 0.42
SVR-rbf-1e30-g0.1	2.79 ± 0.27	0.86 ± 0.13	3.25 ± 0.42
SVR-poly-1e3-degree2	2.87 ± 0.22	0.38 ± 0.05	2.48 ± 0.29
SVR-poly-1e3-degree1	2.87 ± 0.22	0.38 ± 0.05	2.48 ± 0.29
SVR-poly-1e3-degree4	2.87 ± 0.22	0.38 ± 0.05	2.48 ± 0.29
SVR-poly-1e10-degree2	3.21 ± 0.33	0.48 ± 0.06	2.95 ± 0.39

Finally, the **final results obtained** by this thesis and the run experiments are displayed in **Table 4.10** and **Table 4.9**, whereas in **Table 4.15** and **Table 4.16** show the results obtained in publication (Shalit, Johansson, and Sontag, 2017).

TABLE 4.15: ICML 2017 - "Estimating individual treatment effect: generalization bounds and algorithms" (Shalit, Johansson, and Sontag, 2017)

	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}
OLS/LR-1	5.8 ± .3	.73 ± .04
OLS/LR-2	2.4 ± .1	.14 ± .01
BLR	5.8 ± .3	.72 ± .04
k-NN	2.1 ± .1	.14 ± .01
TMLE	5.0 ± .2	.30 ± .01
BART	2.1 ± .1	.23 ± .01
RAND.FOR.	4.2 ± .2	.73 ± .05
CAUS.FOR.	3.8 ± .2	.18 ± .01
BNN	2.2 ± .1	.37 ± .03
TARNET	.88 ± .0	.26 ± .01
CFR MMD	.73 ± .0	.30 ± .01
CFR WASS	.71 ± .0	.25 ± .01

Within sample IHDP 1000 replications

TABLE 4.16: ICML 2017 - "Estimating individual treatment effect: generalization bounds and algorithms" (Shalit, Johansson, and Sontag, 2017)

	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}
OLS/LR-1	5.8 ± .3	.94 ± .06
OLS/LR-2	2.5 ± .1	.31 ± .02
BLR	5.8 ± .3	.93 ± .05
k-NN	4.1 ± .2	.79 ± .05
BART	2.3 ± .1	.34 ± .02
RAND.FOR.	6.6 ± .3	.96 ± .06
CAUS.FOR.	3.8 ± .2	.40 ± .03
BNN	2.1 ± .1	.42 ± .03
TARNET	.95 ± .0	.28 ± .01
CFRMMD	.78 ± .0	.31 ± .01
CFRWASS	.76 ± .0	.27 ± .01

Out-of-sample IHDP 1000 replications

4.2 Other experiments

4.2.1 Recursive Feature Elimination

Even thought in the Related Work section, powerful Feature Selection methods implementation publications were shown, there was a experimented performed in the developed code that performs machine learning Recursive Feature Elimination (RFE) using the sci-kit learn library framework.

In addition, assuming Strong Ignorability on the dataset studied, it would not be appropriate to perform such experiment but due to the nature of the machine learning regressors and their sensibility to highly correlated input features this might relief some of the errors made by them.

The results perform significantly worst in all algorithms for the causality inference metrics detailed in this work, thus the results are not show but can be revised by the reader in the code implementation for further analysis.

4.2.2 Domain Adaptation Neural Networks

A Domain Adaptation Neural Networks implementation was tested on just 10 replications of the IHDP dataset. However, the code in the github repository of Dr. Spyros Samothrakis, was executed to obtain the results for 10 replications, in this work, the code uploaded contains the straightforward implementation for the 1,000 replications used in the other experiments.

The results shown bellow in Table 4.17, clearly state promising results. Although the results are not directly comparable with the ones in the previous subsection, the code uploaded is ready to run the 1,000 replication in a GPU powered machine. In terms of CPU the estimated finished time was about 4 days and a half with a Intel Dual Core i7.

Domain Adaptation algorithms are a promising field to explore ITE and ATE predictions due to its architectural design.

TABLE 4.17: Domain Adaptation Neural Networks

	ϵ_{ITE}	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$
DANN (Within-sample)	1.18±0.17	0.12 ± 0.04	1.02 ± 0.48
DANN (Out-of-sample)	1.20±0.11	0.17 ± 0.08	0.76 ± 0.23

Within-sample and Out-of-sample IHDP 10 replications

4.3 Discussion

As it can be clearly noticed, machine learning regressor algorithms applied in this dissertation are very close to the one obtained by the work published by the cited compared authors.

