12-2022

An Ensemble Machine Learning Approach To Causal Inference in High-Dimensional Settings

Sami Saad Alanazi

UNIVERSITY OF NORTHERN COLORADO

Greeley, Colorado

The Graduate School

AN ENSEMBLE MACHINE LEARNING APPROACH TO CAUSAL INFERENCE IN HIGH-DIMENSIONAL SETTINGS

A Dissertation Submitted in Partial Fulfillment
Of the Requirements for the Degree of
Doctor of Philosophy

Sami Saad Alanazi

College of Educational and Behavioral Sciences
Department of Applied Statistics and Research Methods

December 2022

This dissertation by: Sami Saad Alanazi

Entitled: *An Ensemble Machine Learning Approach to Causal Inference in High-Dimensional Settings*

has been approved as meeting the requirements for the Degree of Doctor of Philosophy in the College of Educational and Behavioral Sciences in the Department of Applied Statistics and Research Methods.

Accepted by the Doctoral Committee

Han Yu, Ph.D., Research Advisor

Khalil Shafie, Ph.D., Committee Member

Bahaedin Khaledi, Ph.D., Committee Member

Yoon Tae Sung, Ph.D., Faculty Representative

Date of Dissertation Defense _____

Accepted by the Graduate School

Jeri-Anne Lyons, Ph.D.
Dean of the Graduate School
Associate Vice President of Research

ABSTRACT

Alanazi, Sami Saad. *An ensemble machine learning approach to causal inference in high-dimensional settings*. Published Doctor of Philosophy dissertation, University of Northern Colorado, 2022.

The machine-learning algorithms have gained popularity and have gotten the attention of many researchers in the fields of statistics and computer sciences in recent decades. Due to their computational capabilities in big data, many researchers have been attempting to incorporate machine-learning in prediction and inference problems. One of the recent methods that got a lot of attentions was referred to as the double machine learning method (DML). This method attempts to estimate the effect of the treatment variable in the presence of high-dimensional nuisance function by incorporating machine-learning algorithms. Previous studies have shown that the DML method is able to reduce the bias in estimating the targeted parameter when many covariates are present in the dataset. In this dissertation, a method was proposed that is referred to as the double super learner method (DSL). Since there are many machine-learning algorithms in existence today that are different in their searching strategy, there is no way to know which algorithm performs best for a given dataset. The proposed DSL method was developed in parallel with the DML method and works by incorporating several machine-learning algorithms via the super learner function. Numerical simulation was performed across various data settings in terms of the sample, the number of associated covariates, and the type of treatment variable. In comparison with the original DML method, numerical simulation showed that the proposed method achieved reduction in bias and provided valid confidence intervals in situations where

the original method did not. A package called DoubleSL was then developed and made public for those who desire to use this method in their research. In addition, real-data examples were included in the package to demonstrate the use of this method.

ACKNOWLEDGEMENTS

I would like to start this part by expressing my sincere gratitude to UNC for the wonderful learning and social experience that have significantly influenced me to be the person I am today. I had great memories over the years of my study, I met amazing groups of people, and learned from competent faculty members that I will always remember.

I would like to thank my professors at the Department of Statistics and Research Methods from whom I have learned a lot. My sincere gratitude to Dr. Shafie, Dr. Khaledi, Dr. Schaffer, Dr. Lalonde, and Dr. Hutchinson. I will never forget your efforts, dedication, and kindness as long as I am alive. I also would like to express my special appreciation and gratitude to my academic and research advisor, Dr. Han Yu, for his support, patience, and good advising throughout the development of this dissertation. I would like also to thank my faculty representative, Dr. Sung, for being part of my dissertation journey and for his constructive comments. I also would like to express my appreciation towards my classmate, David Agboola, who helped a lot in learning how to run my simulation using RMACC resources. Also, special thanks go to Keyleigh Gurney for her support during the time she has in the department.

I cannot pass this opportunity without thanking my family and my close friends who I missed greatly during my studying abroad. Without their love and support, I would not been able to make it. I would like to thank my father, my mother, and my siblings for being there for me when I needed them the most, and for loving me when I was difficult to love. I love you all and I hope I made you proud of me today. Special thanks go to my friends who have been connecting with me and sent me their wishes and prayers, I really hope we keep staying this close forever.

I also would like to thank the leaders of Saudi Arabia, the King and the Crown Prince for supporting so many young Saudis to pursue their dreams and education by providing generous scholarship programs across the globe. I also would like to thank my employer, the Institute of Public Administration (IPA), for supporting me through my study in the United States. I would like to express my appreciation to the Saudi Arabia Cultural Mission in United States (SACM) for their communication, facilitation, and commitment towards Saudi students.

TABLE OF CONTENTS

Chapter

I.	INTRODUCTION…….	1
	Motivation……..	5
	Purpose of Study…….	6
	Research Questions …….....	7
	Abbreviation and Terminology Definitions…….....	8
II.	REVIEW OF THE LITERATURE ……..	11
	Classical Semi-Parametric Modeling……..	11
	The Rise of Machine Learning…….....	15
	The Double Selection Methods……...	20
III.	METHODOLOGY……	25
	Double Machine Learning……...	28
	The Super Learner……...	34
	The Proposed Method: Double Super Learner…….....	38
	Improving the Computational Efficiency of the Double Super Learner Algorithm.……...	46
IV.	RESULTS……..	51
	Simulation Scheme……..	52
	Simulation Analysis for Continuous Case Treatment…….	55
	Simulation Analysis for Binary Case Treatment…….	77
	Implementing the Double Super Learner Function Using R Package……	95
	Empirical Example: Student's Performance Dataset……..	97
	Findings……...	100
V.	DISCUSSION AND CONCLUSIONS ……...	103
	Limitations and Future Studies……...	104
	Conclusion……..	105

REFERENCES…… 107

APPENDIX
A. Outliers Detection……... 112
B. Estimation of Binary Treatment Effects Using Equations (44)-(46)…….. 114
C. Empirical Example: Communities and Crimes Dataset……. 126
D. R Syntax…….. 128
E. Permission to Reproduce Figure 3……. 130

LIST OF TABLES

Table

1 The Average of Estimated Weights for the Response and the Treatment Variables across Five Machine Learnings in the Super Learner Given the Training and Testing Sets..... 47

2 The Average of Estimated Targeted Parameter for 500 Replicates..... 49

3 The Average of Estimated Weights for the Response and the Treatment Variables across Five Machine Learnings in the Super Learner Given the Training and Testing Sets..... 50

4 Settings for Sample Size and Number of Extraneous Variables under Investigation..... 55

5 Performance of Each Candidate Learner in the Super Learner Analysis for p = 20 when the Treatment Variable is Continuous..... 59

6 Summary Statistics for Datasets with p = 20 when the Treatment Variable is Continuous..... 60

7 Performance of Each Candidate Learner in the Super Learner Analysis for p = 100 when the Treatment Variable is Continuous..... 65

8 Summary Statistics for Datasets with p = 100 when the Treatment Variable is Continuous..... 66

9 Performance of Each Candidate Learner in the Super Learner Analysis when p = 1,000..... 69

10 Summary Statistics for Datasets with p = 1,000 when the Treatment Variable is Continuous..... 70

11 Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 10,000 when the Treatment Variable is Continuous..... 73

12 Summary Statistics for Datasets with p = 10,000 when the Treatment Variable is Continuous……. 74

13 Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 20 when the Treatment Variable is Binary……. 79

14 Summary Statistics for Datasets with p = 20 when the Treatment Variable is Binary……... 80

15 Performance of Each Candidate Learner in the Super earner Analysis for Datasets with p = 100 when the Treatment Variable is Binary…… 84

16 Summary Statistics for Datasets with p = 100 when the Treatment Variable is Binary…… 85

17 Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 1,000 when the Treatment Variable is Binary……. 88
.

18 Summary Statistics for Datasets with p = 1,000 when the Treatment Variable is Binary…… 89

19 Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 10,000 when the Treatment Variable is Binary…..... 91

20 Summary Statistics for Datasets with p = 10,000 when the Treatment Variable is Binary……. 92

21 Summary of Functions and Datasets Included in the DoubleSL Package……. 96

22 Summary Statistics of School Effect on Students' Final Portuguese Scores…. 99

23 Candidate Machine-Learning Algorithms' Performance in Estimating Nuisance Functions…….. 100

24 Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 20 when the Treatment Variable is Binary…..…….. 116

25 Summary Statistics for Datasets with p = 20 when the Treatment Variable is Binary…… 117

26 Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 100 when the Treatment Variable is Binary…… 120

27 Summary Statistics for Datasets with p = 100 when the Treatment Variable is Binary……..…… 121

28 Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 1,000 when the Treatment Variable is Binary……. 124

29 Summary Statistics for Datasets with p = 1,000 when the Treatment Variable is Binary…….…… 125

30 Summary Statistics about the Employment Effect on the Total Number of Violent Crimes…….. 127

31 Candidate Machine-Learning Algorithms' Performance in Estimating Nuisance Functions…….. 127

LIST OF FIGURES

Figure		
1	Directed Acyclic Graph of the Variables ……..	26
2	Comparisons of the Estimation Method for 500 Simulated Data with (n=500, p=20) ……..	32
3	Flow Diagram of the Super Learner Algorithm…….	38
4	The Distribution of ($\tilde{\theta}_0 - \theta_0$) Using the Double Super Learner Method (n=500, p=20) ……..	44
5	Comparison of the Distribution Density of ($\hat{\theta}_0 - \theta_0$) for All Methods (n=500, p=20) ……..	45
6	The Distribution of ($\tilde{\theta}_0 - \theta_0$) Using for DSL Method Using Only LASSO…….	48
7	The Distribution of ($\tilde{\theta}_0 - \theta_0$) when p = 20 when the Treatment Variable is Continuous……..……	57
8	The Distribution of ($\tilde{\theta}_0 - \theta_0$) when p = 100 when the Treatment Variable is Continuous……..	62
9	The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 100 when the Treatment Variable is Continuous after Applying the Necessary Trimming……..…	64
10	The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 1,000 when the treatment variable is continuous…….……..	67
11	The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 1,000 when the Treatment Variable is Continuous after Applying the Necessary Trimming…….	68
12	The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 10,000 when the Treatment Variable is Continuous.…….	71
13	The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 10,000 when the Treatment Variable is Continuous after Applying the Necessary Trimming……	72
14	Assessing the Bias across Different Sample Sizes as p Increases when the Treatment Variable is Continuous……	75

15 Assessing the Variance across Different Sample Sizes as p Increases when the Treatment Variable is Continuous…… 76

16 The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 20 when the Treatment Variable is Binary…… 78

17 The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 100 when the Treatment Variable is Binary……. 82

18 The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 100 when the Treatment Variable is Binary after Applying the Necessary Trimming…… 83

19 The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 1,000 when the Treatment Variable is Binary……. 87

20 The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 10,000 when the Treatment Variable is Binary……..... 90

21 Assessing the Bias of the Three Methods Across Different Sample Sizes as p Increases when the Treatment Variable is Binary…….. 93

22 Assessing the Variance of the Three Methods across Different Sample Sizes as p Increases when the Treatment Variable is Binary……... 94

23 Descriptive Statistics of School and Final Portuguese Scores Variables…….. 98

24 Outliers for Estimating the Targeted Parameter when the Treatment D is Continuous…….. 113

25 The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 20 when the Treatment Variable is Binary……..….. 115

26 The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 100 when the Treatment Variable is Binary……..….. 118

27 The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 100 when the Treatment Variable is Binary after Applying the Necessary Trimming……... 119

28 The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 1,000 when the Treatment Variable is Binary…… 122

29 The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 1,000 when the Treatment Variable is Binary after Applying the Necessary Trimming……... 123

CHAPTER I

INTRODUCTION

According to a report published, 90% of the data on the globe were generated in the last two years (Marr, 2018). This high percentage is unsurprising, given the increased use of information technology, alongside the rise of the social network that resulted in automatically generating vast and complex datasets with thousands of subjects being measured on thousands of different metrics. Data influx has led to revolutions in several fields, such as machine learning and artificial intelligence.

Learning from complex datasets can be approached from a variety of perspectives. The problem with these kinds of vast and complex datasets for data mining is analogous to trying to find a needle in a haystack. In a semi-automated data-driven method, data mining detects patterns and relationships (Grossman et al., 1999). Johnstone and Titterington (2009), on the other hand, emphasized that most statistical approaches for complex datasets tend to address the complexity from the perspective of the number of participants or cases (n) under a smaller number of attributes (p), whereas recent applications such as image analysis usually work the other way around.

Consider the following scenario as an illustrative example. Assume a researcher is interested in the effect of a student being at least bilingual on their GPA score. In addition, assume there is a large number of other predictors being included in the research such as age, race, sex, level of education, working experience, citizenship, etc. If some of these predictors

might have a confounding effect on the response on the targeted parameter, what statistical models can be incorporated in such a setting?

Many statisticians, including Bickel (1982), Newey (1994), and others, have viewed semi-parametric models as an effective approach to search through high-dimensional data while focusing on specific effects of interest to the researcher. The semi-parametric model, loosely speaking, can be defined as a hybrid form of model constructed using parametric and nonparametric components. In addition, the semi-parametric models can be employed when the targeted parameter is characterized as dimensionally finite, in the presence of infinitely dimensional nuisance function (Powell, 1994). To illustrate the difference between the parametric, nonparametric, and semi-parametric models, assume the following continuous outcome model on the finite number of predictors x:

$$y = x'\beta_0 + \varepsilon, \tag{1}$$

The estimator for this model can be viewed as parametric, nonparametric, and semi-parametric estimators depending on the restriction placed on the model. If the response y is conditioned on the regressor x, and the errors ε are normally distributed independently of the regressor x with a mean of 0, then $\hat{\beta}$ is considered as a parametric solution for the model (1). On the other hand, if the model assumes that the error terms have the density which unconditionally satisfies the following restriction:

$$E[\varepsilon \cdot x] = 0, \tag{2}$$

then the estimation of β_0 in this scenario is considered as a nonparametric solution for model (1). Finally, if the model (1) places a further restriction where the error terms are distributed conditionally such as:

$$E[\varepsilon_i \mid x_i] = 0, \tag{3}$$

then, the model in (1) would be considered as a semi-parametric model due to this particular model restriction (Powell, 1994). Examples of popular semi-parametric models are restricted moment model and the proportional hazards model viewed in Tsiatis (2006).

The advantage of using this approach is that the effect of unrelated variables can be blocked while concentrating on finding an efficient estimator for the effect of the treatment variable. Despite the fact that it took a long time to gain popularity and faced numerous challenges, semi-parametric modeling have seen several breakthroughs and advancements in recent years, allowing it to become a very efficient method. The efficiency comes into play when attempting to obtain an accurate estimate for the treatment of interest that allows for causal inference, which has become a hot topic in statistics and data analysis in recent years due to the large number of applications that can be used.

Methods have been developed based on the concept of semi-parametric models in such a way that they account for the confounders effect, or controls, that have been observed in the resultant data sets via the variable selection process. Urminsky et al. (2019) and Chernozhukov et al. (2018), for example, focused on using Robinson's semi-parametric model and applying a double selection for those variables that affect both the response and the treatment effect of interest. Modern estimation methods that can handle a large number of collected controls are required for these double-stage selection techniques.

In recent years, machine learning algorithms have emerged as a very powerful tool for handling large and complex datasets for both prediction and inference. Classic statistical approaches fail to achieve meaningful estimates for the effect of the treatment of interest in the presence of a large number of controls, according to Chernozhukov et al. (2017) and Chernozhukov et al. (2018), especially when the number of variables in the data set exceeds the

number of collected cases (i.e., when $n < p$). This is a common issue in many situations, such as when scientists are studying the genomes of a small number of patients in biomedical studies, where the researcher may be dealing with thousands of parameters (Wang et al., 2020). Another evident example of this type of dataset arises from image analysis, where the number of associated variables can exceed hundreds, such as variables concerns with wavelength, mass, polarization, electron energy, and so on (Wise & Geladi, 2000).

There is a major difference between statistical and machine learning. In statistical learning, a causal inference can be drawn about a treatment of interest from a relatively small sample, but machine learning is able to reliably predict from more complex data using relatively large and complex datasets (Bzdok et al., 2018). By developing sophisticated estimation methods, such as those proposed by Chernozhukov et al. (2017) and Chernozhukov et al. (2018), machine learning algorithms were able to provide an unbiased estimator for the targeted parameter that outperformed traditional statistical methods when a large number of covariates are present during the data collection process. In this dissertation, I proposed an estimation method for the targeted parameter in the presence of high dimensional nuisance function, which I called the double super learner (DSL) method. The goal from this method, the DSL, was to improve estimation by achieving a reduction in the bias resulting from estimating the targeted parameter. Furthermore, it is critical to comprehend how this method compares to the existing double or debiased machine learning (DML) method. This DSL method as proposed in the methodology section is used in the context of semi-parametric modeling such as the partial linear model introduced by Robinson (1988), and it employs the orthogonalization technique such as the one introduced by Neyman (1959). Examples also include the super learner algorithm introduced by Van der Laan et al. (2007), and the DML introduced by Chernozhukov et al. (2017) and

Chernozhukov et al. (2018), while taking advantage of sample splitting, cross-fitting, and cross-validation techniques during the analysis.

Motivation

A semi-parametric model can provide good estimates of treatment effects because it combines the advantages of parametric and nonparametric models. With the explosion in the number and size of datasets available today, it is more important than ever to employ a new technique that can keep up with the data's complexity. Aside from that, the rise of machine learning algorithms and the constant development in recent years in advancing more learner algorithms has opened a door for researchers to make more discoveries in the field. In recent years, the potentials for employing machine learning algorithms have greatly increased due to the improved computing capabilities, such that computers nowadays can handle larger data and perform very intensive computational procedures more efficiently than they could 30, 20, or even 2 years ago.

The ordinary least squares (OLS) estimation method has dominated other methods in regression applications for decades, thanks to its strong theoretical foundations and obvious inferential advantages. The OLS estimation method for regression and classification problems, however, has its own set of drawbacks. Multicollinearity among the independent variables is common in real data, such as economic and medical datasets, which makes calculating $(X^TX)^{-1}$ required for OLS estimation nearly impossible, resulting in very unstable solutions (O'Driscoll & Ramirez, 2016). The benefit of advanced machine learning can be seen in terms of model constraints. In contrast to linear regression, ridge regression does not require the full rank of the design matrix, nor the distributional or independence assumptions that are common in OLS models. These more relaxed conditions can help ridge regression handle datasets that are more

complex, which are more like real-world data as in situations where the number of confounders affecting both the response and the predictors of interest increase the dataset's dimensionality.

Given the undeniable importance of statistical inference in drawing causal conclusions and quantifying uncertainty in high-dimensional settings, using machine-learning algorithms in double selection methods such as the DML has shown to be very promising in terms of estimation and causal inference (Chernozhukov et al., 2018). Although there is limited literature on this concept due to its novelty, the DML method proposed by Chernozhukov et al. (2017) and Chernozhukov et al. (2018) has shown great promise in a wide range of applications, overcoming many of the issues faced by classical methods that seek to make casual inference on the treatment of interest. When compared to traditional semi-parametric method approaches, what makes DML such a powerful tool is that there is no need to set a number of regularity conditions, allowing it to be applied to a wide range of complex datasets.

Purpose of Study

It is hard to not to argue that the DML method Chernozhukov et al. (2017) is considered quite revolutionary when it comes to obtaining unbiased estimates in the presence of high dimensional nuisance parameters that this dissertation attempts to build on his work. The method proposed in this dissertation attempts to integrate the DML concept with a powerful machine-learning algorithm known as the super learner introduced by Van der Laan et al. (2007). Since the super learner algorithm ensembles a number of machine-learning algorithms and present their predictions as a linear model to predict the response, this strategy has proven to reduce prediction cross-validation risk (Van der Laan et al., 2007). Some researchers, when working on prediction problems, might find themselves arguing about which machine-learning algorithm to incorporate into the analysis where personal favorites toward learners might affect their decision. The super

learner can be a solution to this dilemma since it can incorporate all the machine-learning algorithms at once and results in lowering of the prediction cross-validation risk more than any other machine-learning algorithm that is implemented separately.

The method proposed in this dissertation, referred to as the double super learner (DSL), will integrate the DML method with the super learner algorithm. Because the DML method can perform only one machine-learning algorithm at a time, the researcher might find himself risking being biased towards which machine learning algorithm to incorporate in the analysis for estimating the targeted parameter in the DML context. To minimize the impact of the researcher's personal favorites among machine-learning algorithms, the purpose of this dissertation was to provide a modern framework via the inclusion of a number of machine learning algorithms in a single estimation procedure. In particular, the goal of this dissertation was to propose an estimation method in the context of semi-parametric modeling for the targeted parameter in the presence of high dimensional nuisance function with the aim of achieving an improvement in the resultant estimates of the targeted parameter when it comes to the resultant bias in comparison with the original DML method developed by Chernozhukov et al. (2017) and Chernozhukov et al. (2018)

Research Questions

Throughout this dissertation, I sought to investigate the following research questions in order to assess the DSL method.

Q1 How can the estimator of the targeted parameter in presence of high-dimensional nuisance functions be constructed using the DSL method, given the conceptual differences between the DML method and the super learner algorithm that the DSL method is trying to integrate?

Q2 How will the DSL perform in terms of bias reduction for the estimated targeted parameter in the presence of high-dimensional nuisance functions in comparison with the original DML method, and whether the respective confidence intervals

contain the true values of the targeted parameter under varying number of predictors and sample sizes?

Q3 What criteria can be used to select the best machine learning algorithm among those incorporated in the DSL method in order to improve computational efficiency, and how does the reduction of candidate learners impact the estimation bias in comparison with the DML method and the DSL method when considering all candidate learners.

Q4 How can the algorithm of the proposed DSL method be developed using R, and what settings needed for implementation so the estimates of the targeted parameter and its associated confidence interval can be numerically calculated?

Abbreviation and Terminology Definitions

Complex settings are used to describe the complex structure of a dataset and the relationships between the features found in the data. The complex settings referred to in this dissertation are the types of data structural complexity that results from many covariates, some of which have confounding effects both on the treatment and the response variables, where ignoring them would lead to poor and biased estimation.

Double machine learning (DML) is a statistical machine learning method developed by Chernozhukov el at. (2017) and Chernozhukov et al. (2018). In the method, an estimator for the treatment, policy, and effect was constructed using the semi-parametric approach with Neyman (1959) orthogonality and sample splitting using machine learning algorithms.

Double super learner (DSL) is the new method proposed in this dissertation that aims to construct an estimator for the targeted parameter in the presence of high-dimensional setting. The double super learner method is the result of integrating the DML with the super learner (SL) concepts.

High-dimensional data are produced when the number of features, or variables, collected in the dataset is extremely large. In high-dimensional data, the observed features may outnumber the observations (i.e., $p > n$).

Machine learning (ML) is a type of nonparametric components are estimated with the new generation of nonparametric statistical methods, branded as "machine learning" methods. ML is efficient artificial intelligence computation that consists of data-driven algorithms built on statistical foundations that train data in order to obtain predictions that mimic human decisions. Examples of machine learning algorithms will be discussed throughout this dissertation, including lasso, random forests, and boosting.

Nuisance functions are functions that express the effect of independent variables that are not of interest but must be accounted for so that inference can be made about the targeted parameter in a specific context. These variables in the datasets can be tagged as irrelevant variables and confounders, or controls, which are variables that have some sort of relationship with both the independent variable of interest and the response.

Ordinary least squares (OLS) estimation is a statistical method that can be used in a variety of applications when $p < n$, including simple and multiple regression, to estimate the relationship between the response variable and the predictors by minimizing the sum of squares based on the difference between the response variable and its predicted values.