It is remarkable that no custom metric function, Integral Probability Metric to overcome the unbalanced treated dataset, or any other custom loss function were applied to obtain the shown results in Table 4.10 and Table 4.9.

It seems to be an excessive amount of effort from the authors, that leads into complicated methods, not gaining much more into causality prediction from observational data.

However, they claim that with more unbalanced representations of the feature space or treatment assignment, their methods can help to overcome this problem much better than out-of-the-box machine learning algorithms. No other metrics are reported on heavily unbalanced treatment assignment datasets.

Finally, the 10 replications of the Domain Adaptation Neural Networks training and testing errors showed promising results that needs to be addressed in future works.

Chapter 5

Conclusions

5.1 Concluding Remarks

The results obtained applying machine learning regressors with significant less, or no added, complexity out-of-the-box to predict the factual and counterfactual outcomes given a *unit* are close to the ones obtained in more elaborated and custom techniques that are implemented as the state-of-the-art performance results.

It has to be taken into account that no custom metric functions, nor special preprocessing steps, except from scaling the features, which is part of the must do tasks when using machine learning algorithms) have been performed to achieve similar results to the state-of-the-art metrics achieved in the mentioned and compared papers in **??**.

In addition, this dissertation shows the results for *not applied before* machine learning techniques on the adopted benchmark IHDP dataset for performing predictions on both the factual and counterfactual outcomes, to later present the ITE, ATE and PEHE error calculations. This was the main goal of this thesis but it changed when the obtained metrics were almost as close as the ones in the state-of-the-art numbers.

It is important to notice that there are machine learning techniques that had been introduced in the last years that are potentially more suitable than both machine learning regressors and custom or generalized metric and error functions, like Domain Adaptation Neural Networks, as well as other methods from the Deep Learning literature. Moreover, there are continuous space causality from observational data that include more than two possible outcomes to apply that are substantially more suitable to solve with Reinforcement Learning algorithms better than any other Deep Neural Network or Regressor.

Finally, this work is intended to cover a considerably empty space of straightforward definitions to apply machine learning to causality. Although in the last two years, several noticeable papers were published, there are difficult to follow when relating terms from the causal inference field to the computer and data sciences background researchers. I gave my best to compile, define, explain, detail and relate, causal inference with machine learning terminology.

5.2 Future work

Future directions on this work will include four different approaches that should be taken.

First, it would be important to try the applied machine learning methods to other benchmark datasets and compare the results with other published papers and algorithms with the same or more complexity.

Second, extending the functionality developed for a binary treatment and a continuous output to a multi-valued treatment To the best of my knowledge, it should not be costly to perform such modification in the code, however at least one new dataset that supports this kind of treatment size would need to be processed.

Third, applying this method to perform binary factual predictions and Policy Risk threshold for which a treatment should be applied or not, should be an important next step regarding causal inference from observational data. The machine learning algorithms applied in this work are suitable to test with this type of datasets, solving a common real life problem in the field.

Fourth, implement Domain Adaptation Neural Networks on the IHDP 1,000 replications dataset is a very promising task due to both the architectural design of the algorithm, as well as the outperforming state-of-the-art precision that they had for the experiment run.

Lastly, the application of these methods on causal datasets that accounts for outcomes that varies against the application of time and applied treatments are framed within time series problems in the continuous space. These kind of datasets will be possibly the next focus on the researchers of machine learning applied to treatments applied over time.

Bibliography

Alaa, Ahmed M, Michael Weisz, and Mihaela Van Der Schaar (2017). *Deep Counterfactual Networks with Propensity-Dropout*. Tech. rep. arXiv: arXiv:1706.05966v1. URL: https://arxiv.org/pdf/1706.05966.pdf.

Arjas, Elja and Jan Parner. *Causal Reasoning from Longitudinal Data*. DOI: 10.2307/4616822. URL: https://www.jstor.org/stable/4616822.

Atan, Onur, William R Zame, and Mihaela Van Der Schaar (2018). *Counterfactual Policy Optimization Using Domain-Adversarial Neural Networks*. Tech. rep. URL: http://medianetlab.ee.ucla.edu/papers/cf{_}treat{_}v5.

Atan, Onur et al. (2016). *Constructing Effective Personalized Policies Using Counterfactual Inference from Biased Data Sets with Many Features*. Tech. rep. arXiv: arXiv:1612.08082v1.