Semi-parametric models are models with a parametric and nonparametric component. The parametric component is finite dimensional and nonparametric component is infinite dimensional. The term nonparametric is typically reserved for models with only infinite-dimensional components or for statistical procedures that do not require knowledge of underlying distributions. In semiparametric models, the parametric part is for scientific

interpretability, while the nonparametric part is for flexibility. A main question for semiparametric models is how to conduct efficient inference which requires semiparametric inference. The non-parametric portion is made up of variables that are not necessarily of interest but must be taken into account. The partially linear regression model (PLR) introduced by Robinson (1988), which will be discussed later in this dissertation, is one of the most popular semi-parametric models.

Super learner (SL), developed by Van der Laan et al. (2007), is a machine learning algorithm that can be employed for both regression and classification cases. A great advantage of this algorithm is that it can be constructed using a variety of machine learning algorithms that the researcher can choose. The methodology section of this dissertation will go into greater detail about the construction of the super learner algorithm.

CHAPTER II

REVIEW OF THE LITERATURE

Semi-parametric models can be simply defined as models that include both a parametric part, which is a parameterization of the treatment effect, and a non-parametric part, which can be characterized by a nuisance function for the irrelevant covariates and confounders present in the dataset (Tsiatis, 2006). Given the enormous potentials of this approach, several researchers have set out to estimate a parameter of interest in the presence of a high-dimensional (nuisance) parameter.

Classical Semi-Parametric Modeling

When the dimension of the parameter space is large, semi-parametric modeling has become a hot topic that has appeared regularly in various types of literature that concerns understanding the influence of specific treatments or policies. The misspecification of the nuisance function would lead the estimation of the targeted parameter to be inconsistent (Robinson, 1988). Many early statisticians focused on addressing this issue by defining sets of regulatory conditions and introducing frameworks of adaptation or orthogonalization in order to construct sophisticated estimators that are root-N consistent and asymptotically normal, allowing for valid and causal inference conclusions. These attempts can be seen in publications such as those of Bickel (1982), Robinson (1988), Andrews (1994), Newey (1994), and van der Vaart (1998).

Take, for example, Bickel (1982), who provided a series of empirical results on developing adaptive estimators, drawing on the work of Charles Stein. Stein's embodiment of semi-parametric modeling sense started when he wondered how one might estimate the parameter of interest in Euclidean distance in his case asymptotically in the presence of an unknown nuisance shaped parameter, G, as if G were known (Stein, 1956). Stein introduced a condition that holds when the true parameter is regular and the estimator is adaptive. Bickel (1982) considered Stein's concept of adaptable estimators as problematic since it is difficult to mathematically verify as well as the lack of a clear approach on how to design an adaptive estimator based on these concepts.

In semi-parametric modeling, the adaptation concept can be described as whether a parameter of interest, such as the treatment effect, can be approximated without knowing the true nuisance function. Take the mean of a normal population (μ) as an example to understand adaptation, where the unbiased estimator ($\bar{x}$) is called an adaptable estimator because we do not need to know the actual population variance in order to generate a reasonable estimation for the mean.

Bickel (1982) introduced a set of regularity conditions for the estimator of interest, including the existence of a Fisher information matrix, the differentiability of the log likelihood function and square root likelihood, and root-N consistency. Despite the fact that these regularity conditions paved the way for adaptable estimators, they still had severe drawbacks, such as in cases where adaptation is impossible similar to the case found in the Neyman-Scott example (Schmetterer, 1960), where the estimation of the score functions was inconsistent. They also ignore the case where the Fisher information matrix is singular, despite the importance of obtaining the inverse of the Fisher information matrix for determining the limiting distribution

within which the adaptive estimator is meant to be confined. Another limitation of these adaptive estimators which was highlighted by Bickel (1982) is that they lack natural invariance properties due to the use of discretization in constructing the adaptive estimators.

Because the goal of this dissertation was to estimate a single treatment effect in the presence of many confounders, I also considered Robinson's (1988) partial least squares model, which is a semi-parametric model and can be represented as the following set of structural equations:

$$Y = D\theta_0 + g_0(\boldsymbol{X}) + U, \quad E[U|\boldsymbol{X}, D] = 0, \tag{4}$$

$$D = m_0(\boldsymbol{X}) + V, \quad E[V|\boldsymbol{X}] = 0, \tag{5}$$

where,

Y represents the response (outcome) variable,

D represents the treatment variable of interest,

X represents the confounders such as $\mathbf{X} = \{x_1, x_2, \ldots, x_p\}$ where p represents the number of extraneous variables to be included in the model,

U represents the model error in (4),

V represents the model error in (5),

θ_0 represents the treatment effect which is the targeted parameter,

$g_0(\mathbf{X})$ represents the effect of the confounders of **X**, where **X** influences the outcome variable Y throughout the function g_0,

$m_0(\mathbf{X})$ represents the effect of the confounders of **X**, where **X** influences the treatment variable D throughout the function m_0.

Using (4) and (5), Robinson (1988) was able to show, with the assumption of other 10 regularity conditions to be met, the condition that the following matrix:

$$\boldsymbol{\Phi} \equiv E[\{D - E(D|\boldsymbol{X})\}\{D - E(D|\boldsymbol{X})\}'], \quad (6)$$

is being a positive definite is sufficient enough to conclude that the estimator $\hat{\theta}_0$ is root-N consistent and that $\sqrt{N}\left(\hat{\theta}_0 - \theta_0\right)$ converges in distribution to normal with 0 mean and some variance.

The lack of the invariance property problem raised by Bickel (1982) has prompted academics to go back and use data from previous studies to address the issue. Andrews (1994), Newey (1994), and a host of others have improved on Robinson's (1988) work by combining the Donsker requirements (Donsker, 1951) with Neyman's orthogonality scores (Neyman, 1959) in an attempt to constrain the space of the nuisance function.

The Donsker theorem, named for American mathematician Monroe D. Donsker, extends the well-known central limit theorem (CLT) approach to a class of functions as an index set to tackle the empirical distribution's invariance property. To simplify, let's say X_1, X_2, …, X_n ,… are identically and independently distributed random variables with some mean (m) and variance (v). Furthermore, let $\bar{X}$ be the sample mean of sample size (n) which is an estimate of the true population mean (m). Then it can be shown that the scaled estimation error of $\sqrt{n}(\bar{X} - m)$ on the class of functions weakly converges to a Gaussian process. As a result, a confidence intervals can be constructed using only information about the mean and the variance, and a valid causal inference can be drawn. For Neyman orthogonality, on the other hand, it is highly advantageous to use the orthogonalization idea in semi-parametric models, where orthogonalization was demonstrated to play a substantial role in eliminating estimation bias in the partial least squares model (Robinson, 1988). The idea of Neyman orthogonality is to introduce a set of orthogonal moment conditions such that the treatment effect can be estimated without being affected by the nuisance non-parametric function.

Andrews (1994) developed a comprehensive framework for a large number of alternative estimators that are constructed to be root-N consistent and asymptotically normal. Andrews' estimators were all in the form of MINimizing a criterion function that may be based on a Preliminary Infinite dimensional Nuisance parameter estimator (MINPIN). These estimators have been proved to be extremely efficient under regularity assumptions, including consistency, Neyman orthogonality, and stochastic equicontinuity as well as the weak law of large numbers and the well-known CLT. Andrews demonstrated in his article that the distribution of the scaled estimating error, $\sqrt{N}(\hat{\theta}_0 - \theta_0)$, will follow the normal distribution with 0 mean and some variance v under a certain set of regularity conditions. Although this method has its own limits, it was thought to be a very beneficial option in order to address the issue of convergence rate. The primary ones that Andrews (1994) has expressed explicitly are that not all estimators can be derived without additional conditions, such as smoothness condition, and that non-parametric function estimators have more trimming limits. Furthermore, even if the root-N consistency and asymptotic normality assumptions are met, finding the estimator may not be appropriate in some scenarios. Newey (1994) and Van der Vaart (1998) developed comparable work with similar challenges, where primitive regularity conditions were incorporated to obtain estimators that were proved to be root-N consistent and asymptotically normal.

The Rise of Machine Learning

Many advancements have taken place in the last decade as a result of major developments in statistical software in a brand-new class of nonparametric estimating approaches known as machine learning. Unlike statistical learning that arose from the field of statistics, machine learning estimation is derived from the field of artificial intelligence (Mitchell, 1997). Machine learning estimators like LASSO, Random Forest, and Boosting,

among others, offer the advantage of being more flexible and capable of handling complex situations like highly dimensional data (as cited in James et al., 2013).

A comparison between OLS regression and Ridge regression can be very helpful in illustrating the distinctions between classical statistical methods and machine learning methods. Remember the OLS regression model as follows:

$$Y = \boldsymbol{X}\beta + \epsilon, \tag{7}$$

where the error terms of ϵ are assumed to be normally distributed, have a constant variance, and are not being correlated with each other. This is also assuming that the design matrix, **X**, is full rank matrix. Given these assumptions, the OLS method works by obtaining the coefficient solutions that minimize the RSS such as:

$$RSS = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2, \tag{8}$$

which leads the OLS solution for the estimated parameters to be as the following:

$$\hat{\beta}_{OLS} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'Y, \tag{9}$$

The problem with the preceding OLS approach is that multicollinearity exists in real data, such as in the economic and medical domains, where two or more predictors are correlated with each other. This issue of multicollinearity among the predictors will make the computation of $(\boldsymbol{X}'\boldsymbol{X})^{-1}$ difficult, causing the OLS solution, $\hat{\beta}_{OLS}$, to be inconsistent, resulting in an increase in the variance of these estimators (Yu et al., 2015).

Hoerl and Kennard (1970) proposed what is now known as the ridge regression concept, in which some bias is accepted as a trade-off in exchange for a large reduction in the estimators' variance. Compared to the OLS estimators provided earlier, the ridge regression estimators are more stable due to this bias-variance trade-off. Although various model selection strategies, such as backward and stepwise selection, can be used in the traditional OLS selection to cope with

collinearity and eliminate irrelevant predictors from the model to enhance the model fit, these techniques are computationally intensive. When the dimension of the design matrix is relatively large (i.e., when the number of observed predictors is significant), overfitting bias occurs, which increases the variance of the coefficient estimates (James et al., 2013).

Unlike traditional methods, which utilize least squares to fit a small number of predictors, ridge regression uses a shrinkage technique to fit a larger number of predictors. The ridge regression's goal is to reduce variance by including all predictors in the model and then shrinking their effect towards 0 (James et al., 2013). What makes ridge regression estimates differ from the OLS regression estimates is that the ridge regression has a penalty term that helps to shrink the influence of the estimates. The ridge regression estimates, to be more explicit, are the values that minimize

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \; = RSS + \lambda\sum_{j=1}^{p}\beta_j^2 \,, \tag{10}$$

The first part of the minimization is the same as the OLS regression for the estimated effect, while the second part is related to the penalty term. The penalized term has a tuning parameter, $\lambda \geq 0$ that can be selected by introducing a grid of values and using cross-validation to find the best value. As a result, the estimated parameters for the ridge regression predictors will be as follows:

$$\hat{\beta}_{Ridge} = (\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I_n})^{-1}\boldsymbol{X}'Y, \tag{11}$$

where I_n represents the identity matrix. Ridge regression estimates have been shown to have a smaller mean square error than OLS regression estimates; for example, $MSE\left(\hat{\beta}_{Ridge}\right) \leq MSE\left(\hat{\beta}_{OLS}\right)$, for some value of λ (Hoerl & Kennard, 1970).

A recent form of regularized estimation known as LASSO, introduced by Tibshirani (1996), is another machine learning method. The LASSO is constructed in a similar way to the ridge regression, using in L1 penalty rather in L2 penalty such as:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \; = RSS + \lambda\sum_{j=1}^{p}\left|\beta_j\right|, \tag{12}$$

LASSO has an advantage over ridge regression using L2 penalty in that, unlike ridge regression, which decreases the estimated parameters to values towards to 0, LASSO can reduce the estimated effect to the exact 0, improving interpretability and prediction accuracy at the same time (James et al., 2013). With that in mind, it's worth noting that the LASSO conducts both shrinkage and model selection approaches, which is why Tibshirani coined the acronym LASSO (Least Absolute Shrinkage and Selection Operator).

Machine learning algorithms diverge from statistical methods in their development, such as decision trees algorithms, which many believe are a better reflection of human decision-making (James et al., 2013). The advantage of decision trees is that they may be easily explained and illustrated for audiences without a statistical background, as well as can be graphically displayed. The main principle behind decision trees is to divide the predictor space into multiple distinct regions, then make the same prediction for responses with predictors in the same region. The leaves, or terminal nodes, are the regions that come from the decision tree analysis, whereas the internal nodes are the decision-breaking points. The resulted decision tree would eventually look just like a normal tree that is drawn upside down, where the leaves are shown in the bottom of the resulted tree. The optimal decision is being made by finding the regions that minimize the RSS, which is given by:

$$\sum_{j=1}^{J}\sum_{i\epsilon R_j}\left(y_i - \hat{y}_{R_j}\right)^2, \tag{13}$$

where $\hat{y}_{R_j}$ is the mean response for the training data within the j[th] region. The decision trees have long been used in a wide range of industrial applications (Mittal et al., 2017), with Wald using them in game theory in 1950.

Many researchers have built on the concept of decision tree, such as Mansour (1997), who introduced the concept of tree pruning, which involved removing part of the tree's leaves to enhance prediction accuracy since a tree with many leaves can overfit the training data. While pruning a fully developed tree can improve prediction accuracy for fresh datasets, it often comes at the expense of training dataset accuracy (Ho, 1995). Ho (1995) was inspired by this apparent constraint of pruning to take a different method, mixing numerous small trees instead of pruning them, resulting in random decision forests as we know them today.

Many machine-learning algorithms have been developed over the years as a result of the advancements in information technology and statistical computing. This prompted statisticians like Van der Laan to develop what is now known as the super leaner (Van der Laan et al., 2007). The super learner (SL) is an ensemble machine learning method that uses a linear model to weight a number of machine learning algorithms based on their cross-validation risk. In his study, Van der Laan et al. (2007) demonstrated that SL asymptotic performance is on par with, if not better than, any other machine learning algorithm. The SL generates a prediction that uses the probabilistic weights of each machine learning method as model parameters, and the predictors in the model are the predicted response values for weighted parameters. As a result, the SL has the following structure:

$$y_i = \beta_1 z_{1i} + \cdots + \beta_k z_{ki} + \epsilon_i, \tag{14}$$

where

y_i represents the response of the i[th] case,

β_k represents the weight of the k^{th} machine learning algorithm,

z_{ki} represents the predicted value of the i^{th} case using k^{th} machine learning algorithm,

ϵ_i represents the error of predicting the i^{th} case.

The Double Selection Methods

Previous research on the causal inference of the average treatment effect has prompted researchers like Robins et al. (1994) to develop methods for estimating the treatment of interest when some of the regressors are not seen when using the inverse probability weighted (IPW) approach. He proposed a method for obtaining consistent estimators when some of the data are missing at random, using missingness probabilities that are known or can be represented in a parametric sense. This work was later adopted by Funk et al. (2011) to develop what is now known as the doubly robust (DR) estimators in which the researchers attempted to find a framework to account for the relationship between confounders and response in subjects who received treatment and those who did not separately. In their study, Funk et al. (2011) combined the propensity scores (PS) model with the outcome regression model for the response given the confounders and the treatment of interest to represent the link between the treatment variable and the confounders, such as PS = E[Treated| Observed Confounders]. Funk et al. (2011) claimed that correct specification of at least one of these models would be sufficient to obtain an unbiased estimator for the average treatment effect using the DR estimation approach.

To illustrate the method proposed by Funk et al. (2011), we can assume that the observed data are structured as (Y, D, $\boldsymbol{X}$), where Y represents the collected response, D represents the treatment in a study, which is a binary variable with values of 1 and 0, and $\boldsymbol{X}$ represents a set of confounders collected in the dataset. As a result, we get PS = E[D=1| $\boldsymbol{X}$]. The logistic and probit models are typical methods for calculating PS estimations. The following model can be used:

$$Logit[P(D = 1|\,\boldsymbol{X})] = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k, \quad (15)$$

On the other hand, the outcome regression model might be used to explain the model that specifies the link between the response and the treatment as well as the confounders. This model can be described as follows:

$$E(Y|\,D, \boldsymbol{X}) = \alpha_0 + \alpha_1 D + \alpha_2 X_1 + \cdots + \alpha_{k+1} X_{k,} \quad (16)$$

where α_1 is targeted parameter the DR method is seeking to estimate, which takes the interpretation of the average treatment effect (ATE).

The DR method is used to estimates the PS and outcome mean separately for individuals who received treatment in one group and those who did not in the other, utilizing both treatment groups for the ATE. As a result, the DR estimators would have the following two equations.

$$DR_1 = \frac{Y_{(D=1)}D}{\widehat{PS}} - \frac{\hat{Y}_{(D=1)}(D-\widehat{PS})}{\widehat{PS}}, \quad (17)$$

$$DR_0 = \frac{Y_{(D=0)}(1-D)}{1-\widehat{PS}} + \frac{\hat{Y}_{(D=0)}(D-\widehat{PS})}{1-\widehat{PS}}, \quad (18)$$

After calculating the DR estimator for those who received the treatment, DR_1, and those who didn't, DR_0, the average for the entire population would be implemented and the difference between those estimators would describe the ATE such as:

$$\hat{\alpha}_{DR} = N^{-1}\left(\frac{Y_{i\,(D=1)}D_i}{\widehat{PS}_i} - \frac{\hat{Y}_{i\,(D=1)}(D_i-\widehat{PS}_i)}{\widehat{PS}_i}\right) - N^{-1}\left(\frac{Y_{i\,(D=0)}(1-D_i)}{1-\widehat{PS}_i} + \frac{\hat{Y}_{i\,(D=0)}(D_i-\widehat{PS}_i)}{1-\widehat{PS}_i}\right) \quad (19)$$

Using the DR estimator obtained in (19), Funk et al.'s (2011) argument was that the correct specification for at least one model would result in an unbiased estimator. To illustrate Funk et al.'s argument briefly, take the first part of (19) when the treatment is received, i.e., when D = 1. This first part of the model consists of two smaller components: the first of which represents the average response for those who received the treatment; the second component is referred to as the augmentation part. This augmentation part constructed by multiplying two

biased terms: one results from the outcome model (16), and the second results from the PS model in (15), respectively. If at least one of these models is correctly specified, then a bias of 0 or close to it would cancel the other by multiplication process. For more details, please refer to Appendix A (Funk et al., 2011). Similar work being conducted by Luedtke et al. (2017), and Li and Shen (2019).

Belloni et al. (2014) introduced the post double selection approach, which is another prominent double stage selection method. Extension of this concept can be also found in studies by Urminsky et al. (2019) and Wang (2020). The key to this method is to use LASSO regression to perform model selection for confounders in both sub-models. The idea behind this method is that removing confounders that are irrelevant to the response but have significant effect in the case of the treatment variable (i.e., the average treatment effect) can lead to a decrease in the bias of the omitted variables. Assume we have data for the structure (*Y, D,* $\boldsymbol{X}$), where *Y* and *D* represent the response and treatment, respectively, and $\boldsymbol{X}$ represents the confounders discovered throughout the data collection process. As a result, there are three essential steps to implementing the post double selection method. The first step is to fit the response to the data using the associated covariates including the confounders such as:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_k X_{ik} + \epsilon_i \tag{20}$$

using LASSO regression and keeping track of confounders in predicting the response *Y*. The second step in the post double selection method is to fit the confounders that predict the independent variable of interest in the same way as the first step implemented earlier:

$$D_i = \alpha_0 + \alpha_1 X_{i1} + \cdots + \alpha_k X_{ik} + \epsilon_i \tag{21}$$

Just as before, it is important to maintain tracking of those confounders that have been shown to be important in predicting the independent variable *D*. The final stage is to create a set

named *G* that includes every confounder that showed significance in predicting either *Y* or *D*. As a result, the following is how the final prediction model should be fitted:

$$Y_i = \delta_0 + \delta_1 D_i + \sum_{k \in G} \delta_k X_{ik} + \epsilon_i \quad (22)$$

After obtaining the estimated effect for the variable of interest, $\hat{\delta}_1$, simulation and empirical examples have shown that the resulting estimator is unbiased, root-N consistent, and asymptotically normal around 0 when the scaled estimation error is considered; for example, $\sqrt{N}(\hat{\delta}_1 - \delta_1)$. This method made machine learning, such as LASSO, a valid tool for making a causal inference as well.

Advanced machine learning algorithms and sample splitting techniques have paved the way for further development in constructing efficient and root-N consistent estimators. The concept of DML, in which a new class estimator was developed to make an inference about a low dimensional parameter of interest in the presence of a high-dimensional nuisance function, is one of the most popular concepts that has been attracting a lot of attention. (Chernozhukov et al., 2018).

In this novel approach, Chernozhukov et al. (2018) looked at a variety of sophisticated machine learning algorithms for estimating the ATE, in the presence of a high-dimensional nuisance function, and found that they all produced consistent results that were root-N consistent. The Frisch-Waugh-Lovell style (Frisch & Waugh, 1933) was incorporated in this method, which used a two-step estimation procedure. The Neyman orthogonality condition, proposed by Neyman (1959), guarantees a very low bias resulted from employing regularization at the expense of a significant reduction of variance, while sample splitting and cross fitting are used to improve the efficiency and stability of the resulting estimator (Chernozhukov et al., 2018). When compared to classical semi-parametric methods, which have limitations when the

number of predictors is large, this DML method has proved to produce a clean and reliable estimator with fewer restrictions and regularity conditions. Another advantage of using the DML method is that one can construct a confidence interval so that inference can be meaningful using a variety of semi-parametric models, including the partially linear model introduced earlier, (4) and (5), and the partially linear instrumental variables model, since the resulted estimator is root-N consistent. Other related literature worth mentioning on the DML can be found in studies by Knaus (2021), Yang et al. (2020), and Bach et al. (2021).

CHAPTER III

METHODOLOGY

In this chapter of the dissertation, I will be proposing an estimation method that is designed to handle high-dimensional and complex data. The goal from using this method was to be able to make causal inference on the targeted parameter in the presence of high-dimensional nuisance function. Assume that each observation has the following data structure:

$$O_i = (Y_i, D_i, \boldsymbol{X}_i),\ i = 1, \dots, N,$$

where,

Y_i represents the observed response for the i^{th} case,

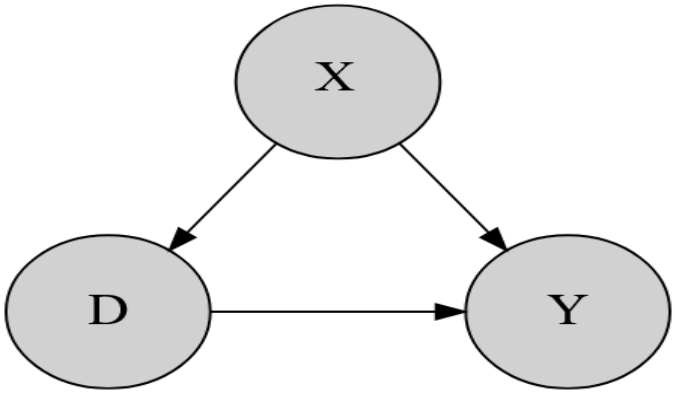
D_i represents the observed treatment or independent variable for the i^{th} case,

$\boldsymbol{X}_i$ represents a matrix of all the observed confounders for the i^{th} case.

The term high-dimensional refers to the dataset contains a large number of variables, including treatment variable, alongside many confounders. On the other hand, the complexity emphasizes the interrelationships between these variables. The presence of confounders, or controls, usually indicates the presence of some kind of relationship among them and between confounders the response and treatment variable of interest. When such confounders are present, accounting for them is critical for making meaningful inferences about the treatment of independent variable of interest (Skelly et al., 2012). Figure 1 is a directed acyclic graph (DAG) that depicts the relationship that confounders have with the response and the treatment or independent variable of interest.

Figure 1

Directed Acyclic Graph of the Variables



Typically, datasets with this kind of relationship between the variables presented in Figure 1 with high-dimensional **X** poses a number of difficulties. Some examples include, but are not limited to, the following:

- The curse of dimensionality: while it may appear that having a large number of features in a dataset reduces uncertainty and improves prediction and inference, modeling a large number of features that aren't proven to be relevant to the response can result in a significant decrease in prediction power due to an increase in the test error, besides can lead to the issue of over-fitting (James et al., 2013).
- The multicollinearity issue: this issue can arise when the predictors have a high degree of correlation among each other. Multicollinearity can cause a number of issues, including standard error inflation, which reduces the statistical power of fitted model, making it difficult to understand the role and contribution of each predictor in explaining the response (Yu et al., 2015).
- The presence of missing values: because this type of data is typically collected over a large number of cases and measured across a large number of features, the likelihood

> of missing values is extremely high. This issue can be problematic in a number of ways, including a reduction in statistical power and bias in the resulting estimates, both of which can jeopardize the validity of any conclusions drawn (Kang, 2013).

Many studies, such as those of Bickel (1982), Robinson (1988), Andrews (1994), Newey (1994), Robins et al. (1994), to name a few, have attempted to model this type of data using the classic semi-parametric modeling approach, as shown in equations (4) and (5). All of these attempts have aimed to create unbiased estimators that are root-N consistent, allowing for valid causal inference. These attempts all have one thing in common: they are constructed by introducing large sets of regularity conditions to constrain the space of the nuisance function that explains the dataset's confounders. Although, the recent double machine learning method proposed by Chernozhukov et al. (2018) has shown great promises in the sense of being simple and easy to implement as well as in the ability to produce consistent estimators without the need for setting the number of complex regularity conditions.

The method in this research is incorporating the concept of double machine learning introduced by Chernozhukov et al. (2018), using the super learner algorithm of Van der Laan et al. (2007). It aims to obtain an efficient and robust root-N consistent estimator using a number of ML algorithms, which is referred to as the DSL. For the exposition of the proposed method, this section is presented in five sections. In order to set the stage for the proposed method, the first two subsections will cover the DML and the SL algorithm. The proposed method, the DSL, will be described in the third subsection. Simple simulated results will be presented in the fourth subsection of this section of the paper to demonstrate the potential of the proposed method in obtaining an accurate estimator for the effect for the independent variable of interest that are unbiased and root-N consistent. The last subsection will investigate an attempt to improve the

computational efficiency by looking into the analysis of the weighted effects of the incorporated ML algorithms for predicting the response and the causal variable in the presence of confounder matrix.

Double Machine Learning

To demonstrate the advantages the double machine learning has over classical methods, the steps outlined by Chernozhukov et al. (2018) will be followed, beginning with what is called the naive approach. The partial linear model introduced by Robinson (1988), which is made up of two parts: parametric and nonparametric, as presented earlier in the literature review chapter as model (4). It is preferable to begin with the regular, naive, machine learning approach to obtain an estimate for the targeted parameter, denoted as $\hat{\theta}_0$. The data were divided into two halves in this approach: the first half represents the testing set (T) of size n, and the second half represents the training set (Tr). The nuisance function, g_0, is estimated using the training set based on the split in the overall data and retain $\hat{g}_0$. After that, $\hat{g}_0$ can be plugged into the PLR model equation (4). Finally, the naive estimator, denoted earlier as $\hat{\theta}_0$, can be constructed using the testing set as the following:

$$\hat{\theta}_0 = \left(\frac{1}{n}\sum_{i\epsilon\, T} D_i^2\right)^{-1} \frac{1}{n}\sum_{i\epsilon\, T} D_i\left(Y_i - \hat{g}_0(\boldsymbol{X}_i)\right), \tag{23}$$

Although the above resulted estimator in (23) may seem quite sophisticated in the way it's constructed, it has an obvious bias. The expectation $E\left(\hat{\theta}_0 - \theta_0\right)$ can be decomposed into two terms, as shown in the study by Chernozhukov et al. (2018). The term "regularization bias" has issues, and it is expected to have a form of:

$$E\left(m_0(\boldsymbol{X}_i)\left(g_0(\boldsymbol{X}_i) - \hat{g}_0(\boldsymbol{X}_i)\right)\right), \tag{24}$$

The above term cannot diminish to 0, since $m_0(\boldsymbol{X}_i)$ keeps track of the effect of confounders on the variable D via the function m_0, which is not centered around 0 (i.e., $m_0(\boldsymbol{X}_i) \neq 0$). If the second model, (5), is ignored when constructing the estimator $\hat{\theta}_0$, the resulting estimator for the effect of the treatment variable of interest $\hat{\theta}_0$ will be biased. The rate of convergence of the scaled estimation error would go to infinity due to this neglection: $\sqrt{N}(\hat{\theta}_0 - \theta_0) \rightarrow \infty$. Since the ultimate goal of incorporating ML into a semi-parametric model is to be able to construct valid confidence intervals that allow for causal inference, the resultant estimator should satisfy the root-N consistency criteria. This is in order to achieve the required rate of convergence, which the naive ML estimator approach, introduced earlier, fails to do.

A brand-new statistical concept called DML was developed to overcome those two key issues in the field of semi-parametric modeling (Chernozhukov et al., 2018). The construction of the DML estimator is based on two key strategies: the first is the use of orthogonalization to overcome regularization bias; the second crucial step is to use sample splitting to achieve the required rate of convergence.

The orthogonalization introduced by Neyman (1959) plays an important role in producing debiased estimates for the effect of the independent variable, or the treatment, using the Frisch-Waugh-Lovell style (Frisch & Waugh, 1933) for the first key construction of the DML estimator. Consider the second construction in equation (5). The effect of the confounders, which are presented throughout the function $m_0(\boldsymbol{X})$, is partially removed from the treatment or independent variable of interest (D) by orthogonalization. It is specifically intended to obtain the estimated error shown in equation (25) below:

$$\hat{V} = D - \hat{m}_0(\boldsymbol{X}), \tag{25}$$

The second step now is to return to the main model (4), but this time, patrial out the effect of the confounder from the response Y using the function $g_0(\boldsymbol{X})$ without adding the term concerning the treatment or independent variable D using the training sample, such as:

$$\widehat{U} = Y - \hat{g}_0(\boldsymbol{X}), \tag{26}$$

Once the estimated error terms are obtained, regress $\widehat{U}$ on $\widehat{V}$ in the Frisch-Waugh-Lovell style to obtain the following DML estimator using the testing set such as:

$$\check{\theta}_0^T = \left(\frac{1}{n}\sum_{i\epsilon\, T}\widehat{V}_i\widehat{V}_i\right)^{-1}\frac{1}{n}\sum_{i\epsilon\, T}\widehat{V}_i\left(Y_i - \hat{g}_0(\boldsymbol{X}_i)\right), \tag{27}$$

The second key element of the construction of the DML estimator is achieved by switching the role of the training, or auxiliary, set with that of the main, or testing, set. The resulting estimators can then be averaged to produce an efficient estimator such as:

$$\check{\theta}_0 = \frac{(\check{\theta}_0^T + \check{\theta}_0^{Tr})}{2}, \tag{28}$$

The advantage of using DML is that it eliminates regularization bias. The scaled estimation error, $\sqrt{N}\left(\check{\theta}_0 - \theta_0\right)$, can be decomposed into three parts using the steps described by Chernozhukov et al. (2018). The expectation of estimation error, $E\left(\check{\theta}_0 - \theta_0\right)$, can be decomposed as the following:

$$E\left(\check{\theta}_0 - \theta_0\right) = \underbrace{\frac{E(\widehat{U}_i\widehat{V}_i)}{E(\widehat{V}_i^2)}}_{a} + \underbrace{\frac{E\left((\hat{m}_0(\boldsymbol{X}_i) - m_0(\boldsymbol{X}_i))(\hat{g}_0(\boldsymbol{X}_i) - g_0(\boldsymbol{X}_i))\right)}{E(\widehat{V}_i^2)}}_{b} + \underbrace{E\left(V_i\left(\hat{g}_0(\boldsymbol{X}_i) - g_0(\boldsymbol{X}_i)\right)\right)}_{c}, \tag{29}$$

The first part of the decomposition, part a, will vanish to 0 if $E\left(\widehat{U}_i\widehat{V}_i\right) = 0$, which is the case if the two terms are independent, and both are expected to have 0 mean. Following the decomposition, the second term, b, expressing the regularization bias will also vanish to 0 if:

$$E\left(\left(\hat{m}_0(\boldsymbol{X}_i) - m_0(\boldsymbol{X}_i)\right)\left(\hat{g}_0(\boldsymbol{X}_i) - g_0(\boldsymbol{X}_i)\right)\right) = 0, \tag{30}$$

This can occur if the two components are uncorrelated, which is the case when using the machine learning estimation to approximate the data generating process for D via $\hat{m}_0(\boldsymbol{X}_i) - m_0(\boldsymbol{X}_i)$, and in Y via $\hat{g}_0(\boldsymbol{X}_i) - g_0(\boldsymbol{X}_i)$, The final term resulting from $E\left(\check{\theta}_0 - \theta_0\right)$ is supposed to be something like:

$$c = E\left(V_i\left(\hat{g}_0(\boldsymbol{X}_i) - g_0(\boldsymbol{X}_i)\right)\right), \tag{31}$$

This term also vanishes in probability since V_i and $\hat{g}_0(\boldsymbol{X}_i)$ are uncorrelated and obtained from different samples, the training set and the test set, respectively. Thanks to the incorporation of the sample splitting technique, the last term which represents the over-fitting bias is eliminated. Figure 2 compares the distribution of the estimated targeted parameter and their biases using three methods: the OLS method, the naive ML method described previously, and the DML proposed by Chernozhukov et al. (2018) using Random Forests.

Figure 2

Comparisons of the Estimation Method for 500 Simulated Data with (n = 500, p = 20)

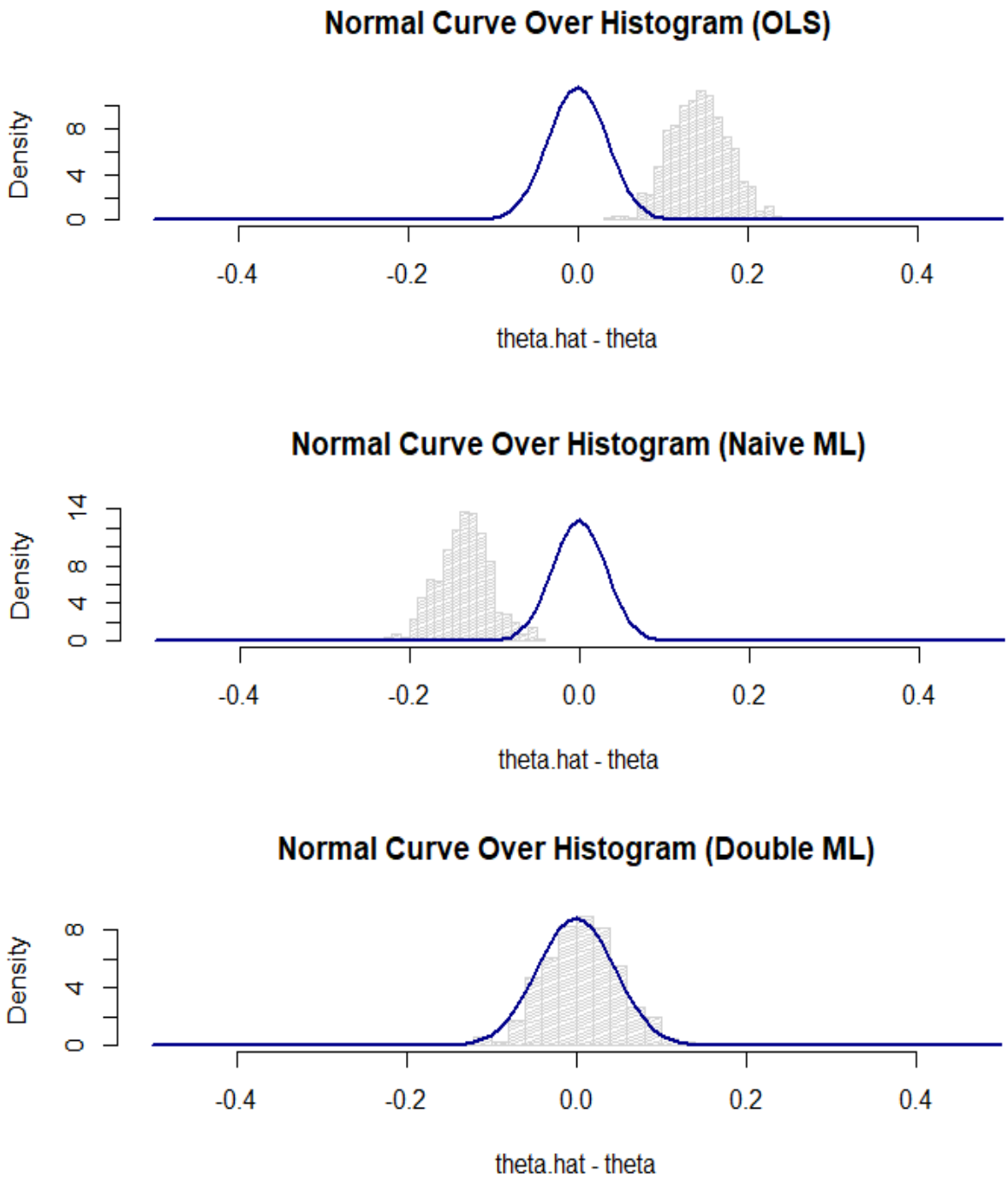


Figure 2 displays the simulated estimators of the OLS, the naive ML estimator from equation (23), and the double ML estimator from equation (28) to showcase the performance of the DML method in comparison with classical methods in terms of estimation bias. The data-generating process follows the partially regression model presented in model (23) and (24). Both error terms, U_i and V_i, are assumed to have a standard normal distribution. Further details about the analysis can be found in the preliminary results section later in this chapter. As it appears in Figure 2, it is not too hard to notice that the distribution of $(\hat{\theta}_0 - \theta_0)$ from the two estimation methods, OLS and NML, are shown to be biased and not centered around 0. On the other hand, the last plot is simulated using the DML method, where the distribution of $(\check{\theta}_0 - \theta_0)$ appears to be unbiased, centered around 0, and aligns with the theoretical distribution.

Calculating the associated estimated variance is required to draw conclusions about the resultant DML estimator obtained in (28). The associated estimated variance for the first sample can take the formula followed by Yang et al (2020) for the DML estimator introduced in this section where cross fitting is used:

$$\check{\sigma}_T^2 = \left(\frac{1}{n}\sum_{i\epsilon\, T} \hat{V}_i^2\right)^{-1} \left(\frac{1}{n}\sum_{i\epsilon\, T} \hat{V}_i^2 \hat{U}_i^2\right) \left(\frac{1}{n}\sum_{i\epsilon\, T} \hat{V}_i^2\right)^{-1} \tag{32}$$

Similarly to what has been done previously when the DML estimator, $\check{\theta}_0$, was constructed, it is necessary to switch the role of the training and testing set to obtain another estimate for constructing the estimated variance such as:

$$\check{\sigma}_{Tr}^2 = \left(\frac{1}{n}\sum_{i\epsilon\, Tr} \hat{V}_i^2\right)^{-1} \left(\frac{1}{n}\sum_{i\epsilon\, Tr} \hat{V}_i^2 \hat{U}_i^2\right) \left(\frac{1}{n}\sum_{i\epsilon\, Tr} \hat{V}_i^2\right)^{-1} \tag{33}$$

Hence, the estimated variance of the DML estimator can be then calculated as follows:

$$\check{\sigma}_0^2 = \frac{1}{2}\left[(\check{\sigma}_T^2 + \check{\sigma}_{Tr}^2) + \left(\check{\theta}_T - \check{\theta}_0\right)^2 + \left(\check{\theta}_{Tr} - \check{\theta}_0\right)^2\right] \tag{34}$$

The construction of a valid confidence interval became feasible once the variance associated with the DML estimator was estimated. The asymptotic confidence interval for the DML estimator for the casual variable, according to Chernozhukov et al. (2018), is calculated as follows:

$$\check{\theta}_0 \pm \Phi^{-1}_{\left(1-\frac{\alpha}{2}\right)} \check{\sigma}_0/\sqrt{N}, \tag{35}$$

where $\Phi^{-1}_{\left(1-\frac{\alpha}{2}\right)}$ is the corresponding z score for the confidence interval and $\check{\sigma}_0/\sqrt{N}$ is the standard error for estimating the DML estimator of $\check{\theta}_0$.

The Super Learner

The issue with the DML approach, as provided in recent studies by Belloni et al. (2014), Chernozhukov et al. (2018), Wang (2020), and Yang (2020), is that the researcher must choose which ML algorithm to use in a particular double selection process. Since the performance of the ML algorithms rely on the success of finding the optimal predictor, the performance of a certain ML will depend on knowing the true distribution from which the dataset was generated. This would lead to the conclusion that predicting which ML algorithm will work best for a given dataset is nearly impossible (Van der Laan et al., 2007).

The aim of the existing papers on DML was to deal with the regularization bias that resulted from the model selection process, and it did so in a very efficient way by using sample splitting across a variety of ML algorithms. However, one might think of the issue on what I would like to call "researcher bias." Which ML algorithm should be incorporated in the analysis? Which learner would be the most effective for given data? Is LASSO a better learner than random forest or support vector machine for estimating the true effect of the independent or treatment variable of interest?

When Van der Laan et al. (2007) proposed the concept of the super learner (SL), they suggested one solution that could help answer these questions. Why not use all of the ML algorithms together at once, rather than debating which one to use? The SL algorithm is an ensemble ML method that employs multiple ML algorithms simultaneously and assigns each one a weight, with the higher weights being assigned to the ML algorithms having the lowest cross-validation risk and vice versa. When it comes to its asymptotic performance, the SL is proven to be as good as, if not better than, any other proposed ML algorithm, according to Van der Laan et al. (2007).

The steps outlined in Van der Laan et al. (2007) and Polley (2010) will be followed to show how the SL algorithm is constructed. In this section, the construction of the SL is conceptually broken down into nine steps for the sake of making the concept of this ML easier to grasp. Before going over these nine steps for constructing the SL algorithm, assume for the sake of simplicity that we are only looking at five different ML algorithms, as Van der Laan et al. (2007) did, denoted as ML_1, …, ML_5. It's worth noting that incorporating a larger set machine is a possibility, though doing so would increase the computational intensity.

The SL employs the technique of cross-validation in addition to the five ML algorithms. For the sake of simplicity and based on empirical studies that favor this cross-validation rate due to the bias-variance trade-off, this section will use v-fold cross-validation (James et al., 2013). Finally, the data are assumed to be typical multiple regression datasets with a large number of predictors, such as:

$$O_i = (Y_i, \boldsymbol{X}_i), \qquad i = 1, \ldots, N,$$

where:

Y_i represents the observed response for the i^{th} case,

$\boldsymbol{X}_i$ represents a matrix of predictors over the i^{th} case.

After defining the parameters for performing the SL in terms of the number of ML algorithms, the v-fold cross-validation specification, and the data structure type, the following nine steps are carried out to obtain super learner predictions.:

1. Input the data alongside the five candidate ML algorithms, ML_1, …, ML_5, which are being considered in constructing the SL analysis. Note that the SL algorithm can incorporate more or less than five learners, but for the sake of clarity, I will limit the analysis to only five.
2. Split the data set into 10 cross-validation blocks such as B_1, B_2, …, B_{10}. In this step, consider the first block of data (i.e., B_1) as a testing set, while treating the remaining nine blocks (i.e., $B_2, \dots, B_{10}$) as the training sets.
3. For each of the ML algorithms considered in the analysis, ML_1, …, ML_5, perform a model fitting using the remaining training sets considered in the blocks $B_2, \dots, B_{10}$.
4. Use the first block considered as a testing set, B_1, to predict the response, Y_i, and then return sets of predicted values, denoted as **Z**, for each machine algorithms such as $Z_{i\in B_1}^{ML_1}, \dots, Z_{i\in B_1}^{ML_5}$.
5. Switch the role of testing and training set 10 times. Consider a different testing and training blocks for each iteration. For example, consider B_2 as a testing block in the second rotation, B_3 in the third rotation, and so on, with the remaining blocks in each rotation serving as training blocks.
6. For each rotation of the blocks' roles, repeat the Steps 3-4 and return sets of predicted values, Z, for each machine algorithms.
7. Construct a matrix of predicted values using all blocks such as:

$$\begin{bmatrix} Z_{i\in B_1}^{ML_1} & \cdots & Z_{i\in B_1}^{ML_5} \\ \vdots & \ddots & \vdots \\ Z_{i\in B_{10}}^{ML_1} & \cdots & Z_{i\in B_{10}}^{ML_5} \end{bmatrix} \tag{36}$$

8. Fit the matrix of predicted values obtained from the previous step against the response Y to propose a family of weights α_1, …, α_5 for each candidate ML algorithms and obtain their estimated values such as:

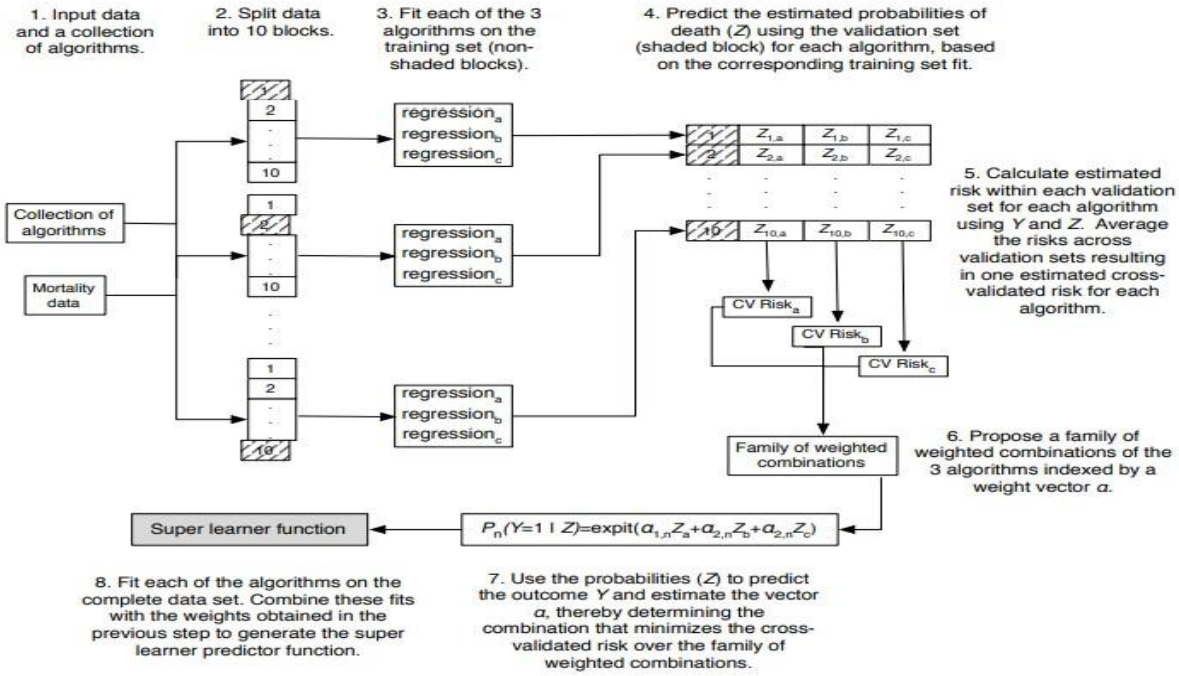
$$Y_i = \hat{\alpha}_1 Z_i^{ML_1} + \cdots + \hat{\alpha}_5 Z_i^{ML_5} + \epsilon_i \tag{37}$$

9. Fit the entire data set using the five ML algorithms, then combine the results with the previous step to generate the SL.