Athey, Susan and Guido Imbens (2016). "Recursive partitioning for heterogeneous causal effects." In: *Proceedings of the National Academy of Sciences of the United States of America* 113.27, pp. 7353–60. ISSN: 1091-6490. DOI: 10.1073/pnas.1510489113. URL: http://www.ncbi.nlm.nih.gov/pubmed/27382149http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4941430.

Athey, Susan and Guido W Imbens. *Recursive Partitioning for Heterogeneous Causal Effects* *. Tech. rep. arXiv: arXiv:1504.01132v3. URL: https://arxiv.org/pdf/1504.01132.pdf.

Austin, Peter C (2011). "An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies." In: *Multivariate behavioral research* 46.3, pp. 399–424. ISSN: 1532-7906. DOI: 10.1080/00273171.2011.568786. URL: http://www.ncbi.nlm.nih.gov/pubmed/21818162http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3144483.

Bang, Heejung and James M Robins (2005). "Doubly Robust Estimation in Missing Data and Causal Inference Models". In: DOI: 10.1111/j.1541-0420.2005.00377.x.

Ben-David, Shai et al. (2007). *Analysis of Representations for Domain Adaptation*. URL: https://papers.nips.cc/paper/2983-analysis-of-representations-for-domain-adaptation.

Beygelzimer, Alina and John Langford (2008). "The Offset Tree for Learning with Partial Labels". In: arXiv: 0812.4044. URL: http://arxiv.org/abs/0812.4044.

Blitzer, John, Ryan McDonald, and Fernando Pereira (2006). "Domain adaptation with structural correspondence learning". In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing - EMNLP '06*. Morristown, NJ, USA: Association for Computational Linguistics, p. 120. ISBN: 1932432736. DOI: 10.3115/1610075.1610094. URL: http://portal.acm.org/citation.cfm?doid=1610075.1610094.

Bottou, Léon et al. (2013). *Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising*. Tech. rep., pp. 3207–3260. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2013/11/bottou13a.pdf.

Chernozhukov, Victor et al. (2016). "Double/Debiased Machine Learning for Treatment and Causal Parameters". In: arXiv: 1608.00060. URL: http://arxiv.org/abs/1608.00060.

Chipman, Hugh A., Edward I. George, and Robert E. McCulloch (2010). "BART: Bayesian additive regression trees". In: *The Annals of Applied Statistics* 4.1, pp. 266–298. ISSN: 1932-6157. DOI: 10.1214/09-AOAS285. URL: http://projecteuclid.org/euclid.aoas/1273584455.

Daumé, Hal (2009). "Frustratingly Easy Domain Adaptation". In: arXiv: 0907.1815. URL: http://arxiv.org/abs/0907.1815.

Dorie, Vincent. (2016). *NPCI: Non-parametrics for Causal Inference*. URL: https://github.com/vdorie/npci.

Doroudi, Shayan, Philip S Thomas, and Emma Brunskill (2017). *Importance Sampling for Fair Policy Selection* *. Tech. rep. URL: https://www.ijcai.org/proceedings/2018/0729.pdf.

Dudik, Miroslav, John Langford, and Lihong Li (2011). "Doubly Robust Policy Evaluation and Learning". In: arXiv: 1103.4601. URL: http://arxiv.org/abs/1103.4601.

Gan, Chuang et al. (2016). "Webly-Supervised Video Recognition by Mutually Voting for Relevant Web Images and Web Video Frames". In: pp. 849–866. DOI: 10.1007/978-3-319-46487-9_52. URL: http://link.springer.com/10.1007/978-3-319-46487-9{_}52.

Gelman, Andrew. and Jennifer Hill (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press, p. 625. ISBN: 9780521686891.

Gross, Ruth T (1993). *Infant Health and Development Program (IHDP): Enhancing the Outcomes of Low Birth Weight, Premature Infants in the United States, 1985-1988*. DOI: 10.3886/ICPSR09795.v1.

Hill, Jennifer L. (2011). "Bayesian Nonparametric Modeling for Causal Inference". In: *Journal of Computational and Graphical Statistics* 20.1, pp. 217–240. ISSN: 1061-8600. DOI: 10.1198/jcgs.2010.08162. URL: http://www.tandfonline.com/doi/abs/10.1198/jcgs.2010.08162.

Hoiles, William and Mihaela Van Der Schaar (2016). "Bounded Off-policy Evaluation with Missing Data for Course Recommendation and Curriculum Design". In: ICML'16, pp. 1596–1604. URL: http://dl.acm.org/citation.cfm?id=3045390.3045559.