The flow diagram in Figure 3, from Polley et al. (2011) ,illustrates how the SL is constructed in nine steps using 10-fold cross-validation and a variety of classical statistical methods and ML algorithms.

Figure 3

Flow Diagram of the Super Learner Algorithm



Note. From "Targeted learning" by Polley, E. C., Rose, S., & van der Laan, Mark J, (p. 51), 2011, Springer New York (https://doi.org/10.1007/978-1-4419-9782-1_3). Reprinted with permission.

The Proposed Method: Double Super Learner

So far, two concepts have been introduced in the previous two sections of the dissertation: the DML method (Chernozhukov et al., 2018) and the ensemble ML algorithm known as the super learner (Van der Laan et al., 2007). The use ML algorithms is proven to be extremely effective in terms of prediction accuracy. Although a ML algorithm's strong performance in the prediction context does not always imply a similar performance in the estimation context, in fact, ML methods can perform poorly in comparison with statistical methods when it comes to making good causal inferences, whereas the opposite is true when the

goal is to make accurate predictions (Bzdok et al, 2018). The issue is that as the model becomes more complex, the statistical method's inferential performance will become less precise as a large number of nuisance confounders are present in the data set.

Once the DML concept was introduced, comparing the capabilities between ML methods and classical statistical methods in high-dimensional data settings in terms of prediction and estimation became less relevant. The advantage of using the DML method has shown great potentials not only in terms of prediction, where ML methods already have an advantage over statistical methods, but also in terms of estimation performance. In the presence of a high-dimensional nuisance parameter, it is possible to obtain a valid causal inference about the low-dimensional parameter of interest, which could be a causal parameter or treatment effect, using the DML method. Based on theoretical results, empirical examples, and simulations, the resultant estimators of the causal or treatment effect parameter are shown to be approximately unbiased, stable, and root-N consistent, implying that drawing causal conclusions about the targeted parameter via the construction of an associated confidence interval has a high degree of validity.

The second concept introduced in the previous section is the SL, which is an ensemble ML algorithm developed by Van der Laan et al. (2007). The SL is a flexible ML algorithm that can use a multitude of candidate learners to improve prediction accuracy across a wide range of data-generating distributions. Because of the way cross-validation is used, the SL algorithm has the property of being a data-adaptive algorithm, which means that there is no need to restrict the SL to a specific parametric regression fit, as I did in Step 8 of the previous section on the SL algorithm's construction. To put it another way, one can use more data-adaptive ML methods like LASSO or Random Forest to generate the predicted values of each ML algorithm, or even

use the SL itself to complete this step. The heavy use of cross-validation is proven to be an effective way of protecting the final ensemble fit from over-fitting (Polley, 2010).

This section of the dissertation proposes a new method called the DSL method that takes into account the marriage of those two concepts, the DML method and the SL algorithm. The strategy used by Chernozhukov et al. (2018) in constructing the DML estimator by employing the orthogonalization technique using the Frisch-Waugh-Lovell style (Frisch & Waugh, 1933) alongside the cross-fitting technique were incorporated in the DSL method. Furthermore, using a linear weighted functional of different machine-learnings, the super learning algorithm is employed for estimating the nuisance functions $g_0(\boldsymbol{X}_i)$ and $m_0(\boldsymbol{X}_i)$ which express the confounding effects of $\boldsymbol{X}_i$ on the outcome variable Y_i and the independent or treatment variable of interest D_i, respectively. By integrating the two approaches, the DML method and the SL algorithm, the goal of the DSL method to take advantage of the potentials described earlier in producing more accurate estimators of the targeted parameter. In this dissertation, a framework of the DSL method is presented for making a valid causal inference on a low-dimensional targeted parameter, such as the causal or treatment parameter, in the presence of a high-dimensional nuisance parameter, with good performance in terms of estimation accuracy, unbiasedness, and root-N consistency of the produced estimators.

The steps for constructing the targeted parameter estimator using the DSL method will be presented in this section of the dissertation. The data structure assumptions for the DSL method are similar to those used to describe the data structure in the DML method. Furthermore, I used the partial linear regression model of Robinson (1988) introduced earlier in (4) and (5). To make a distinction between the two models used in this section, where the notation used in the DML

method for estimating the functions $g_0(\boldsymbol{X}_i)$ and $m_0(\boldsymbol{X}_i)$ and their associated respective errors U_i and V_i, I used the following notations:

$$Y = D\theta_0 + g_0^*(\boldsymbol{X}) + U^*, \quad E[U^*|\boldsymbol{X}, D] = 0, \tag{38}$$

$$D = m_0^*(\boldsymbol{X}) + V^*, \quad E[V^*|\boldsymbol{X}] = 0, \tag{39}$$

where,

Y represents the response (outcome) variable,

D represents the treatment variable of interest,

X represents a matrix of confounders such as **X** = {x_1, x_2, …, x_p} where p represents the number of associated covariates to be included in the model,

θ_0 represents the treatment effect which is the targeted parameter,

$g_0^*(\mathbf{X})$ represents the nuisance super learner function for the confounders, where **X** influences the outcome variable Y throughout the super learner function g_0^*,

$m_0^*(\mathbf{X})$ represents the nuisance super learner function for the confounders, where **X** influences the treatment variable D throughout the super learner function m_0^*,

U^* represents the model error in (38),

V^* represents the model error in (39).

In addition to the model assumptions, similar steps used in the SL section with respect to ML candidates were used along with the use of cross-validation folds, which involve using five algorithms and a 10-fold cross-validation approach. With these considerations in mind, the following are the steps of the proposed DSL algorithm for estimating the causal or treatment effect parameter:

1. Split the dataset into two parts as in the DML method, with a testing set denoted as T, and a training set which denoted as Tr.

2. Consider model (39), apply the nine steps the SL algorithm introduced earlier using the sample assigned for the training set to obtain the following predictive linear model as follows:

$$D_{i\in Tr} = \hat{\beta}_1 W_{i\in Tr}^{ML_1} + \cdots + \hat{\beta}_5 W_{i\in Tr}^{ML_5} + \hat{V}_{i\in Tr}^{*} = \sum_{k=1}^{5}\left(\hat{\beta}_k W_{i\in Tr}^{ML_k}\right) + \hat{V}_{i\in Tr}^{*}, \quad (40)$$

where:

$\hat{\beta}_k$ represents the estimated weight of the k^{th} ML algorithm using the training set,

$W_{i\in Tr}^{ML_k}$ represents the predicted values of the independent or treatment variable D using the k^{th} ML algorithm based on the training set,

$\hat{V}_{i\in Tr}^{*}$ represents estimated errors from predicting the independent or treatment variable using the super learner method.

3. Partial out the effect of $W_{i\in Tr}^{ML_k}$ from the independent or treatment variable $D_{i\in Tr}$ based on the training sample to obtain the orthogonalized estimated error such as:

$$\hat{V}_{i\in Tr}^{*} = D_{i\in Tr} - \sum_{k=1}^{5}\left(\hat{\beta}_k W_{i\in Tr}^{ML_k}\right), \quad (41)$$

4. Now consider model (38), which is concerned with the response, and apply the nine steps of the SL algorithm introduced earlier based on the training sample to obtain the following predictive linear model:

$$Y_{i\in Tr} = \hat{\alpha}_1 Z_{i\in Tr}^{ML_1} + \cdots + \hat{\alpha}_5 Z_{i\in Tr}^{ML_5} + \hat{U}_{i\in Tr}^{*} = \sum_{k=1}^{5}\left(\hat{\alpha}_k Z_{i\in Tr}^{ML_k}\right) + \hat{U}_{i\in Tr}^{*}, \quad (42)$$

where:

$\hat{\alpha}_k$ represents the estimated weight of the k^{th} ML algorithm using the training set,

$Z_{i\in Tr}^{ML_k}$ represents the predicted values of the outcome variable Y using the k^{th} ML algorithm based on the training set,

$\widehat{U}^{*}_{i\epsilon Tr}$ represents the estimated errors from predicting the response using the super learner method.

5. Partial out the effect of $Z^{ML_k}_{i\epsilon Tr}$ from the observed response $Y_{i\epsilon Tr}$ based on the training sample to obtain the orthogonalized estimated error such as:

$$\widehat{U}^{*}_{i\epsilon Tr} = Y_{i\epsilon Tr} - \sum_{k=1}^{5}\left(\hat{\alpha}_k Z^{ML_k}_{i\epsilon Tr}\right), \tag{43}$$

6. After obtaining $\widehat{U}^{*}_{i\epsilon Tr}$ and $\widehat{V}^{*}_{i\epsilon Tr}$ using the training sample, it is the time to apply the Frisch-Waugh-Lovell style by regressing $\widehat{U}^{*}_{i\epsilon Tr}$ on $\widehat{V}^{*}_{i\epsilon Tr}$ using the testing set this time and obtain the following DSL estimator such as:

$$\tilde{\theta}_0^{T} = \left(\frac{1}{n}\sum_{i\epsilon\, T}\widehat{V}^{*}_i\widehat{V}^{*}_i\right)^{-1}\frac{1}{n}\sum_{i\epsilon\, T}\widehat{V}^{*}_i\widehat{U}^{*}_i, \tag{44}$$

7. After obtaining the first DSL estimator based on the testing set, switch the role of the training and testing sets and repeat the previous steps to get a second DSL estimator such as:

$$\tilde{\theta}_0^{Tr} = \left(\frac{1}{n}\sum_{i\epsilon Tr}\widehat{V}^{*}_i\widehat{V}^{*}_i\right)^{-1}\frac{1}{n}\sum_{i\epsilon\, Tr}\widehat{V}^{*}_i\widehat{U}^{*}_i, \tag{45}$$

8. As a result, perform cross-fitting by averaging the two resulted DSL estimators to obtain the final estimator of the causal or treatment effect parameter, as described previously in the DML method:

$$\tilde{\theta}_0 = \frac{(\tilde{\theta}_0^{T} + \tilde{\theta}_0^{Tr})}{2}, \tag{46}$$

9. Finally, calculating the estimated variance is critical in order to be able to draw a causal conclusion by constructing a confidence interval for the resulted DSL estimator, $\tilde{\theta}_0$, the estimated variance resulting from the first sample, like the DML method, is of the white-type estimator:

$$\tilde{\sigma}_T^2 = \left(\frac{1}{N_T}\sum_{i\epsilon\, T} \hat{V}_i^{*2}\right)^{-1} \left(\frac{1}{n}\sum_{i\epsilon\, T} \hat{V}_i^{*2}\hat{U}_i^{*2}\right) \left(\frac{1}{n}\sum_{i\epsilon\, T} \hat{V}_i^{*2}\right)^{-1}, \quad (47)$$

The variance estimator can then be obtained by switching the role of the training and the testing sets and computing the variance of the second sample, $\tilde{\sigma}_{Tr}^2$, as follows:

$$\tilde{\sigma}_0^2 = \frac{1}{2}\left[(\tilde{\sigma}_T^2 + \tilde{\sigma}_{Tr}^2) + \left(\tilde{\theta}_T - \tilde{\theta}_0\right)^2 + \left(\tilde{\theta}_{Tr} - \tilde{\theta}_0\right)^2\right] \quad (48)$$

After applying the previous nine steps and obtaining the resultant DSL estimator and its estimated variance, the classical confidence interval can be constructed as follows to draw causal conclusions about the estimated targeted parameter, $\tilde{\theta}_0$, such as:

$$\tilde{\theta}_0 \pm \Phi^{-1}_{\left(1-\frac{\alpha}{2}\right)} \tilde{\sigma}_0/\sqrt{N} \quad (49)$$

The distribution of $(\tilde{\theta}_0 - \theta_0)$, as shown in Figure 4, displays the distributional behavior of the resultant causal effect estimates using the DSL approach. This estimator is shown to behave well, in that it is approximately unbiased and aligns with the theoretical asymptotic normal distribution, the blue curve, which is centered on 0 and diverges according to the DSL estimator's variance.

Figure 4

The Distribution of $(\tilde{\theta}_0 - \theta_0)$ Using the Double Super Learner Method (n = 500, p = 20)

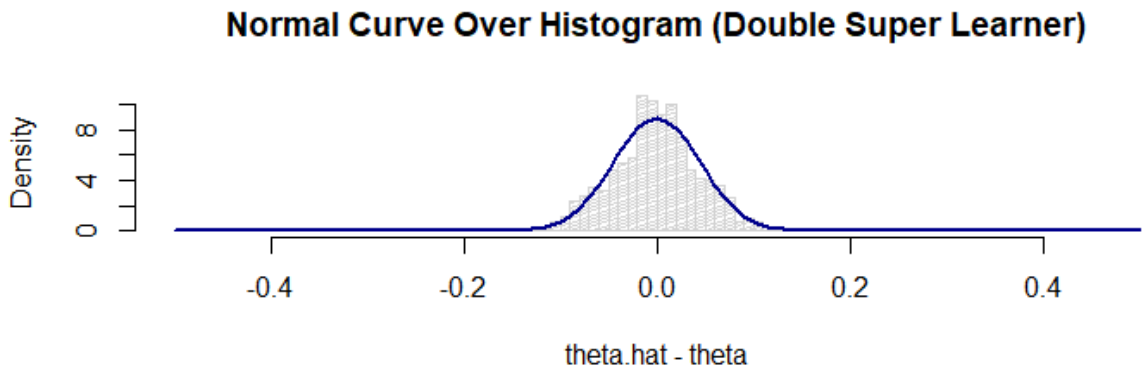
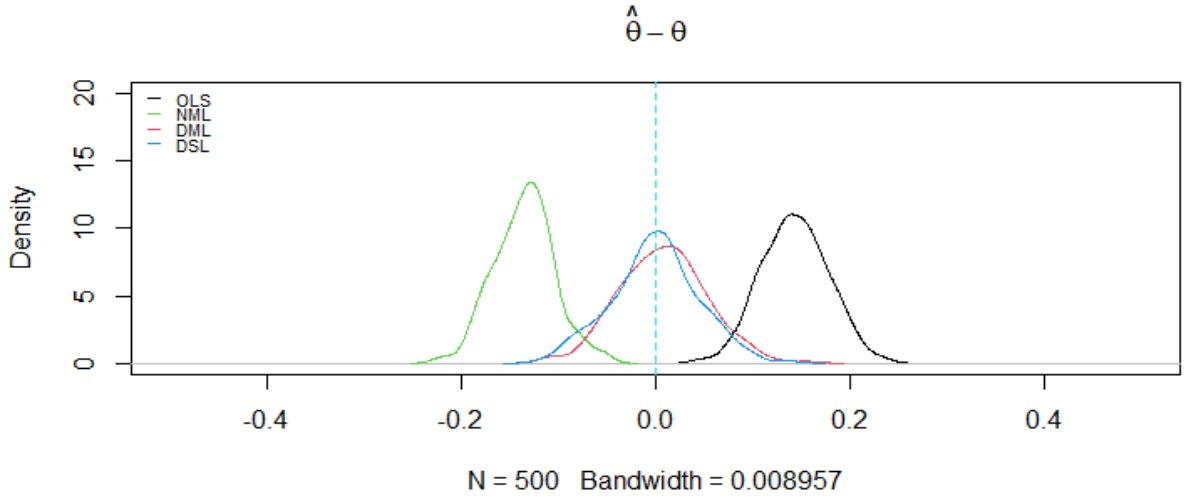


Figure 5 was next created to compare the performance of the four candidate estimation methods discussed in this dissertation. It can be seen that estimators based on the DML and DSL methods outperformed estimators based on the OLS method and naive ML method approaches.

Figure 5

Comparison of the Distribution Density of ($\hat{\theta}_0 - \theta_0$) for All Methods (n = 500, p = 20)



From Figure 5, it is clear that the DML estimator, in red, and the DSL estimator, in blue, have outperformed the OLS, in black, and the naive machine learning (NML) estimator, in green. Both of the latter estimators, OLS and NML, are shown to be biased and deviated significantly from the true value of the parameter θ_0 which was set to 0.5. On the other hand, the resultant estimates from applying the DML and DSL were shown to be unbiased, and that $\hat{\theta}_0 - \theta_0$ is centered around the 0, and that the DSL method yields a set of estimates that are as good, if not slightly better, as the ones that resulted from applying the DML method.

Improving the Computational Efficiency of the Double Super Learner Algorithm

When comparing the computational intensity of the DML method to the proposed method, the DSL, the DML method is more efficient computationally. This difference in computational efficiency between the DML and DSL methods is due to the DSL method's use of multiple ML algorithms. The selection property can be employed when performing the DSL method to reduce the effect of the computational intensity caused by the incorporation of multiple ML algorithms. This can be achieved by applying the DSL method to a subset of the sample, and then selecting the best performing ML algorithm based on the size of the weights that correspond to each ML algorithm. Empirical simulations performed by Polley (2010) show that when running the SL function for a given data dataset on a set of candidate ML algorithms, the algorithm associated with the highest weight in the prediction model will always have the lowest cross-validated risk, indicating that it is the best candidate learners for the given dataset. To apply the selection concept of the best performing learner in the DSL method, the following steps can be followed:

1. Apply the DSL method on a small sample of the dataset, Steps 1-9 in the previous section of the DSL algorithm, using all of the candidate learners.
2. Use the information obtain in Steps 2 and 4 by selecting the learners with the highest weights. Specifically, select the learner with the largest $\hat{\beta}_k$ in (40) that predicts the treatment variable D, and the largest $\hat{\alpha}_k$ in (42) that predicts the response Y.
3. Apply the DSL method on the entire dataset using only the best performing learners selected in the previous step.

To apply the selection concept on the DSL method with the aim of improving the computational efficiency, the results from the previous DSL simulation were incorporated since

there were 500 replications using the DSL method, and with each replication, the values of the candidate machine-learnings weights were saved and then averaged for each learner to have the overall weight. Table 1 shows the results of the weights of the five machine-learnings incorporated in the SL function from the previous section for predicting the response in training and testing sets, denoted as Y_t and Y_{tr}, as well as the treatment variable, denoted as D_t and D_{tr}.

Table 1

The Average of Estimated Weights for the Response and the Treatment Variables across Five Machine Learnings in the Super Learner Given the Training and Testing Sets

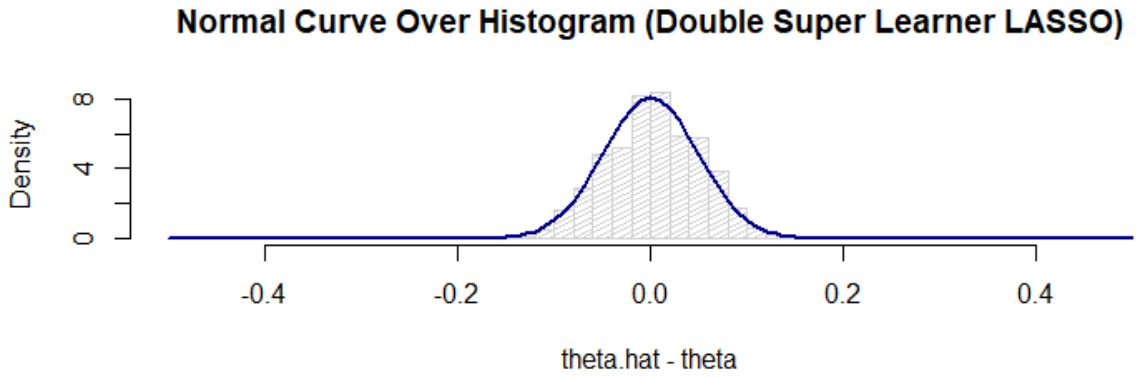
Nuisance	LASSO	GLM	KNN	Random Forest	Boosting
$\widehat{\boldsymbol{g}}_{\boldsymbol{0}}(\boldsymbol{X}_{\boldsymbol{i}\in\boldsymbol{T}})$	0.77119529	0.08508627	0.03579099	0.06164191	0.04628554
$\widehat{\boldsymbol{g}}_{\boldsymbol{0}}(\boldsymbol{X}_{\boldsymbol{i}\in\boldsymbol{Tr}})$	0.76543355	0.09002393	0.03938016	0.06071164	0.04445072
$\widehat{\boldsymbol{m}}_{\boldsymbol{0}}(\boldsymbol{X}_{\boldsymbol{i}\in\boldsymbol{T}})$	0.79465935	0.07550397	0.03251870	0.04963991	0.04767807
$\widehat{\boldsymbol{m}}_{\boldsymbol{0}}(\boldsymbol{X}_{\boldsymbol{i}\in\boldsymbol{Tr}})$	0.81401576	0.06839084	0.03225440	0.04177494	0.04356406

It is not too hard to notice that the LASSO learner had the highest weight, indicating that using LASSO to predict the response and treatment variables has the lowest cross-validation risk. As a result, given the simulated datasets, it is safe to assume that LASSO is the best ML algorithm to use in the DSL method. To test this theory, the simulation is reiterated using only LASSO for the DSL method to see if the resultant estimates for the targeted parameter were influenced by the selection.

Figure 6 displays the distribution of the targeted parameter estimates, $\tilde{\theta}_0$, obtained using the DSL method when LASSO is used only as a machine-learning algorithm in the super learner function.

Figure 6

The Distribution of ($\tilde{\theta}_0 - \theta_0$) Using for DSL Method Using Only LASSO



The resultant estimates appear to be unbiased and follow the theoretical distribution, as shown in the Figure 6, indicating that the SL function-based selection was able to improve the computational efficiency of the DSL method by 80% while still producing valid estimators. The summary results in Table 2 tell the story about which method was closest to the true value of the targeted parameter by taking the average of all resulted estimators from the 500 replications in the simulated data.

Table 2

The Average of Estimated Targeted Parameter for 500 Replicates

Method	OLS	NML	DML	DSL	DSL $_{\text{LASSO}}$
$\widehat{\boldsymbol{\theta}}_{\mathbf{0}}$	0.6429	0.3640	0.5061	0.5001	0.4988

The results displayed in Table 2 clearly show that the average estimates yield a smaller difference from the real value of $\theta_0 = 0.5$ when using the DML, DSL, and DSL $_{\text{LASSO}}$ in comparison with the OLS and NML methods, with a difference that favors DSL $_{\text{LASSO}}$, DSL, and DML methods, respectively. More specifically, the improvement applied to the proposed DSL methods has resulted in a more accurate estimator in comparison with the original DML estimator when considering the difference from the real value.

Table 3 provides more statistical insight into the performance between the three methods. In Table 3, the estimates of the targeted parameter, the bias, the variance, and the corresponding 95% confidence intervals across the tree methods are reported. In viewing Table 3, it is not too had to notice that the proposed method that DSL method, is shown to be the best performing method in terms of the bias size. The selected DSL method, DSL $_{\text{LASSO}}$, that only incorporates LASSO in its estimation comes second, and the DML method comes last among the three methods. In terms of variance, on the other hand, the simulation results show that the original DML method yielded the lowest variance, then comes the DSL $_{\text{LASSO}}$ and DSL methods, respectively.

Table 3

The Average of Estimated Weights for the Response and the Treatment Variables Across Five Machine Learnings in the Super Learner Given the Training and Testing Sets

Method	Estimate	Bias	Variance	95% CI	
DML	0.5061	0.0061	0.0020	0.4179	0.5944
DSL	0.5001	0.0001	0.0023	0.4071	0.5932
DSL $_{\text{LASSO}}$	0.4988	0.0012	0.0022	0.4060	0.5917

Note. Number of replications is 500, p=20, and N=500.

CHAPTER IV

RESULTS

In this chapter of the dissertation, I will be performing analytical simulation in order to investigate the theory behind the proposed DSL method and to be able answer the four research questions introduced in Chapter I. The numerical analyses will be implemented using R software using both simulated and real datasets.

In this chapter, the content will be split into six sections. The first section will present the simulation scheme which includes the data-generating process, the ML algorithms, estimation methods, the dimensions of the simulated datasets, the type of treatments, etc. The second section of this chapter will be looking at numerical results of the proposed method when the treatment variable is considered to be continuous across a grid of values regarding the sample size and the number of predictors included in the analysis. Similar to the second section, the third section of this chapter will explore various simulated datasets in the case where the treatment variable is considered to be binary.

The fourth section of this chapter will introduce a new R package that was created specifically for this dissertation in order to make it is easier for others to perform the analysis needed using the DSL method. The fifth section of this chapter will consider an empirical example of a real-life dataset which aims to demonstrate the application of the DSL method and to show the case of using the DoubleSL package that was developed specifically for the

proposed method in this dissertation. Finally, the sixth section will present findings of the analysis and answers for the research questions.

Simulation Scheme

In this section of the chapter, I will introduce a simulation scheme to help me navigate through the analysis in this chapter. In order to explore the proposed DSL method performance under a variety of settings, I investigated the true effect of treatment using two scenarios: when the treatment variable is continuous, and when the treatment variable is binary.

For the datasets to be simulated in this chapter, I assumed the following data structure similar to one introduced earlier in the methodology chapter:

$$\boldsymbol{O}_i = (Y_i, D_i, \boldsymbol{X}_i),\ i = 1, \dots, 500$$

whereas before, the response is represented via the variable Y_i, the treatment is represented via the variable D_i, and the matrix $\boldsymbol{X}_i$ represents the set of covariates. Since the DML method is the benchmark method that I compared with the proposed DSL method, I followed the same simulation procedure followed by Chernozhukov et al. (2018). For each observation in $\boldsymbol{O}_i$, the following data-generating process was used to generate the data:

$$Y_i = D_i\theta_0 + g_0(\boldsymbol{X}_i) + U_i, \quad U_i \sim N(0,1) \tag{50}$$

$$D_i = m_0(\boldsymbol{X}_i) + V_i, \quad V_i \sim N(0,1) \tag{51}$$

In addition, the matrix of covariates is simulated from a multivariate normal density such as $\boldsymbol{X}_i \sim N_p(0, \boldsymbol{\Sigma})$, where p represents the number of covariates in $\boldsymbol{X}_i$. For $j = 1, 2, \dots, p$; the covariance matrix, $\boldsymbol{\Sigma}$, was simulated as follows:

$$\boldsymbol{\Sigma} = \text{toeplitz}(0.7^p) = \begin{pmatrix} 0.7^1 & \dots & 0.7^p \\ \vdots & \ddots & \vdots \\ 0.7^p & \cdots & 0.7^1 \end{pmatrix}, \tag{52}$$

Furthermore, assume that the nuisance functions $g_0(\boldsymbol{X}_i)$ and $m_0(\boldsymbol{X}_i)$, which describe the relationships of the confounders on the response and the treatment variable, respectively, can be simulated as follows:

$$g_0(\boldsymbol{X}_i) = expit(\beta_1 \boldsymbol{X}_i) \; + 0.25 \; \beta_2 \boldsymbol{X}_i \tag{53}$$

$$m_0(\boldsymbol{X}_i) = \beta_3 \boldsymbol{X}_{i1} + 0.25 \; expit(\beta_4 \boldsymbol{X}_{i3}) \tag{54}$$

where the parameters associated in the data generating process are as follows:

$$\beta_1 = \beta_3 = \{1, 0, 0, \dots\}, \qquad \beta_2 = \beta_4 = \{0, 0, 1, 0, \dots\}, \quad \theta_0 = 0.5$$

Obviously, the above data-generating process will return a dataset that has a continuous treatment variable. In order to create a dataset with a binary treatment variable, the above data-generated process was slightly adjusted; mainly, equation (51) will be changed to:

$$D_i \sim Binomial\left(n = 1, p = \frac{1}{1+\exp(m_0(\boldsymbol{X}_i))}\right), \tag{55}$$

To compensate for the error term V_i in equation (51) when considering the case of the treatment variable as being binary, the approach followed in Yang et al. (2020) was considered to return the necessary values to construct the DML and DSL estimators such as:

$$V_i = \begin{cases} 1 - p, & with\ probability\ p \\ -p, & with\ probability\ 1 - p \end{cases}, \tag{56}$$

where $p = 1/\big(1 + \exp\big(m_0(\boldsymbol{X}_i)\big)\big)$ as noted in (55).

Furthermore, the DML estimator and the DSL estimators were constructed using the orthogonal method introduced in equations (27) and (44), respectively, when considering the treatment variable to be continuous. When the case of the treatment variable is binary, on the other hand, an additional estimator is introduced as suggested by Chernozhukov et al. (2018). The additional estimator using the DML method when the nuisance function is estimated using the testing set was constructed as follows:

$$\check{\theta}_0^T = \left(\frac{1}{n}\sum_{i\epsilon\, T}\hat{V}_i D_i\right)^{-1}\frac{1}{n}\sum_{i\epsilon\, T}\hat{V}_i\left(Y_i - \hat{g}_0(\boldsymbol{X}_i)\right), \quad (57)$$

After the first estimator is obtained, the role of the testing and training sets will switch as before and obtain a second estimator $\check{\theta}_0^{Tr}$, which is then averaged with $\check{\theta}_0^{Tr}$ and the estimator $\check{\theta}_0$ retained as in (28). A similar framework was followed to obtain the DSL estimators. It is important to point out that the estimation using equations (27) and (44), which were incorporated in estimating the effect of continuous treatment variables, were also implemented in estimating the effect of binary treatment variables where the analysis results are retained (Appendix B).

When implementing the DML method, the Random Forest ML algorithm was used to estimate the targeted parameter. As for the DSL method, a set of five candidate learners were incorporated: (a) the general linear model (GLM); (b) the kernel KNN; (c) the LASSO; (d) the Random Forests; and (e) the boosting. The main reason for selecting these candidate algorithms was to have a set of various learners in terms of their constructions and their searching strategy. In addition, the number of 5-fold cross-validation was considered when implementing the DSL methods.

For each of those two types of treatment variables, 16 different data settings were simulated using the data-generating processes described earlier, each of which was replicated over 500 times, and the targeted parameter was investigated using a grid of values for the sample size and the number of extraneous variables (n and p) (Table 4). Using two types of treatment variables alongside four different sample sizes and number of associated covariates, the simulation investigated 32 different datasets that differed in their settings.

Table 4

Settings for Sample Size and Number of Extraneous Variables under Investigation

Number of covariates	$n_1=100$	$n_2=500$	$n_3=1,000$	$n_4=5,000$
$p_1=20$	Simulation 1	Simulation 2	Simulation 3	Simulation 4
$p_2=100$	Simulation 5	Simulation 6	Simulation 7	Simulation 8
$p_3=1,000$	Simulation 9	Simulation 10	Simulation 11	Simulation 12
$p_4=10,000$	Simulation 13	Simulation 14	Simulation 15	Simulation 16

For each dataset, three methods investigated the performance of estimating the targeted parameter: the DML method, the DSL method, and the DSL method using the selection criteria. For each of these three methods, a number of statistics were reported and compared: the estimated targeted parameter, the variance, the bias, and the associated upper and lower confidence limits. In addition, a histogram of the estimated targeted parameter is presented to display the behavior for the simulated estimated effects.

Simulation Analysis for Continuous Case Treatment

In this section, I explore the results from applying the DSL method on 16 different settings to simulate datasets that vary in their number of associated covariates and sample sizes. These datasets were simulated using the data-generating process explained in the previous simulation scheme section of this chapter, equations (50) through (54). In addition, the DSL method is compared with the existing DML method as well as with the improved DSL version

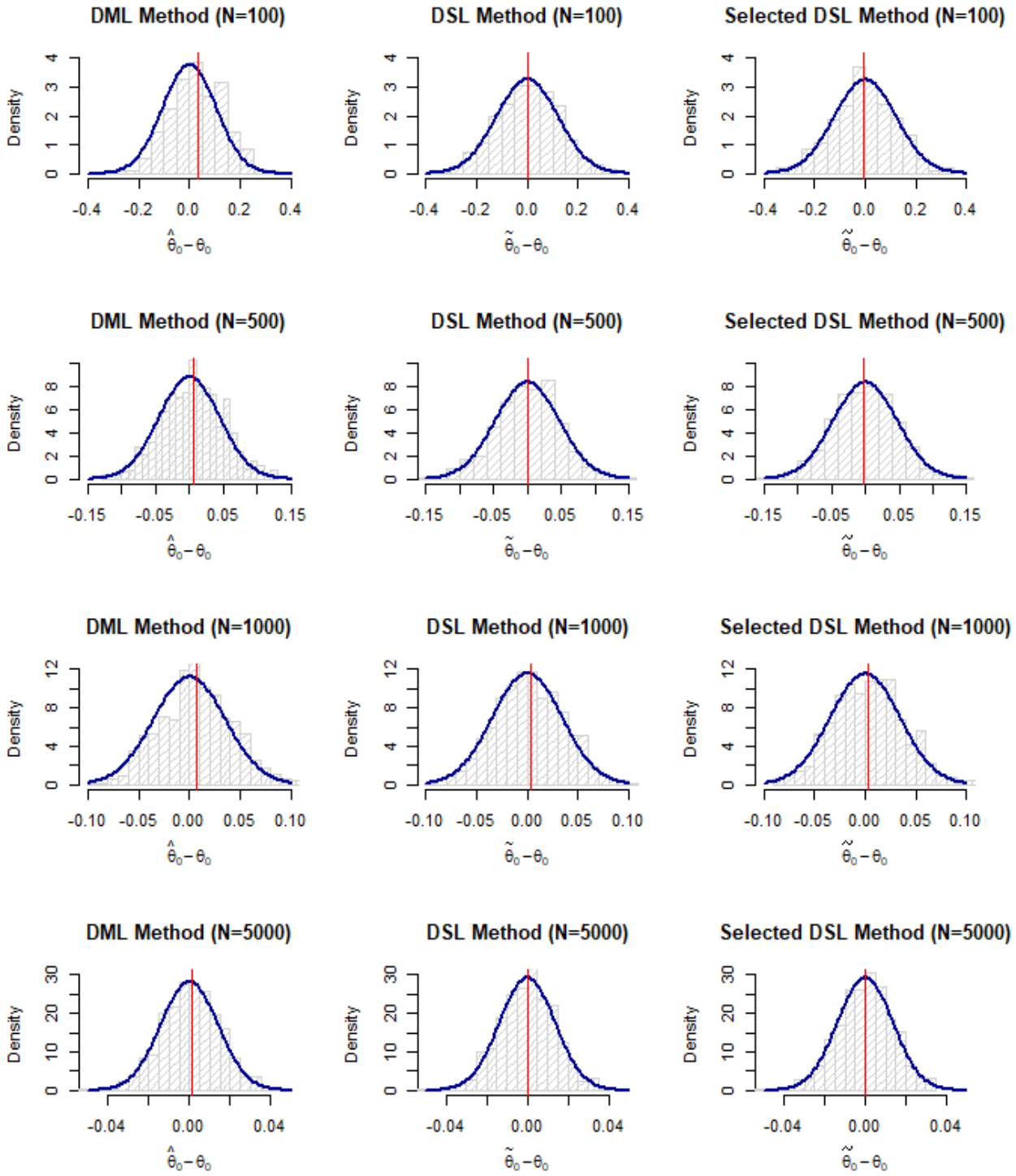
proposed in the previous chapter, the selected DSL method. The information about selecting the best ML algorithm was obtained using the SL function that is incorporated in the DSL algorithm.

In this simulation, I considered four different numbers of associated covariates (p) in the datasets: 20, 100, 1000, and 10000. For each different p, four different sample sizes (N) were considered: 100, 500, 1000, and 5000. The number of replications considered in the analysis was originally 500 replications, but due to the implementation of the analysis across multiple nodes, the number of replications applied in the analysis was 504 replications for the majority of the datasets. In addition, the number of replications considered for analyzing the largest datasets, sample sizes (N) of 5,000 and the number of associated covariates (p) of 10,000, was limited to 264 due to the computational intensity when analyzing such large datasets.

When the simulation analysis for the three estimation methods was implemented, the histograms for the density of the targeted parameters' estimates using each estimation method were retained in order to observe and compare their normal densities and its alignment with the theoretical distribution. To investigate performance of estimating the targeted parameter in the presence of 20 associated covariates, a simulation using 504 replications with different simulated datasets was performed, and the results are displayed in Figure 7.

Figure 7

The Distribution of ($\tilde{\theta}_0 - \theta_0$) when p = 20 when the Treatment Variable is Continuous



The results displayed in Figure 7 show the distribution of the estimated targeted parameter of each method subtracted by the true value of the targeted parameter (θ_0=0.5) when the number of associated covariates is fixed at 20 and the sample size number is variant, which is considered a typical case of low-dimensional datasets, p < N. Figure 7 shows that by applying the proposed DSL method as well as the selected DSL method, a similar distributional behavior of the existing DML method can be achieved judging by the alignment between the density of the results from each estimation method and the blue curve which represents the theoretical normal distribution of 0 mean and the estimates' variance. Furthermore, the red line displayed in Figure 7, which refers to the bias that resulted from each estimation method, clearly shows that the DSL method as well as the selected DSL method have lower bias in comparison with the original DML method proposed by Chernozhukov et al. (2018). The bias was calculated by taking the mean of all estimates subtracted by the true parameter θ_0.

To assess the performance of the candidate ML algorithms that are considered in the DSL method, Table 5 was produced. The information that Table 5 presents is also crucial in order to know which ML algorithm should be considered when implementing the selected DSL method using two indicators: frequency and ratio. For example, when considering the sample size of 100, LASSO was the best performing learner when estimating the nuisance function $\hat{g}_0$ using the testing set, $\hat{g}_0(\boldsymbol{X}_{i\in T})$, 246 times out of 504 and 266 times when $\hat{g}_0$ is estimated using the training set, $\hat{g}_0(\boldsymbol{X}_{i\in Tr})$. In addition, LASSO has the largest weight of 0.452 and 0.451 when estimating $\hat{g}_0(\boldsymbol{X}_{i\in T})$ and $\hat{g}_0(\boldsymbol{X}_{i\in Tr})$, respectively, in comparison with the other ML algorithms. It is clear that the choice of LASSO is obvious as the sample size grow large.

Table 5

Performance of Each Candidate Learner in the Super Learner Analysis for p = 20 when the Treatment Variable is Continuous

Sample N	Nuisance	Best Candidate	LASSO	GLM	KNN	RF	Boosting
100	$\hat{g}_0(X_{i\in T})$	Frequency	246	23	84	117	34
		Ratio	0.452	0.093	0.161	0.192	0.102
	$\hat{g}_0(X_{i\in Tr})$	Frequency	266	19	87	104	28
		Ratio	0.451	0.092	0.161	0.192	0.104
	$\hat{m}_0(X_{i\in T})$	Frequency	374	25	23	52	30
		Ratio	0.637	0.084	0.065	0.112	0.103
	$\hat{m}_0(X_{i\in Tr})$	Frequency	372	16	22	63	31
		Ratio	0.634	0.089	0.069	0.107	0.101
500	$\hat{g}_0(X_{i\in T})$	Frequency	458	18	2	25	1
		Ratio	0.749	0.082	0.042	0.085	0.041
	$\hat{g}_0(X_{i\in Tr})$	Frequency	458	19	0	24	3
		Ratio	0.749	0.084	0.042	0.074	0.050
	$\hat{m}_0(X_{i\in T})$	Frequency	490.	7	0	6	1
		Ratio	0.831	0.056	0.032	0.040	0.041
	$\hat{m}_0(X_{i\in Tr})$	Frequency	494.	9	0	0	1
		Ratio	0.825	0.063	0.035	0.035	0.042
1,000	$\hat{g}_0(X_{i\in T})$	Frequency	481.	18	0	5	0
		Ratio	0.798	0.083	0.025	0.062	0.031
	$\hat{g}_0(X_{i\in Tr})$	Frequency	480.	18	0	6	0
		Ratio	0.818	0.076	0.024	0.052	0.031
	$\hat{m}_0(X_{i\in T})$	Frequency	500.	4	0	0	0
		Ratio	0.848	0.058	0.027	0.038	0.028
	$\hat{m}_0(X_{i\in Tr})$	Frequency	493.	11	0	0	0
		Ratio	0.847	0.060	0.027	0.035	0.031
5,000	$\hat{g}_0(X_{i\in T})$	Frequency	496.	8	0	0	0
		Ratio	0.858	0.078	0.013	0.033	0.017
	$\hat{g}_0(X_{i\in Tr})$	Frequency	494.	10	0	0	0
		Ratio	0.847	0.088	0.013	0.035	0.017
	$\hat{m}_0(X_{i\in T})$	Frequency	493.	11	0	0	0
		Ratio	0.871	0.068	0.016	0.031	0.014
	$\hat{m}_0(X_{i\in Tr})$	Frequency	497.	7	0	0	0
		Ratio	0.876	0.067	0.018	0.024	0.015

To complete the investigation comparing the performance of the proposed methods with the original DML method when the number of associated covariates is set to 20, Table 6 provides summary statistics across the three methods for different sample sizes. For each

method, the overall estimates of the targeted parameter, bias, variance and 95% confidence intervals are reported.

Table 6

Summary Statistics for Datasets with p = 20 when the Treatment Variable is Continuous

N	Method	Estimates	Bias	Variance	95% CI	
100	DML	0.5316	0.0316	0.0112	0.3246	0.7386
	DSL	0.5043	0.0043	0.0148	0.2659	0.7427
	Selected DSL	0.4964	0.0036	0.0150	0.2560	0.7368
500	DML	0.5061	0.0061	0.0020	0.4179	0.5944
	DSL	0.5001	0.0001	0.0023	0.4071	0.5932
	Selected DSL	0.4988	0.0012	0.0022	0.4060	0.5917
1,000	DML	0.5069	0.0069	0.0013	0.4372	0.5766
	DSL	0.5037	0.0037	0.0012	0.4365	0.5708
	Selected DSL	0.5036	0.0036	0.0012	0.4361	0.5711
5,000	DML	0.5011	0.0011	0.0002	0.4734	0.5287
	DSL	0.5000	0.0000	0.0002	0.4734	0.5267
	Selected DSL	0.5002	0.0002	0.0002	0.4735	0.5268

Note. Number of replications is 504.

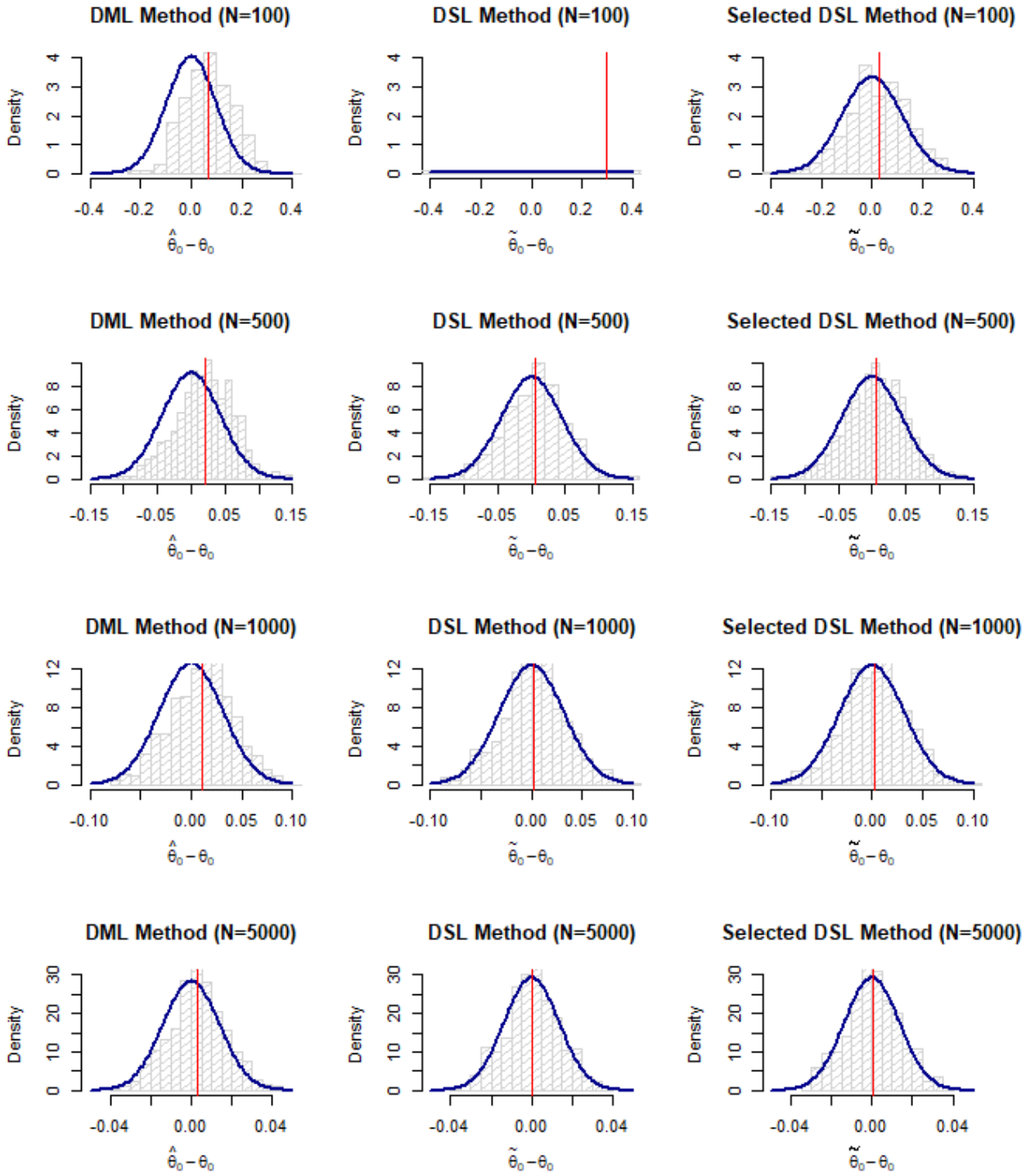
As displayed in Table 6, the summary statistics show that the DSL method and the selected DSL method, which only incorporated LASSO in the estimation process, have lower bias than that of the DML method across different sample sizes. In addition, the associated variances of the proposed methods are fairly close to that of the original DML method, and approach it as the sample size increases, resulting in more competitive and valid confidence intervals.

The following analysis considered applying the proposed DSL methods when the number of associated covariates is set to 100, where it contains a case p = N. Figure 8 displays the distribution of $(\tilde{\theta}_0 - \theta_0)$ using the three methods for comparing their normal densities across four different sample sizes. For the most part, the distribution of $(\tilde{\theta}_0 - \theta_0)$ was shown to be

asymptotically normal in a way that is similar to previous cases when the number of associated covariates was set to be equal to 20. However, it is not too hard to notice that one of these cases behaved differently, which is the histogram that displays the distribution of $(\tilde{\theta}_0 - \theta_0)$ using the DSL method when the sample size was set to 100. When investigating the reason why the histogram looks this way, it appears that the DSL method has returned some values for the estimated targeted parameter that are unusual for what is expected. For example, the 64^{th} replication returned an estimate for the targeted parameter of 138.08, which was far from what is expected when the true value of the targeted parameter is set to 0.5. It is worth mentioning that the 64^{th} replication was the only instance that resulted in an inflated estimate, but it was enough to cause the bias, and the variance significantly increased in comparison with other methods and across other sample sizes. Outliers' detection analysis can be found in Appendix A for this and other cases.