Hoyer, Patrik O et al. *Nonlinear causal discovery with additive noise models*. Tech. rep. URL: https://is.tuebingen.mpg.de/fileadmin/user{_}upload/files/publications/NIPS2008-Hoyer-neu{_}5406[0].pdf.

Jiang, Nan and Lihong Li (2015). "Doubly Robust Off-policy Value Evaluation for Reinforcement Learning". In: arXiv: 1511.03722. URL: http://arxiv.org/abs/1511.03722.

Joachims, Thorsten and Adith Swaminathan (2016). "Counterfactual Evaluation and Learning for Search, Recommendation and Ad Placement". In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '16. Pisa, Italy: ACM, pp. 1199–1201. ISBN: 978-1-4503-4069-4. DOI: 10.1145/2911451.2914803. URL: http://doi.acm.org/10.1145/2911451.2914803.

Johansson, Fredrik D. (2017 (accessed July 19, 2018)). *Fredrik D. Johansson, PhD - MIT Personal Website*. URL: https://stuff.mit.edu/afs/athena.mit.edu/user/f/r/fredrikj/www/.

Johansson, Fredrik D, Uri Shalit, and David Sontag. "Learning Representations for Counterfactual Inference". In: (). URL: https://people.csail.mit.edu/dsontag/papers/JohanssonShalitSontag{_}icml16.pdf.

– (2016a). *Learning Representations for Counterfactual Inference*. Tech. rep. URL: http://proceedings.mlr.press/v48/johansson16.pdf.

– (2016b). *Learning Representations for Counterfactual Inference*. Tech. rep. URL: http://proceedings.mlr.press/v48/johansson16.pdf.

Kang, Joseph D. Y. and Joseph L. Schafer (2007). "Demystifying Double Robustness: A Comparison of Alternative Strategies for Estimating a Population Mean from Incomplete Data". In: *Statistical Science* 22.4, pp. 523–539. ISSN: 0883-4237. DOI: 10.1214/07-STS227. URL: http://projecteuclid.org/euclid.ss/1207580167.

Lang, Ken (1995). "NewsWeeder: Learning to Filter Netnews". In: *Machine Learning Proceedings 1995*. Elsevier, pp. 331–339. DOI: 10.1016/B978-1-55860-377-6.50048-7.

Lok, Judith J. (2008). "Statistical modeling of causal effects in continuous time". In: *The Annals of Statistics* 36.3, pp. 1464–1507. ISSN: 0090-5364. DOI: 10.1214/009053607000000820. URL: http://projecteuclid.org/euclid.aos/1211819571.

Louizos, Christos et al. (2017). "Causal Effect Inference with Deep Latent-Variable Models arXiv : 1705 . 08821v2 [stat . ML] 6 Nov 2017". In: Nips. arXiv: arXiv:1705.08821v2.

Maathuis, Marloes H et al. (2010). "Predicting causal effects in large-scale systems from observational data". In: *Nature Methods* 7.4, pp. 247–248. ISSN: 1548-7091. DOI: 10.1038/nmeth0410-247. URL: http://www.ncbi.nlm.nih.gov/pubmed/20354511http://www.nature.com/articles/nmeth0410-247.

Mooij, Joris M et al. (2016). *Distinguishing Cause from Effect Using Observational Data: Methods and Benchmarks*. Tech. rep., pp. 1–102. URL: http://jmlr.org/papers/volume17/14-518/14-518.pdf.

Morgan, Stephen L. and Christopher Winship (2014). *Counterfactuals and Causal Inference*. Cambridge: Cambridge University Press. ISBN: 9781107587991. DOI: 10.1017/CBO9781107587991.

Nahum-Shani, Inbal et al. (2012). "Q-learning: A data analysis method for constructing adaptive interventions." In: *Psychological Methods* 17.4, pp. 478–494. ISSN: 1939-1463. DOI: 10.1037/a0029373. URL: http://www.ncbi.nlm.nih.gov/pubmed/23025434http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3747013http://doi.apa.org/getdoi.cfm?doi=10.1037/a0029373.

Prentice, Ross (1976). "Use of the Logistic Model in Retrospective Studies". In: *Biometrics* 32.3, p. 599. ISSN: 0006341X. DOI: 10.2307/2529748. URL: https://www.jstor.org/stable/2529748?origin=crossref.

Păduraru, Cosmin et al. (2012). *An Empirical Analysis of Off-policy Learning in Discrete MDPs*. Tech. rep., pp. 89–101. URL: http://proceedings.mlr.press/v24/paduraru12a/paduraru12a.pdf.