Figure 8

The Distribution of ($\tilde{\theta}_0 - \theta_0$) when p = 100 and when the Treatment Variable is Continuous

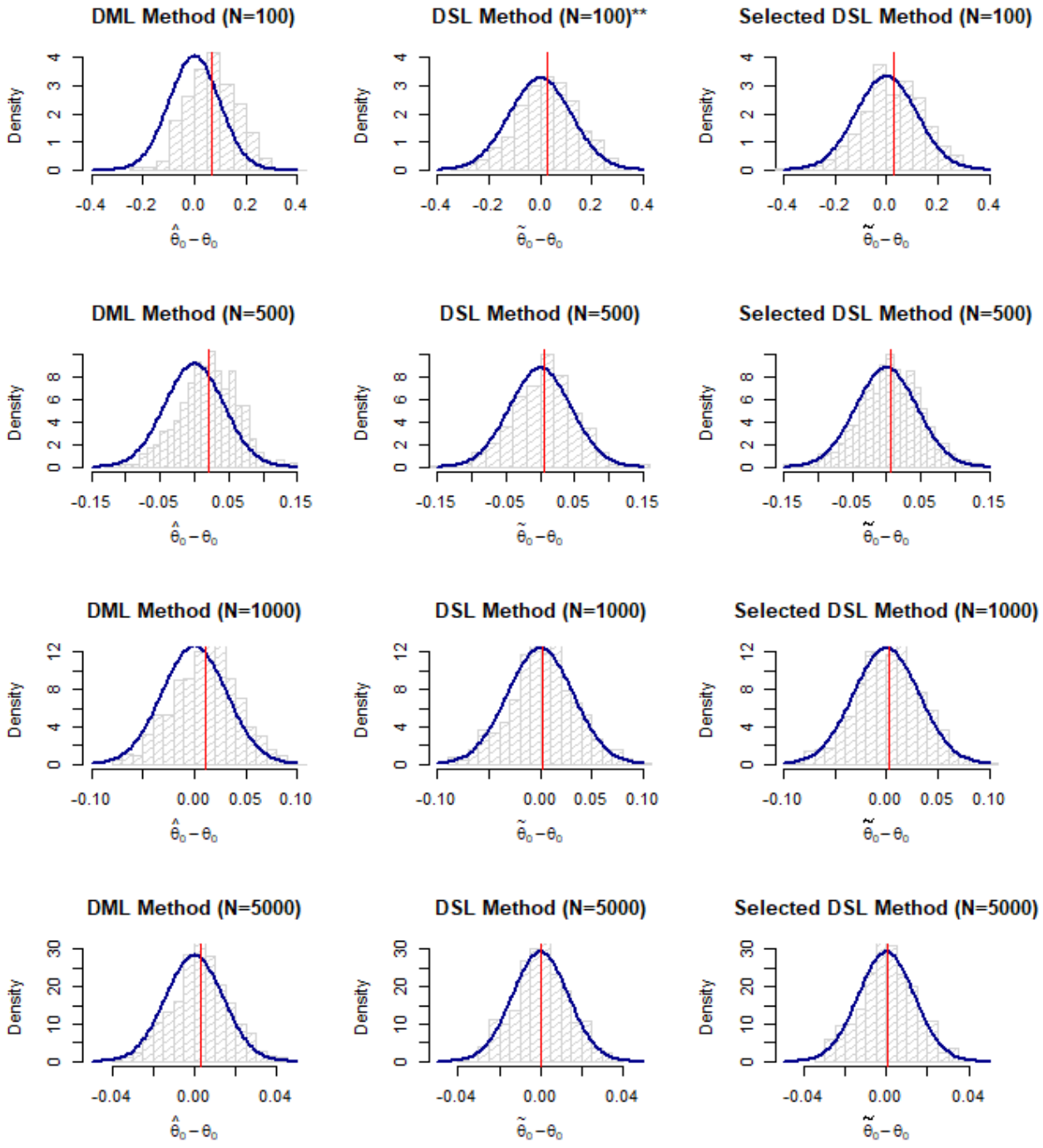


To overcome this issue, a natural solution was to trim the resulted estimates by only 1%, and then calculate the mean and the variance for the trimmed set of estimates. This trick was shown to be very effective in solving this issue, as Figure 9 displays, where the density of $(\tilde{\theta}_0 - \theta_0)$ using the DSL method when the sample size is 100; after trimming, of the density of the estimates is shown to be asymptotically normal just like the others. The trimming trick was employed when the sets of estimated targeted parameters retained over replications are deemed to contain inflated or out of place estimates.

In addition, by observing the red line that represents the bias in Figure 9, it is easy to notice that the DSL method and the selected DSL method have lower bias than that of the DML method, and that bias approaches the 0 as the sample size increases. This was also the case when the number of associated covariates was set to 20, which indicates that even when p was increased to 100, the resulting bias from applying the proposed DSL method is still better than that of the original DML method.

Figure 9

The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 100 when the Treatment Variable is Continuous after Applying the Necessary Trimming



Note: The sign ** indicates that trimming was applied

Table 7

Performance of Each Candidate Learner in the Super Learner Analysis for p = 100 when the Treatment Variable is Continuous

Sample N	Nuisance	Best Candidate	LASSO	GLM	KNN	RF	Boosting
100	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	281	0	54	87	82
		Ratio	0.523	0.004	0.113	0.174	0.185
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	285	0	64	67	88
		Ratio	0.526	0.004	0.126	0.142	0.201
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	395	0	12	29	68
		Ratio	0.721	0.003	0.053	0.054	0.168
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	406	0	10	30	58
		Ratio	0.738	0.004	0.044	0.063	0.149
500	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	495	0	0	2	7
		Ratio	0.861	0.027	0.015	0.021	0.076
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	499	0	0	3	2
		Ratio	0.865	0.028	0.015	0.022	0.070
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	502	0	0	1	1
		Ratio	0.881	0.022	0.023	0.027	0.049
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	503	0	0	1	0
		Ratio	0.876	0.022	0.027	0.029	0.047
1,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.892	0.033	0.009	0.018	0.049
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	503	0	0	1	0
		Ratio	0.894	0.032	0.008	0.017	0.049
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.894	0.023	0.022	0.030	0.032
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	504	0	0	0	0
		Ratio	0.887	0.024	0.022	0.033	0.035
5,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.936	0.027	0.004	0.010	0.023
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	504	0	0	0	0
		Ratio	0.931	0.032	0.004	0.010	0.023
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.920	0.020	0.016	0.030	0.015
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	504	0	0	0	0
		Ratio	0.916	0.024	0.015	0.028	0.017

As I did when analyzing the datasets that involved 20 covariates previously, Table 7 displays the results on the performance of the candidate machine-learning algorithms in estimating the nuisance functions $\hat{g}_0$ and $\hat{m}_0$. As before, the super learner algorithm indicates

that LASSO was the best performing learner given these datasets. Furthermore, Table 8 shows the summary statistics which compares the three methods across different sample sizes. It is not too hard to notice the effect of applying trimming treatment on the DSL method when a sample size of 100 in Table 8. The results shows that when trimming was considered, significant drop in bias and variance was achieved, leading to a much narrower confidence interval in comparison when trimming was not applied.

Table 8

Summary Statistics for Datasets with p = 100 when the Treatment Variable is Continuous

N	Method	Estimates	Bias	Variance	95% CI	
100	DML	0.5684	0.0684	0.0097	0.3757	0.7610
	DSL	0.7961	0.2961	37.5717	-11.2178	12.8101
	DSL *	0.5270	0.0270	0.0147	0.2889	0.7650
	Selected DSL	0.5276	0.0276	0.0142	0.2940	0.7613
500	DML	0.5197	0.0197	0.0019	0.4350	0.6045
	DSL	0.5055	0.0055	0.0020	0.4170	0.5940
	Selected DSL	0.5054	0.0054	0.0020	0.4174	0.5935
1,000	DML	0.5113	0.0113	0.0010	0.4496	0.5730
	DSL	0.5023	0.0023	0.0010	0.4395	0.5652
	Selected DSL	0.5027	0.0027	0.0010	0.4399	0.5654
5,000	DML	0.5034	0.0034	0.0002	0.4759	0.5309
	DSL	0.5003	0.0003	0.0002	0.4737	0.5270
	Selected DSL	0.5007	0.0007	0.0002	0.4740	0.5274

Note: Number of replications is 504.
*1% trimming has been applied.

In general, the results show that both the DSL and the selected DSL methods have an improvement in bias reduction and produce competitive standard errors in comparison with the DML method. The following analysis will consider the number of covariates of 1,000 and 10,000, respectively, and apply trimming when necessary.

Figure 10

The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 1,000 when the Treatment Variable is Continuous

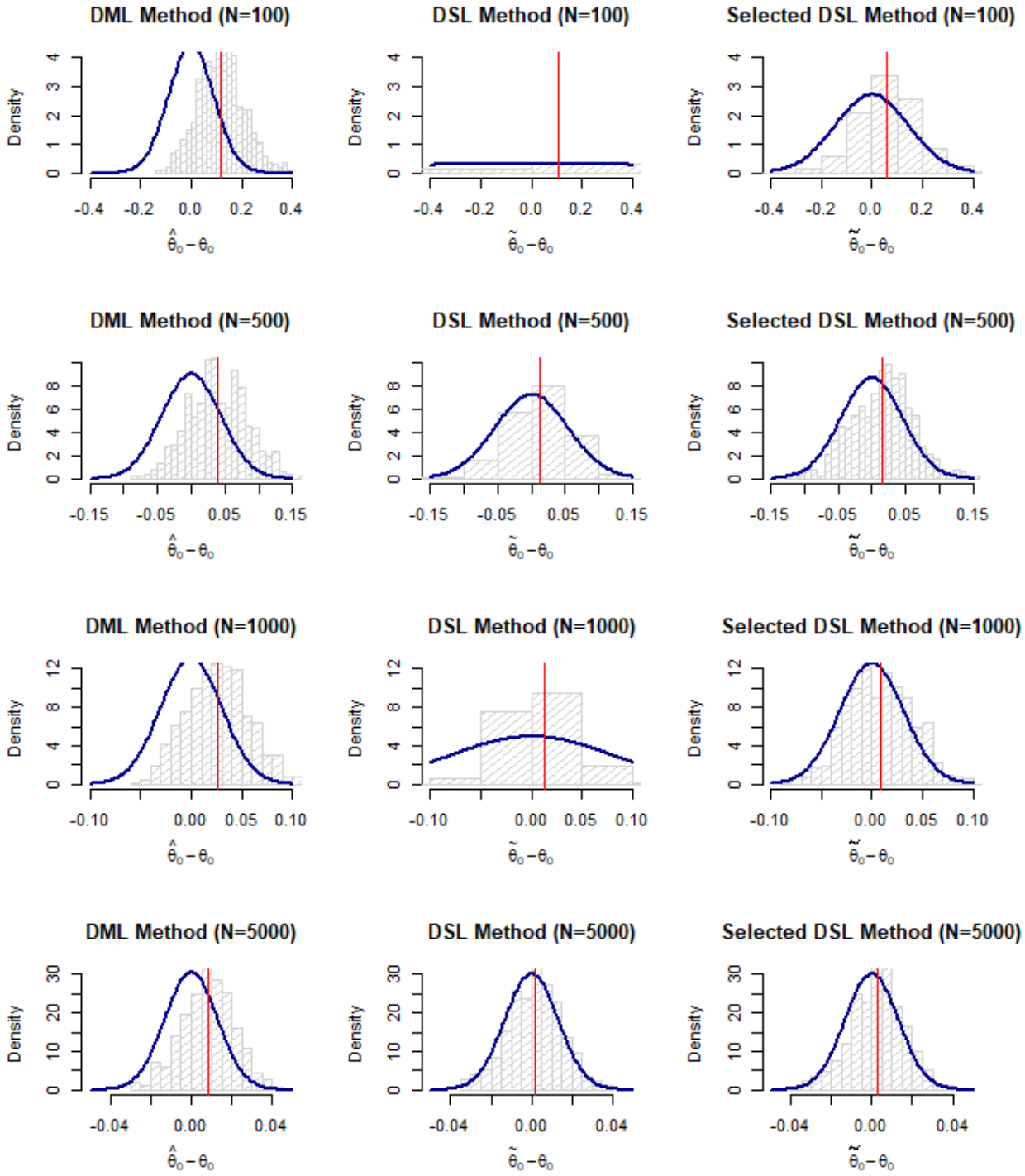
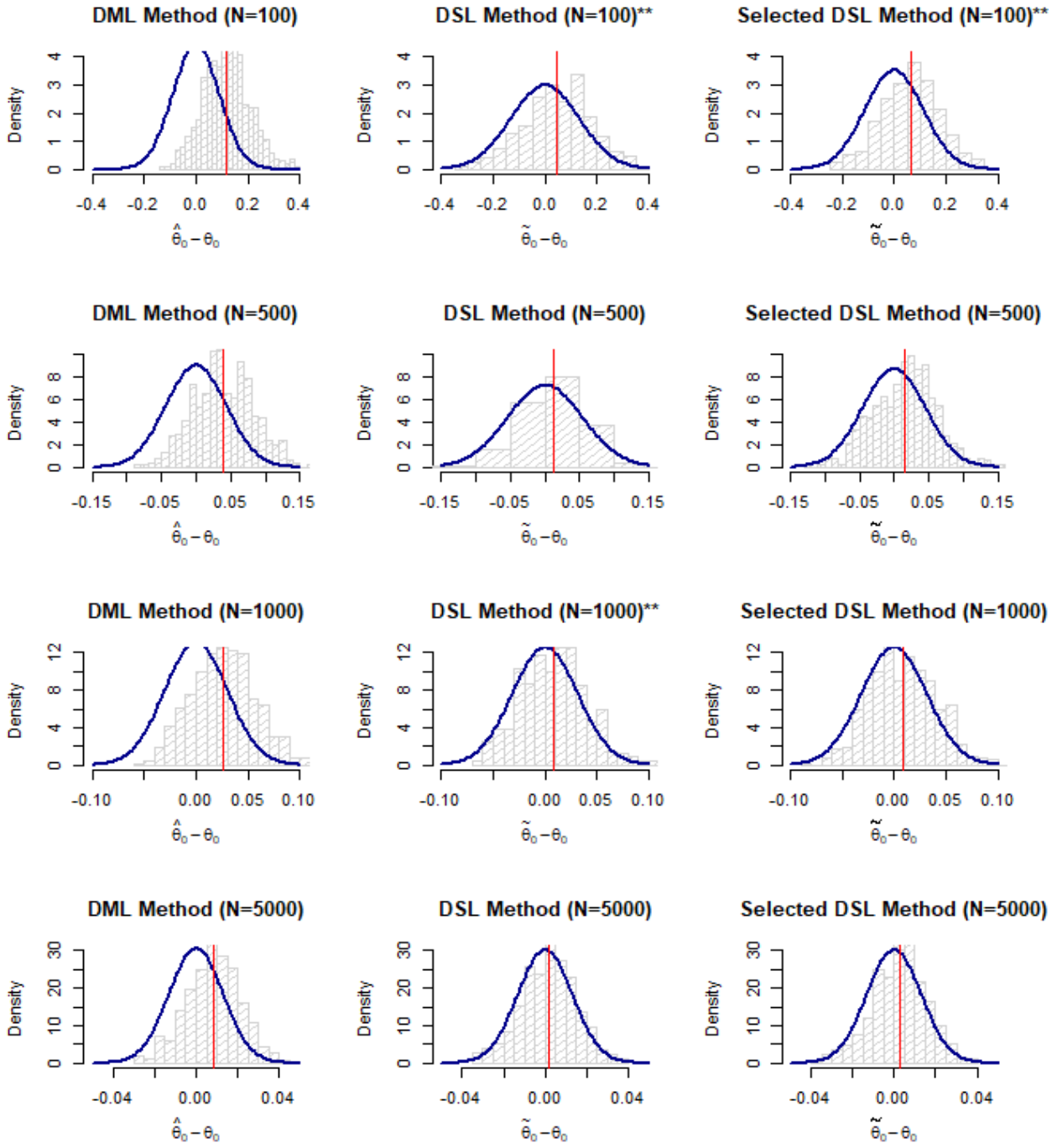


Figure 11

The Distribution of $(\tilde{\theta}_0 - \theta_0)$ *for p = 1,000 when the Treatment Variable is Continuous after Applying the Necessary Trimming*



Note: The sign ** indicates that trimming was applied

Table 9

Performance of Each Candidate Learner in the Super Learner Analysis when p = 1,000

Sample N	Nuisance	Best Candidate	LASSO	GLM	KNN	RF	Boosting
100	$\hat{g}_0(X_{i\in T})$	Frequency	231	0	88	91	94
		Ratio	0.437	0.005	0.168	0.170	0.215
	$\hat{g}_0(X_{i\in Tr})$	Frequency	247	1	73	85	98
		Ratio	0.476	0.008	0.145	0.152	0.218
	$\hat{m}_0(X_{i\in T})$	Frequency	376	6	36	15	71
		Ratio	0.682	0.018	0.080	0.032	0.184
	$\hat{m}_0(X_{i\in Tr})$	Frequency	380	7	22	15	80
		Ratio	0.678	0.022	0.062	0.030	0.205
500	$\hat{g}_0(X_{i\in T})$	Frequency	500	0	0	0	4
		Ratio	0.885	0.001	0.006	0.000	0.108
	$\hat{g}_0(X_{i\in Tr})$	Frequency	498	0	0	0	6
		Ratio	0.884	0.001	0.005	0.001	0.109
	$\hat{m}_0(X_{i\in T})$	Frequency	501	0	0	3	0
		Ratio	0.872	0.001	0.022	0.028	0.078
	$\hat{m}_0(X_{i\in Tr})$	Frequency	500	0	0	1	3
		Ratio	0.880	0.001	0.027	0.018	0.075
1,000	$\hat{g}_0(X_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.934	0.000	0.002	0.000	0.064
	$\hat{g}_0(X_{i\in Tr})$	Frequency	504	0	0	0	0
		Ratio	0.928	0.000	0.002	0.000	0.069
	$\hat{m}_0(X_{i\in T})$	Frequency	503	0	0	0	1
		Ratio	0.906	0.000	0.020	0.022	0.053
	$\hat{m}_0(X_{i\in Tr})$	Frequency	504	0	0	0	0
		Ratio	0.900	0.000	0.020	0.020	0.059
5,000	$\hat{g}_0(X_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.964	0.008	0.000	0.000	0.028
	$\hat{g}_0(X_{i\in Tr})$	Frequency	504	0	0	0	0
		Ratio	0.963	0.009	0.000	0.000	0.028
	$\hat{m}_0(X_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.923	0.008	0.011	0.038	0.020
	$\hat{m}_0(X_{i\in Tr})$	Frequency	504	0	0	0	0
		Ratio	0.919	0.006	0.013	0.037	0.024

Table 10

Summary Statistics for Datasets with p = 1,000 when the Treatment Variable is Continuous

N	Method	Estimates	Bias	Variance	95% CI	
100	DML	0.6186	0.1186	0.0081	0.4426	0.7946
	DSL	0.6056	0.1056	1.7359	-1.9767	3.1880
	DSL*	0.5479	0.0479	0.0177	0.2873	0.8085
	Selected DSL	0.5616	0.0616	0.0215	0.2743	0.8489
	Selected DSL*	0.5649	0.0649	0.0128	0.3435	0.7862
500	DML	0.5393	0.0393	0.0019	0.4533	0.6253
	DSL	0.5119	0.0119	0.0030	0.4049	0.6190
	Selected DSL	0.5159	0.0159	0.0021	0.4265	0.6053
1,000	DML	0.5263	0.0263	0.0009	0.4667	0.5860
	DSL	0.5127	0.0127	0.0064	0.3554	0.6699
	DSL *	0.5081	0.0081	0.0010	0.4459	0.5703
	Selected DSL	0.5090	0.0090	0.0010	0.4468	0.5711
5,000	DML	0.5083	0.0083	0.0002	0.4827	0.5338
	DSL	0.5017	0.0017	0.0002	0.4759	0.5276
	Selected DSL	0.5026	0.0026	0.0002	0.4767	0.5284

Note: Number of replications is 504.

*1% trimming has been applied.

Figure 12

The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 10,000 when the Treatment Variable is Continuous

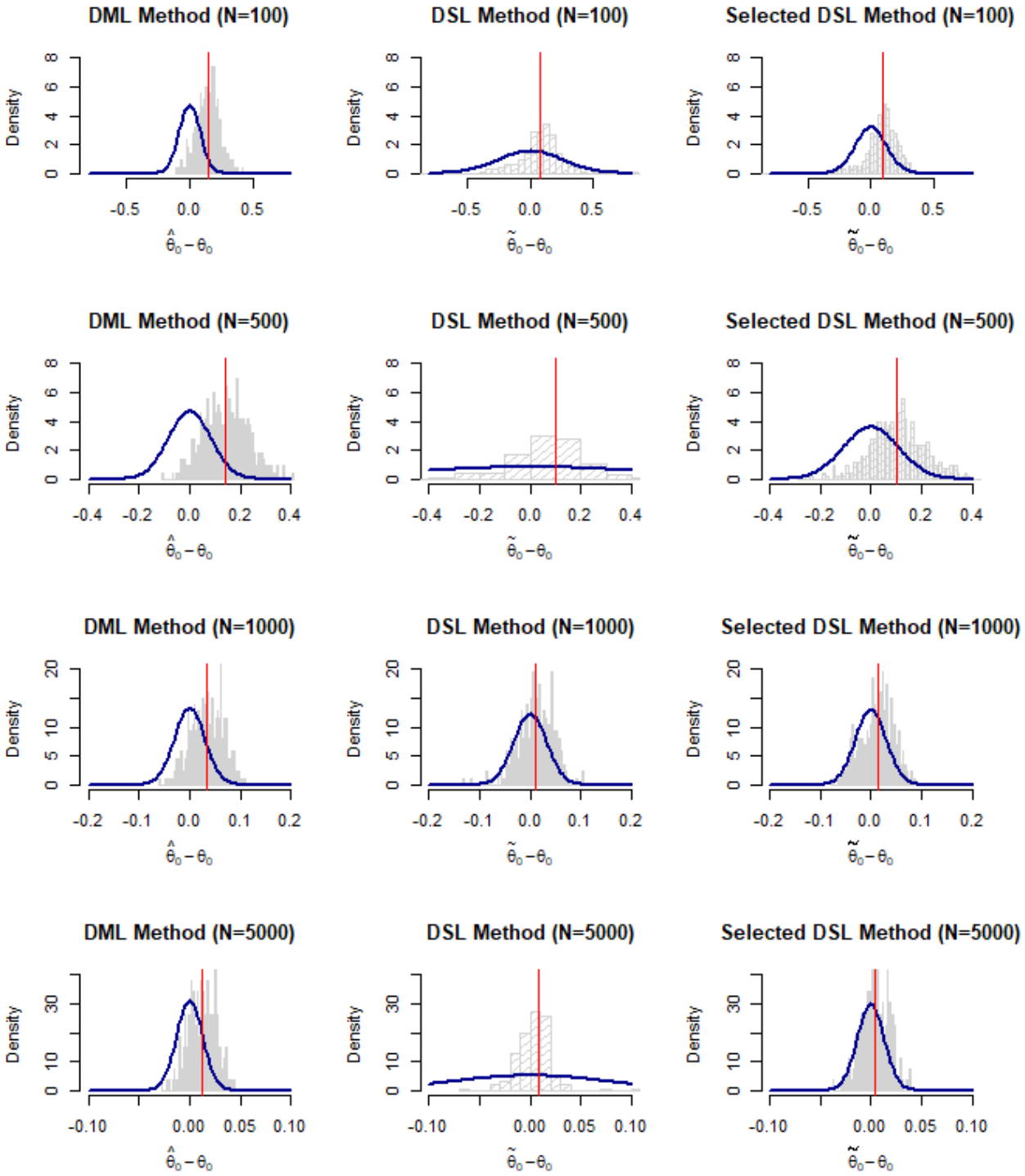
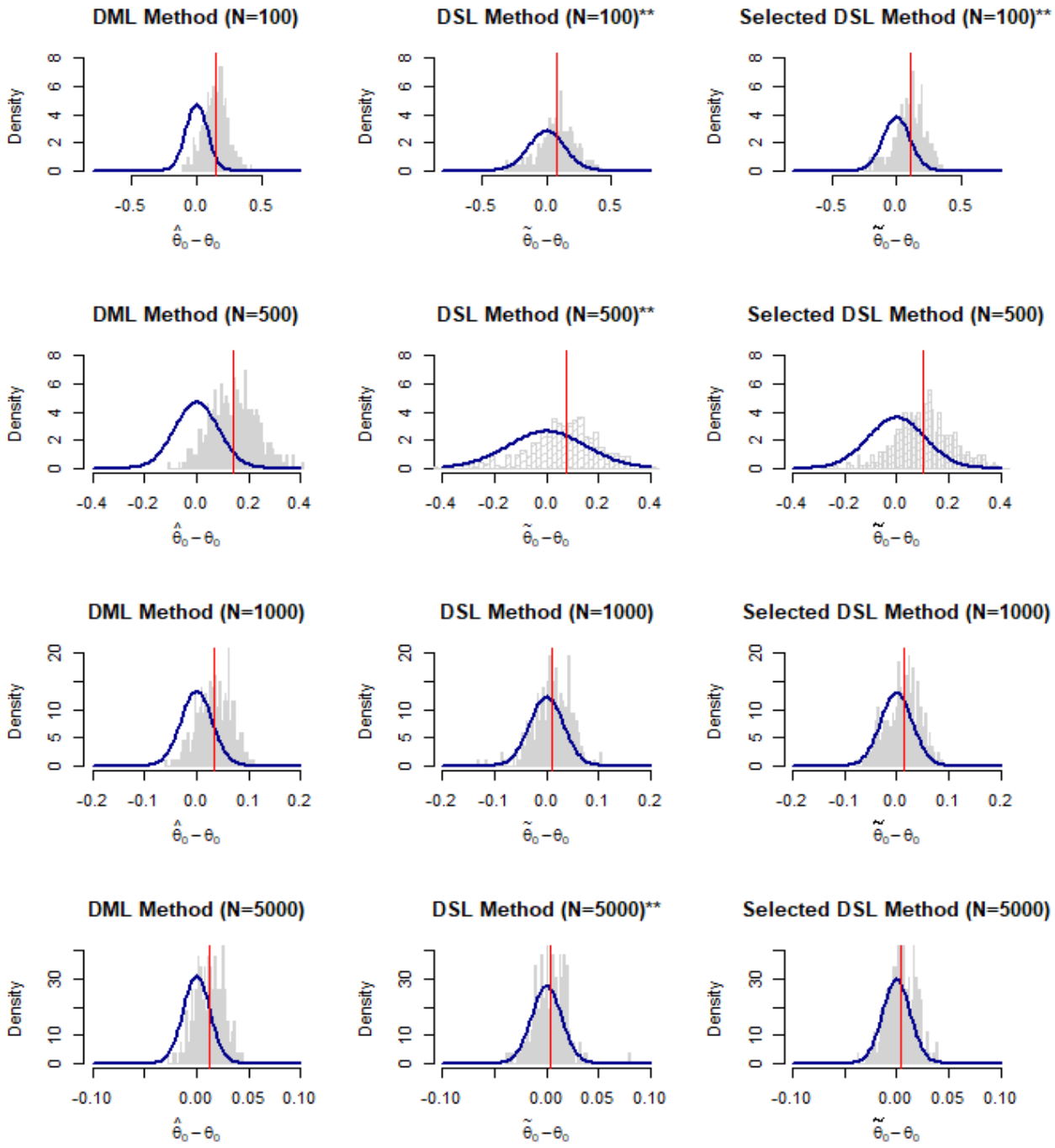


Figure 13

The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 10,000 when the Treatment Variable is Continuous after Applying the Necessary Trimming



Note: The sign ** indicates that trimming was applied.

Table 11

Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 10,000 when the Treatment Variable is Continuous

Sample N	Nuisance	Best Candidate	LASSO	GLM	KNN	RF	Boosting
100	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	173	2	78	92	87
		Ratio	0.378	0.013	0.185	0.209	0.214
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	170	2	90	92	78
		Ratio	0.377	0.011	0.201	0.209	0.202
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	317	9	41	15	50
		Ratio	0.644	0.028	0.110	0.037	0.159
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	299	13	37	16	67
		Ratio	0.622	0.036	0.108	0.037	0.181
500	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	177	3	72	106	74
		Ratio	0.371	0.014	0.178	0.235	0.195
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	166	0	101	87	78
		Ratio	0.387	0.007	0.218	0.195	0.193
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	295	11	40	21	65
		Ratio	0.618	0.031	0.112	0.046	0.179
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	299	7	46	22	58
		Ratio	0.604	0.025	0.122	0.054	0.174
1,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	432	0	0	0	0
		Ratio	0.917	0.000	0.001	0.000	0.082
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	432	0	0	0	0
		Ratio	0.927	0.000	0.000	0.000	0.072
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	431	0	0	1	0
		Ratio	0.907	0.000	0.021	0.012	0.060
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	431	0	0	1	0
		Ratio	0.911	0.000	0.019	0.015	0.055
5,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	264	0	0	0	0
		Ratio	0.963	0.000	0.000	0.000	0.037
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	264	0	0	0	0
		Ratio	0.966	0.000	0.000	0.000	0.034
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	264	0	0	0	0
		Ratio	0.937	0.000	0.011	0.020	0.032
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	264	0	0	0	0
		Ratio	0.945	0.000	0.011	0.016	0.027

Table 12

Summary Statistics for Datasets with p = 10,000 when the Treatment Variable is Continuous

N	Method	Estimates	Bias	Variance	95% CI	
100	DML	0.6418	0.1418	0.0071	0.4768	0.8067
	DSL	0.5768	0.0768	0.0663	0.0722	1.0814
	DSL*	0.5790	0.0790	0.0195	0.3049	0.8530
	Selected DSL	0.5980	0.0980	0.0153	0.3557	0.8403
	Selected DSL*	0.6005	0.1005	0.0109	0.3955	0.8056
500	DML	0.6420	0.1420	0.0072	0.4762	0.8078
	DSL	0.5975	0.0975	0.2230	-0.3280	1.5231
	DSL*	0.5745	0.0745	0.0229	0.2777	0.8714
	Selected DSL	0.6009	0.1009	0.0122	0.3841	0.8177
1,000	DML	0.5351	0.0351	0.0009	0.4761	0.5942
	DSL	0.5106	0.0106	0.0011	0.4468	0.5745
	Selected DSL	0.5134	0.0134	0.0010	0.4530	0.5739
5,000	DML	0.5126	0.0126	0.0002	0.4873	0.5380
	DSL	0.5083	0.0083	0.0059	0.3572	0.6593
	DSL *	0.5037	0.0037	0.0002	0.4752	0.5321
	Selected DSL	0.5045	0.0045	0.0002	0.4782	0.5308

Note: Number of replications is 264 when N is 5,000, and 504 otherwise.
*1% trimming has been applied.

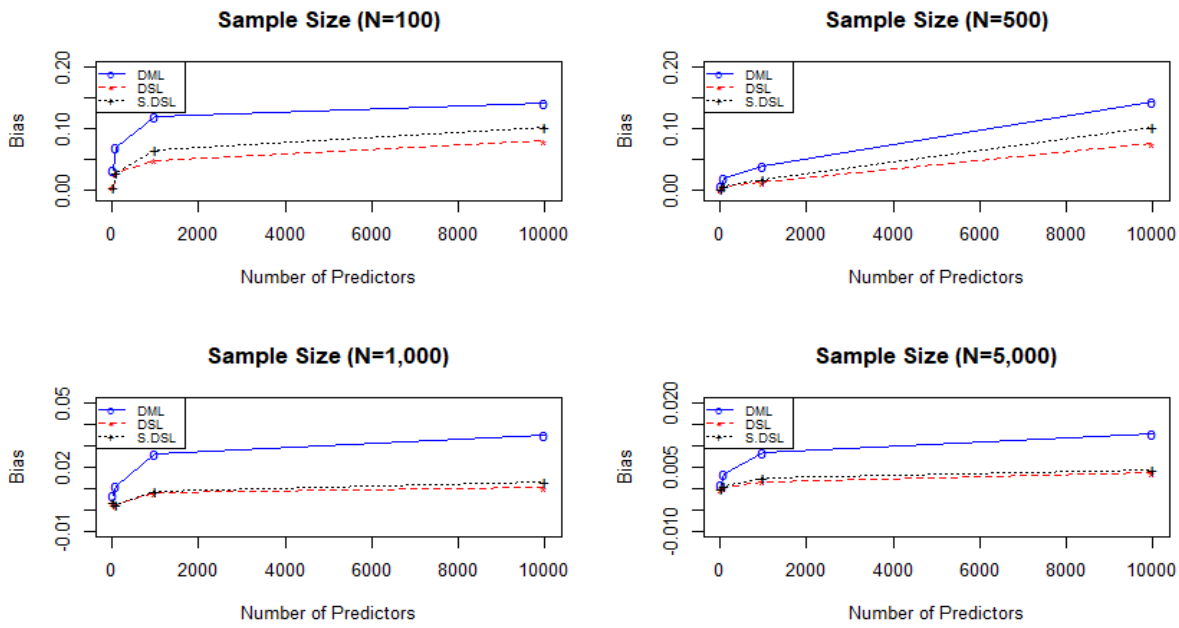
From the results displayed in Figures 10-13 and Tables 11-16, when the three estimation methods are applied on the simulated datasets that involved 1,000 and 10,000 associated number of covariates, respectively, and when the treatment variable is continuous, it can be seen that the DSL methods are producing better estimates in terms of lower bias compared with the DML method. Although it is fair to say that the DML method did not require applying any trimming treatment as for the DSL methods, which could be due to the incorporation of multiple ML algorithms in the DSL method that results in the curse of dimensionality issue.

Now that I have examined the performance of the three estimation methods for different numbers of associated covariates across different sample sizes, it would be interesting to understand how the three methods would perform, in terms of bias and variance, when the

sample size is kept fixed while the number of associated covariates varies. In other words, for a certain sample size, how would the three estimation methods perform as the number of associated covariates increases. The following figure assesses the performance of the three estimation methods for different sample sizes as the number of associated covariates increases.

Figure 14

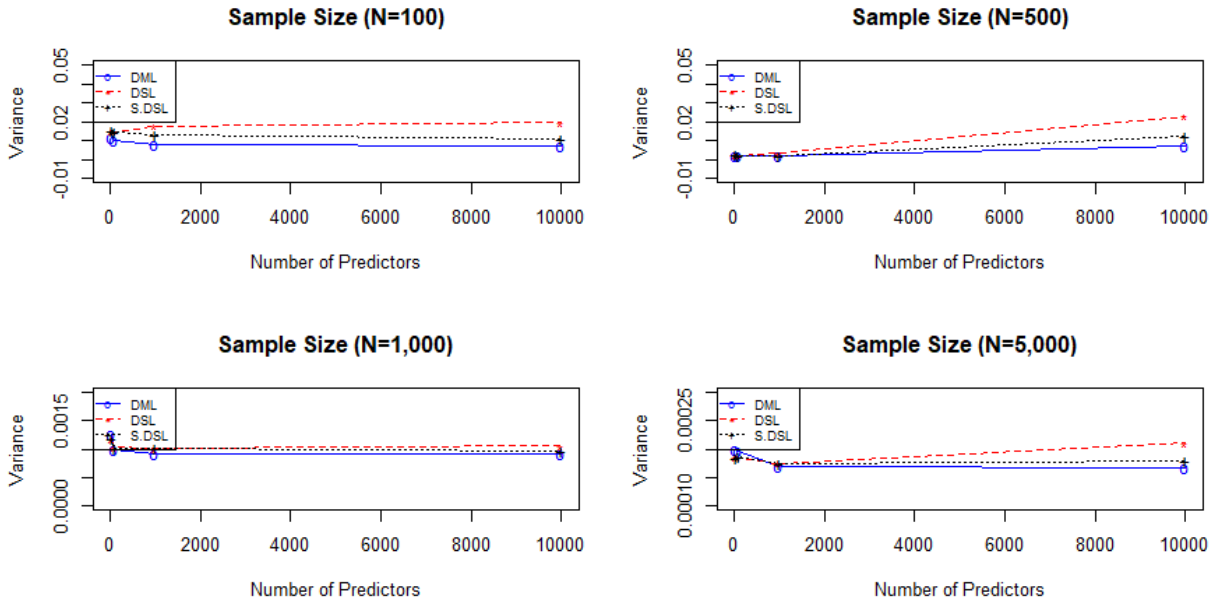
Assessing the Bias across Different Sample Sizes as p Increases when the Treatment Variable is Continuous



From Figure 14, the results show that the DSL method, which is represented via the red line, always produced lower bias as p increased given different sample sizes compared with the DML method (displayed by the blue line) and the selected DSL method (displayed by the black line). On the other hand, the performance of variance was assessed in a similar manner as the bias was in Figure 14, and is displayed in Figure 15.

Figure 15

Assessing the Variance across Different Sample Sizes as p Increases when the Treatment Variable is Continuous



It can be seen in Figure 15 that the resultant estimates, when the DML method was applied, had a slightly lower variance compared with the DSL method, especially for smaller sample sizes. In addition, it is not too hard to notice that as the sample size increases, the estimates' variance of the three methods nearly matches for any given number of associated covariates.

These results obtained in this section indicate that the proposed DSL methods can significantly reduce the bias compared to the DML method without significantly causing inflation in variance, which is considered a very positive and desired conclusion. The next section investigates the performance of the three estimation methods when the treatment variable is considered a binary.

Simulation Analysis for Binary Case Treatment

After performing the DML and the DSL methods on various datasets that considered the cases having a continuous treatment variable discussed in the previous section, it is appropriate to consider the cases where the treatment variable is binary in this part of the dissertation. The analysis in this section was carried out in a similar manner to that in the previous section. There were 16 different settings for the simulated datasets that were created using the data-generating process explained in the simulation scheme section where the treatment variable was binary. In addition, a grid of values for the number of associated covariates and the sample sizes considered in this section was the same as in the previous section when the data simulated used a continuous treatment variable.

When analyzing these simulated datasets, the DML and DSL methods were implemented using two different estimation equations for the targeted parameter. The first analysis used the estimator introduced in equation (57), and the results and outputs are presented in this section. The second analysis (results are included in Appendix B), on the other hand, incorporated the estimation equations (27) and (44) as previously implemented in continuous case treatments. Furthermore, the use of a trimming treatment on the resulted estimates were incorporated, when necessary, just as previously done. The following figures and tables illustrate the results of the investigation of the three methods using 504 replications when the number of associated covariates was fixed at 20 for the case of binary treatment variables.

Figure 16

The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 20 when the Treatment Variable is Binary

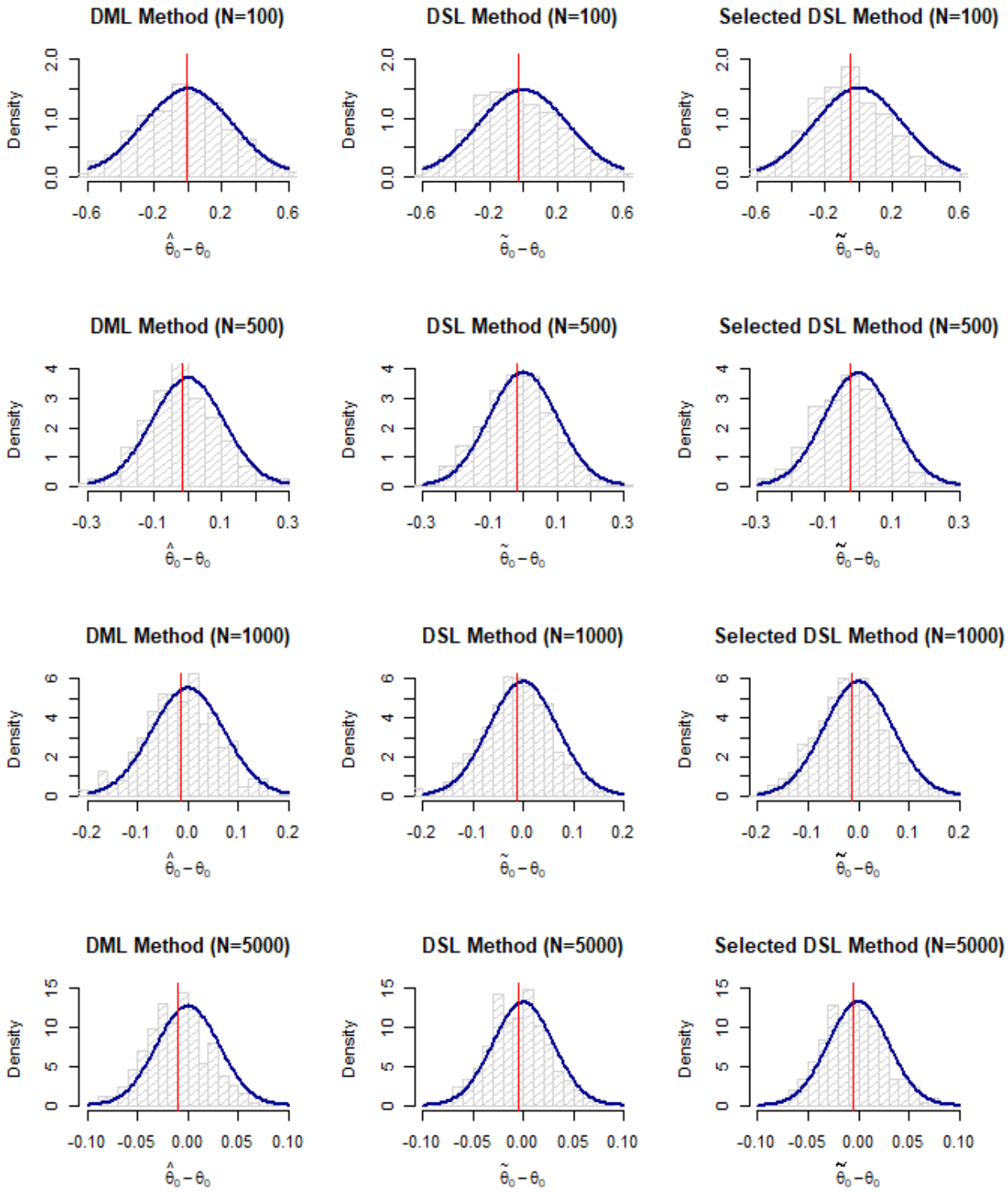


Table 13

Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 20 when the Treatment Variable is Binary

Sample	Nuisance	Best Candidate	LASSO	GLM	KNN	RF	Boosting
N=100	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	193	14	154	119	24
		Ratio	0.362	0.074	0.274	0.204	0.086
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	221	9	133	116	25
		Ratio	0.405	0.064	0.238	0.215	0.078
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	108	6	58	105	227
		Ratio	0.190	0.075	0.118	0.202	0.415
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	116	3	57	108	220
		Ratio	0.202	0.080	0.110	0.207	0.402
N=500	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	394	13	25	71	1
		Ratio	0.642	0.086	0.097	0.140	0.034
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	388	19	23	72	2
		Ratio	0.635	0.098	0.090	0.144	0.034
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	450	12	9	33	0
		Ratio	0.738	0.083	0.052	0.080	0.047
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	449	15	5	33	2
		Ratio	0.742	0.079	0.046	0.088	0.044
N=1,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	445	19	5	34	1
		Ratio	0.715	0.096	0.055	0.106	0.028
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	457	17	4	26	0
		Ratio	0.745	0.087	0.050	0.090	0.028
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	486	8	1	9	0
		Ratio	0.808	0.087	0.029	0.050	0.026
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	484	10	0	10	0
		Ratio	0.810	0.082	0.028	0.044	0.036
N=5,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	489	15	0	0	0
		Ratio	0.829	0.089	0.017	0.046	0.019
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	481	22	0	1	0
		Ratio	0.818	0.106	0.021	0.042	0.013
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	496	8	0	0	0
		Ratio	0.870	0.080	0.014	0.017	0.019
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	499	5	0	0	0
		Ratio	0.865	0.082	0.014	0.020	0.019

The results presented in Figure 16 show the normal distribution of the estimated parameters' density, depicted via histograms, when the process is replicated 504 times. The densities of these estimates aligned with the outputs presented in Table 14. In addition, Table 13

assessed the performance of the candidate ML algorithms that were incorporated in the DSL method, where the SL function suggested that LASSO was the best performing algorithm in terms of ratio and frequency. However, when the sample size was set to 100, the boosting algorithms were shown to be more powerful in estimating the nuisance function $\hat{m}_0(\boldsymbol{X})$.

Table 14 shows the overall performance of the three methods in estimating the targeted parameter when the analysis was replicated 504 times. Based on the findings displayed in Table 14, the three estimation methods were shown to be competitive in terms of bias and variance with slight superiority to the DML method in terms of bias when the sample size was set to 100.

Table 14

Summary Statistics for Datasets with p = 20 when the Treatment Variable is Binary

N	Method	Estimates	Bias	Variance	95% CI	
100	DML	0.4937	0.0063	0.0711	-0.0288	1.0162
	DSL	0.4744	0.0256	0.0724	-0.0528	1.0016
	Selected DSL	0.4530	0.0470	0.0694	-0.0633	0.9692
500	DML	0.4814	0.0186	0.0116	0.2704	0.6924
	DSL	0.4817	0.0183	0.0105	0.2808	0.6825
	Selected DSL	0.4772	0.0228	0.0106	0.2755	0.6789
1,000	DML	0.4869	0.0131	0.0052	0.3453	0.6284
	DSL	0.4869	0.0131	0.0046	0.3536	0.6202
	Selected DSL	0.4854	0.0146	0.0046	0.3519	0.6190
5,000	DML	0.4900	0.0100	0.0010	0.4289	0.5512
	DSL	0.4953	0.0047	0.0009	0.4363	0.5543
	Selected DSL	0.4946	0.0054	0.0009	0.4358	0.5533

Note. Number of replications is 504.

After investigating the estimated targeted parameter of the binary treatment variable when the number of associated covariates was set to 20, I followed the same steps to investigate the performance of the three methods when the number of associated covariates was fixed at 100.

To follow-up on investigating the performance of the three estimation methods, the analysis was replicated 504 times. The results are displayed in Figures 17 and 18 and Tables 15 and 16.