Robins, James (1986). "A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect". In: *Mathematical Modelling* 7.9-12, pp. 1393–1512. ISSN: 0270-0255. DOI: 10.1016/0270-0255(86)90088-6. URL: https://www.sciencedirect.com/science/article/pii/0270025586900886.

Rosenbaum, Paul R. (2002). "Observational Studies". In: pp. 1–17. DOI: 10.1007/978-1-4757-3692-2_1. URL: http://link.springer.com/10.1007/978-1-4757-3692-2{_}1.

Rosenbaum, Paul R and Donald B Rubin (1983). *The Central Role of the Propensity Score in Observational Studies for Causal Effects*. Tech. rep. 1, pp. 41–55. URL: http://www.stat.cmu.edu/{~}ryantibs/journalclub/rosenbaum{_}1983.pdf.

Rubin, Donald B. *Causal Inference Using Potential Outcomes: Design, Modeling, Decisions*. DOI: 10.2307/27590541. URL: https://www.jstor.org/stable/27590541.

Rubin, Donald B. (1978). "Bayesian Inference for Causal Effects: The Role of Randomization". In: *The Annals of Statistics* 6.1, pp. 34–58. ISSN: 0090-5364. DOI: 10.1214/aos/1176344064.

Rubin, Donald B (2005). "Causal Inference Using Potential Outcomes". In: *Journal of the American Statistical Association* 100.469, pp. 322–331. DOI: 10.1198/016214504000001880. eprint: https://doi.org/10.1198/016214504000001880. URL: https://doi.org/10.1198/016214504000001880.

Schulam, Peter and Suchi Saria. *Reliable Decision Support using Counterfactual Models*. Tech. rep. arXiv: arXiv:1703.10651v4. URL: https://arxiv.org/pdf/1703.10651.pdf.

Shalit, Uri, Fredrik D Johansson, and David Sontag (2017). *Supplemental Materials for: Estimating individual treatment effect: generalization bounds and algorithms A. Proofs*. Tech. rep. URL: http://proceedings.mlr.press/v70/shalit17a/shalit17a-supp.pdf.

Shalit, Uri and David Sontag (2016). "CAUSAL INFERENCE FOR OBSERVATIONAL STUDIES". In: URL: https://cs.nyu.edu/{~}shalit/slides.pdf.

Sutton, Richard S and Andrew G Barto (2017). ****Complete Draft****. Tech. rep. URL: http://incompleteideas.net/book/bookdraft2017nov5.pdf.

Swaminathan, Adith and Thorsten Joachims (2015a). *Counterfactual Risk Minimization: Learning from Logged Bandit Feedback*. URL: http://proceedings.mlr.press/v37/swaminathan15.html.

– (2015b). *The Self-Normalized Estimator for Counterfactual Learning*.

Tekin, Cem and Mihaela Van Der Schaar (2018). *Episodic Multi-armed Bandits*. Tech. rep. arXiv: arXiv:1508.00641v4. URL: https://arxiv.org/pdf/1508.00641.pdf.

Tian, Lu et al. (2014). "A Simple Method for Estimating Interactions between a Treatment and a Large Number of Covariates." In: *Journal of the American Statistical Association* 109.508, pp. 1517–1532. ISSN: 0162-1459. DOI: 10.1080/01621459.2014.951443. URL: http://www.ncbi.nlm.nih.gov/pubmed/25729117http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4338439.

Triantafillou, Sofia and Ioannis Tsamardinos (2015). *Constraint-based Causal Discovery from Multiple Interventions over Overlapping Variable Sets*. Tech. rep., pp. 2147–2205. URL: http://jmlr.org/papers/volume16/triantafillou15a/triantafillou15a.pdf.

User guide: contents — scikit-learn 0.19.2 documentation. URL: http://scikit-learn.org/stable/user{_}guide.html (visited on 08/17/2018).

Wager, Stefan and Susan Athey (2015). "Estimation and Inference of Heterogeneous Treatment Effects using Random Forests". In: arXiv: 1510.04342. URL: http://arxiv.org/abs/1510.04342.

– (2017). *Estimation and Inference of Heterogeneous Treatment Effects using Random Forests* *. Tech. rep. arXiv: arXiv:1510.04342v4. URL: http://arxiv.org/abs/1405.0352..

Zhang, Kun et al. (2013). *Domain Adaptation under Target and Conditional Shift*. URL: http://proceedings.mlr.press/v28/zhang13d.html.

Zhao, Siyuan and Neil Heffernan. *Estimating Individual Treatment Effect from Educational Studies with Residual Counterfactual Networks*. Tech. rep.