Figure 17

The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 100 when the Treatment Variable is Binary

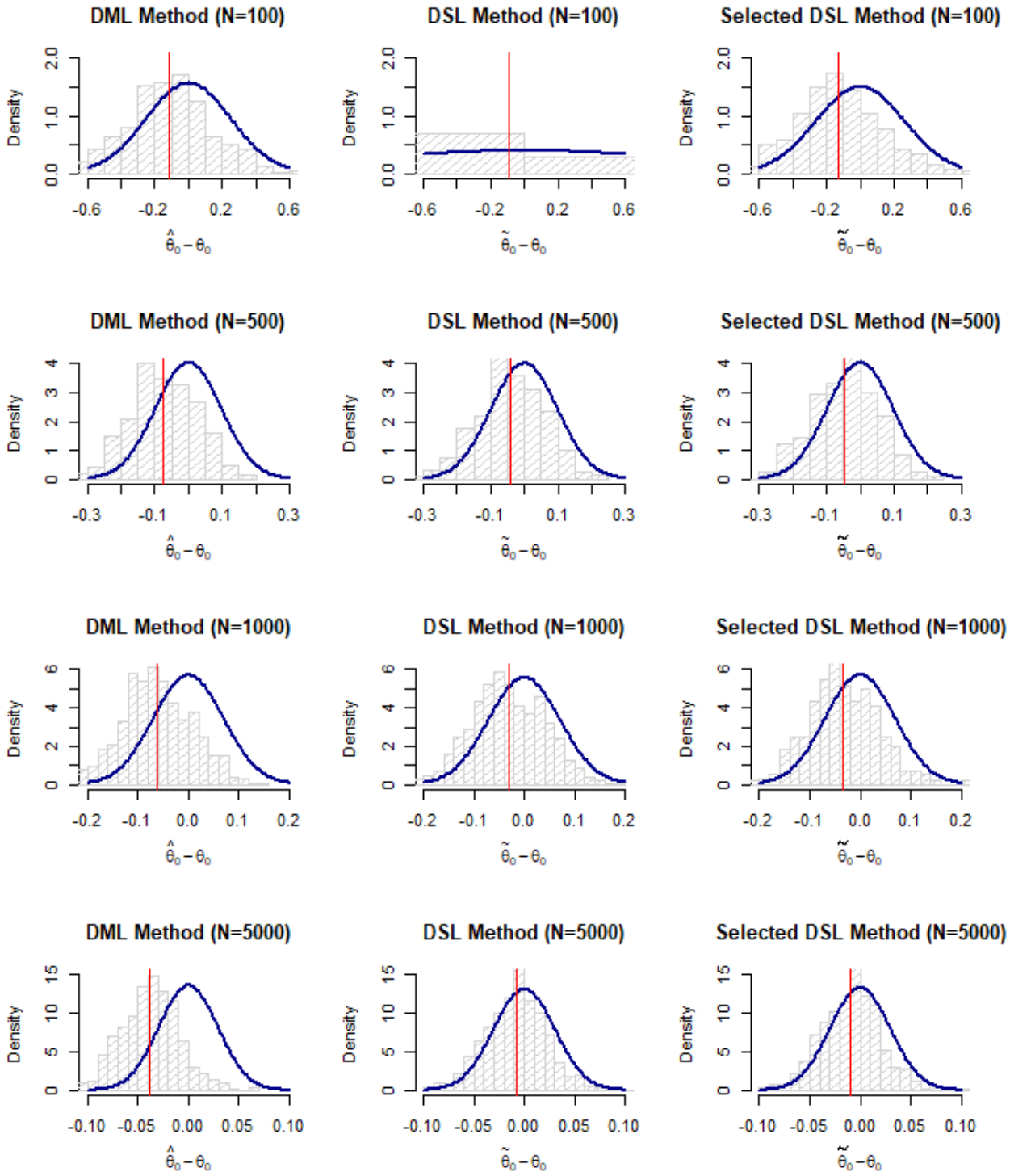
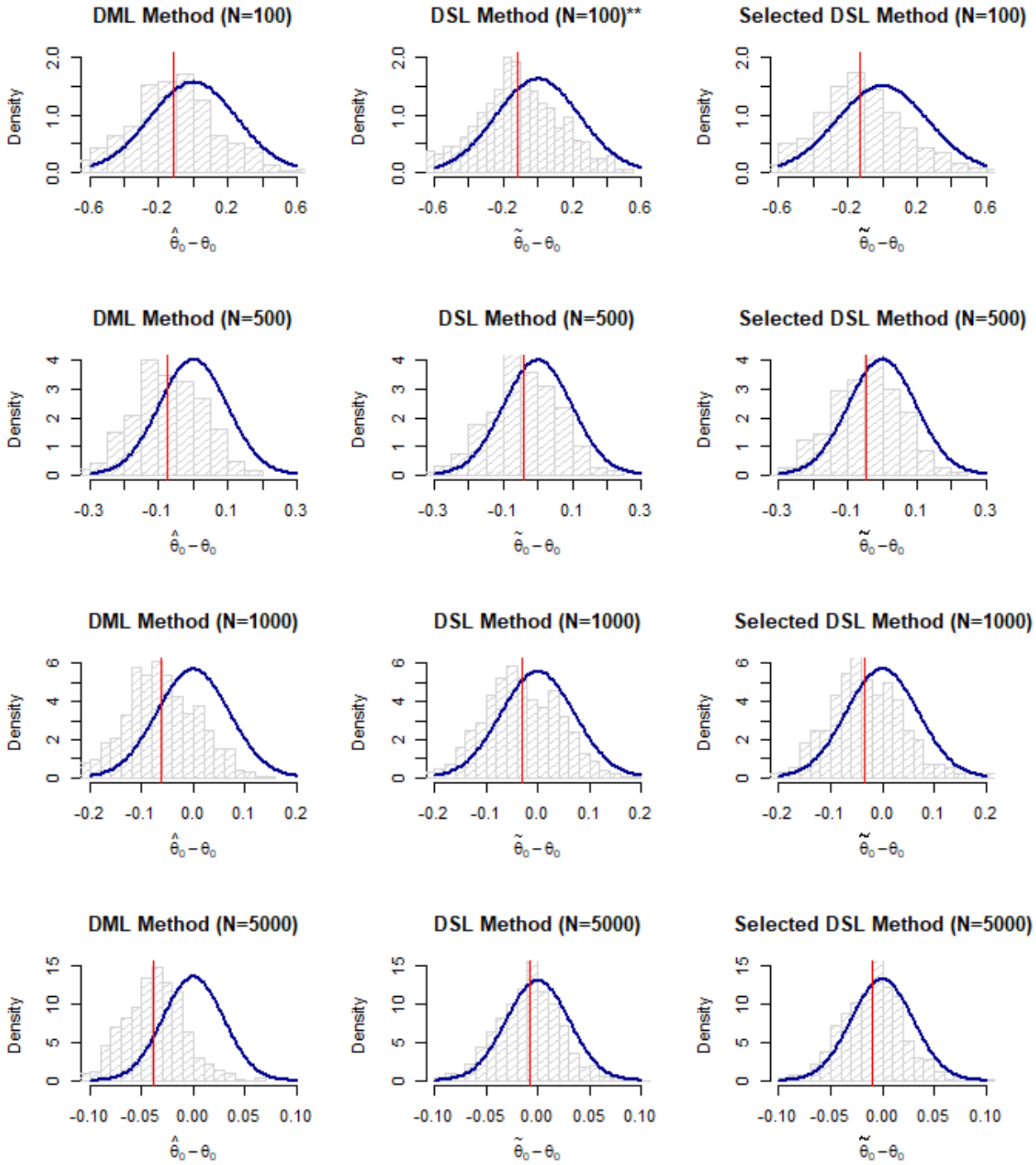


Figure 18

The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 100 when the Treatment Variable is Binary after Applying the Necessary Trimming



Note: The sign ** indicates that trimming was applied.

Table 15

P*erformance of Each Candidate Learner in the Super earner Analysis for Datasets with p = 100 when the Treatment Variable is Binary*

Sample N	Nuisance	Best Candidate	LASSO	GLM	KNN	RF	Boosting
N=100	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	177	0	130	150	47
		Ratio	0.345	0.004	0.235	0.286	0.129
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	183	0	124	145	52
		Ratio	0.350	0.005	0.252	0.265	0.128
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	108	1	58	80	257
		Ratio	0.198	0.065	0.115	0.162	0.460
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	113	3	65	75	248
		Ratio	0.197	0.064	0.126	0.155	0.458
N=500	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	363	0	12	120	9
		Ratio	0.625	0.031	0.071	0.213	0.060
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	347	0	19	134	4
		Ratio	0.596	0.030	0.082	0.235	0.057
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	486	0	6	11	1
		Ratio	0.846	0.023	0.035	0.040	0.056
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	487	0	1	13	3
		Ratio	0.851	0.022	0.028	0.042	0.058
N=1,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	446	0	1	57	0
		Ratio	0.754	0.036	0.042	0.127	0.041
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	454	0	0	49	1
		Ratio	0.761	0.036	0.037	0.123	0.043
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.909	0.027	0.012	0.010	0.042
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	501	0	0	3	0
		Ratio	0.890	0.035	0.015	0.016	0.044
N=5,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	503	0	0	1	0
		Ratio	0.897	0.047	0.009	0.026	0.021
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	503	0	0	1	0
		Ratio	0.896	0.045	0.011	0.029	0.019
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.941	0.030	0.003	0.002	0.024
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	504	0	0	0	0
		Ratio	0.934	0.035	0.004	0.002	0.025

Table 16

Summary Statistics for Datasets with p = 100 when the Treatment Variable is Binary

N	Method	Estimates	Bias	Variance	95% CI	
100	DML	0.3812	0.1188	0.0646	-0.1170	0.8793
	DSL	0.4111	0.0889	0.9901	-1.5391	2.3614
	DSL *	0.3784	0.1216	0.0599	-0.1014	0.8581
	Selected DSL	0.3743	0.1257	0.0704	-0.1456	0.8942
500	DML	0.4258	0.0742	0.0097	0.2324	0.6191
	DSL	0.4587	0.0413	0.0099	0.2641	0.6533
	Selected DSL	0.4524	0.0476	0.0097	0.2597	0.6451
1,000	DML	0.4370	0.0630	0.0049	0.3000	0.5741
	DSL	0.4683	0.0317	0.0051	0.3280	0.6087
	Selected DSL	0.4660	0.0340	0.0048	0.3298	0.6022
5,000	DML	0.4618	0.0382	0.0009	0.4045	0.5192
	DSL	0.4920	0.0080	0.0009	0.4325	0.5516
	Selected DSL	0.4910	0.0090	0.0009	0.4323	0.5497

Note. Number of replications is 504.

*1% trimming has been applied.

Judging by the results displayed in Figures 17 and 18, it became clear that the DSL methods had better alignment with the theoretical distribution compared with the DML method. In particular, the center of the DML estimates tended to shift to the left as the sample size increased. In addition, these two figures show that when the sample size was set to 100, the use of trimming on the DSL estimates improved the estimates' density.

Furthermore, Table 15 shows similar results in assessing the performance of the five-candidate ML algorithms incorporated in the DSL method as in the previous analysis when p was set to 20. In addition, Table 16 shows that the DSL method resulted in bias reduction compared to the other methods. Table 16 also shows that the bias using the DSL method approaches the 0 as the sample size increase. The same applies to the selected DSL and the DML methods, respectively. On the other hand, the variance of the estimates across the three methods remained competitive.

The outputs presented next were produced to investigate the performance of the three estimation methods in estimating the targeted parameter in the presence of 1,000 and 10,000 associated covariates, respectively, across different sample sizes. The analysis of results is organized and presented in the same manner as done previously in this dissertation.

Figure 19

The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 1,000 when the Treatment Variable is Binary

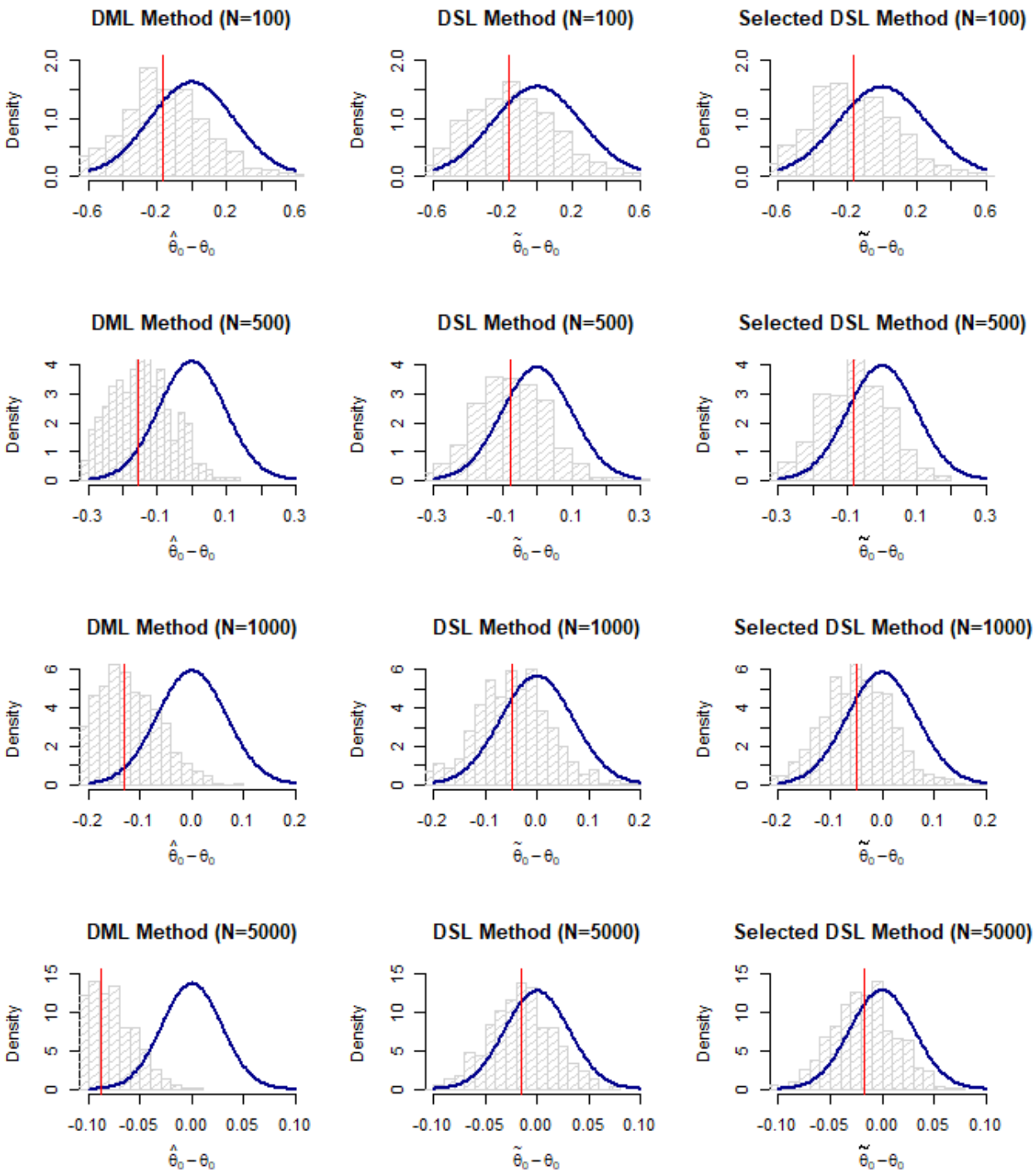


Table 17

Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 1,000 when the Treatment Variable is Binary

Sample N	Nuisance	Best Candidate	LASSO	GLM	KNN	RF	Boosting
100	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	141	0	97	184	82
		Ratio	0.276	0.004	0.211	0.337	0.171
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	141	0	103	196	64
		Ratio	0.277	0.004	0.208	0.358	0.153
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	73	2	77	31	321
		Ratio	0.128	0.072	0.152	0.061	0.588
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	70	2	65	41	326
		Ratio	0.132	0.067	0.132	0.075	0.594
500	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	289	0	22	174	19
		Ratio	0.515	0.001	0.088	0.298	0.099
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	300	0	18	172	14
		Ratio	0.543	0.001	0.076	0.287	0.093
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	481	0	0	13	10
		Ratio	0.824	0.020	0.022	0.026	0.108
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	476	0	1	16	11
		Ratio	0.827	0.018	0.023	0.036	0.095
1,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	449	0	1	50	4
		Ratio	0.793	0.000	0.037	0.088	0.081
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	442	0	2	58	2
		Ratio	0.773	0.000	0.036	0.108	0.084
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.916	0.008	0.005	0.001	0.070
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	503	0	1	0	0
		Ratio	0.918	0.009	0.005	0.001	0.067
5,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	503	0	0	1	0
		Ratio	0.960	0.010	0.002	0.002	0.026
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	503	0	0	1	0
		Ratio	0.954	0.010	0.004	0.002	0.030
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.950	0.004	0.001	0.000	0.045
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	503	0	0	1	0
		Ratio	0.946	0.005	0.001	0.001	0.047

Table 18

Summary Statistics for Datasets with p = 1,000 when the Treatment Variable is Binary

N	Method	Estimates	Bias	Variance	95% CI	
100	DML	0.3286	0.1714	0.0602	-0.1523	0.8094
	DSL	0.3363	0.1637	0.0667	-0.1699	0.8426
	Selected DSL	0.3303	0.1697	0.0670	-0.1769	0.8375
500	DML	0.3427	0.1573	0.0093	0.1532	0.5321
	DSL	0.4230	0.0770	0.0104	0.2233	0.6227
	Selected DSL	0.4169	0.0831	0.0100	0.2208	0.6130
1,000	DML	0.3689	0.1311	0.0045	0.2374	0.5003
	DSL	0.4530	0.0470	0.0049	0.3151	0.5908
	Selected DSL	0.4508	0.0492	0.0047	0.3170	0.5846
5,000	DML	0.4127	0.0873	0.0008	0.3560	0.4693
	DSL	0.4843	0.0157	0.0010	0.4232	0.5454
	Selected DSL	0.4821	0.0179	0.0010	0.4216	0.5425

Note. Number of replications is 504.

Figure 20

The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 10,000 when the Treatment Variable is Binary

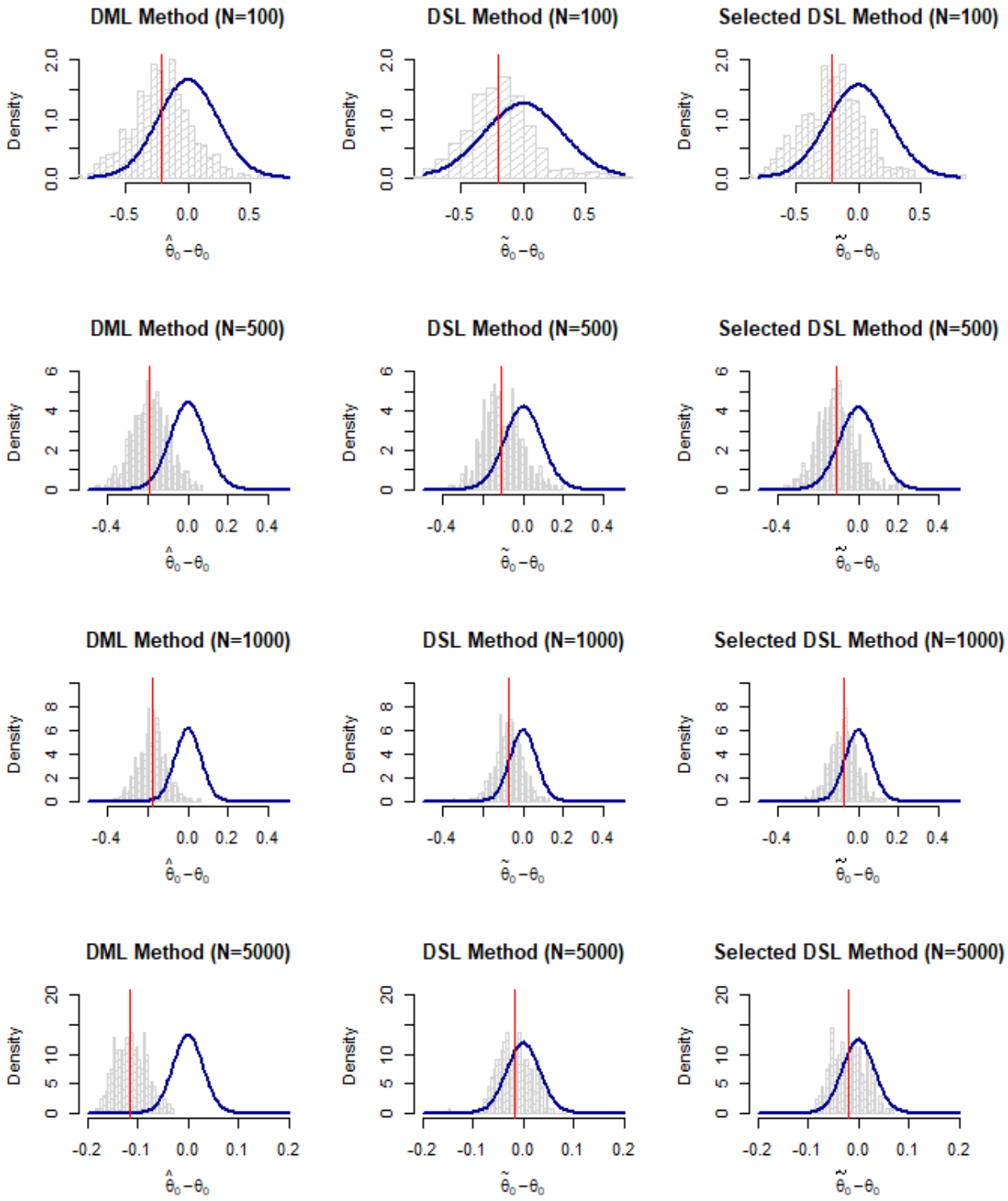


Table 19

Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 10,000 when the Treatment Variable is Binary

Sample N	Nuisance	Best Candidate	LASSO	GLM	KNN	RF	Boosting
100	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	139	0	100	189	76
		Ratio	0.269	0.005	0.202	0.356	0.168
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	122	0	118	195	69
		Ratio	0.234	0.005	0.230	0.371	0.159
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	49	1	74	24	356
		Ratio	0.102	0.074	0.141	0.039	0.645
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	52	1	94	18	339
		Ratio	0.098	0.076	0.177	0.034	0.615
500	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	251	0	8	219	26
		Ratio	0.435	0.001	0.085	0.363	0.116
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	246	0	23	210	25
		Ratio	0.439	0.001	0.094	0.362	0.105
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	453	0	3	24	24
		Ratio	0.771	0.023	0.029	0.048	0.129
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	454	0	2	24	24
		Ratio	0.771	0.022	0.027	0.043	0.137
1,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	422	0	0	70	12
		Ratio	0.720	0.000	0.049	0.131	0.100
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	436	0	1	62	5
		Ratio	0.745	0.000	0.045	0.115	0.094
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.878	0.009	0.006	0.001	0.107
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	503	0	0	0	1
		Ratio	0.873	0.009	0.005	0.001	0.112
5,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	264	0	0	0	0
		Ratio	0.960	0.000	0.001	0.000	0.039
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	264	0	0	0	0
		Ratio	0.967	0.000	0.002	0.000	0.031
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	264	0	0	0	0
		Ratio	0.924	0.002	0.000	0.000	0.073
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	264	0	0	0	0
		Ratio	0.928	0.002	0.000	0.000	0.070

Table 20

Summary Statistics for Datasets with p = 10,000 when the Treatment Variable is Binary

N	Method	Estimates	Bias	Variance	95% CI	
100	DML	0.2902	0.2098	0.0573	-0.1791	0.7596
	DSL	0.3014	0.1986	0.0996	-0.3172	0.9200
	Selected DSL	0.2920	0.2080	0.0637	-0.2027	0.7866
500	DML	0.3092	0.1908	0.0081	0.1329	0.4855
	DSL	0.3928	0.1072	0.0089	0.2084	0.5772
	Selected DSL	0.3909	0.1091	0.0091	0.2035	0.5782
1,000	DML	0.3194	0.1806	0.0042	0.1928	0.4460
	DSL	0.4286	0.0714	0.0043	0.3000	0.5573
	Selected DSL	0.4242	0.0758	0.0043	0.2959	0.5525
5,000	DML	0.3851	0.1149	0.0009	0.3260	0.4442
	DSL	0.4838	0.0162	0.0011	0.4182	0.5493
	Selected DSL	0.4795	0.0205	0.0010	0.4171	0.5418

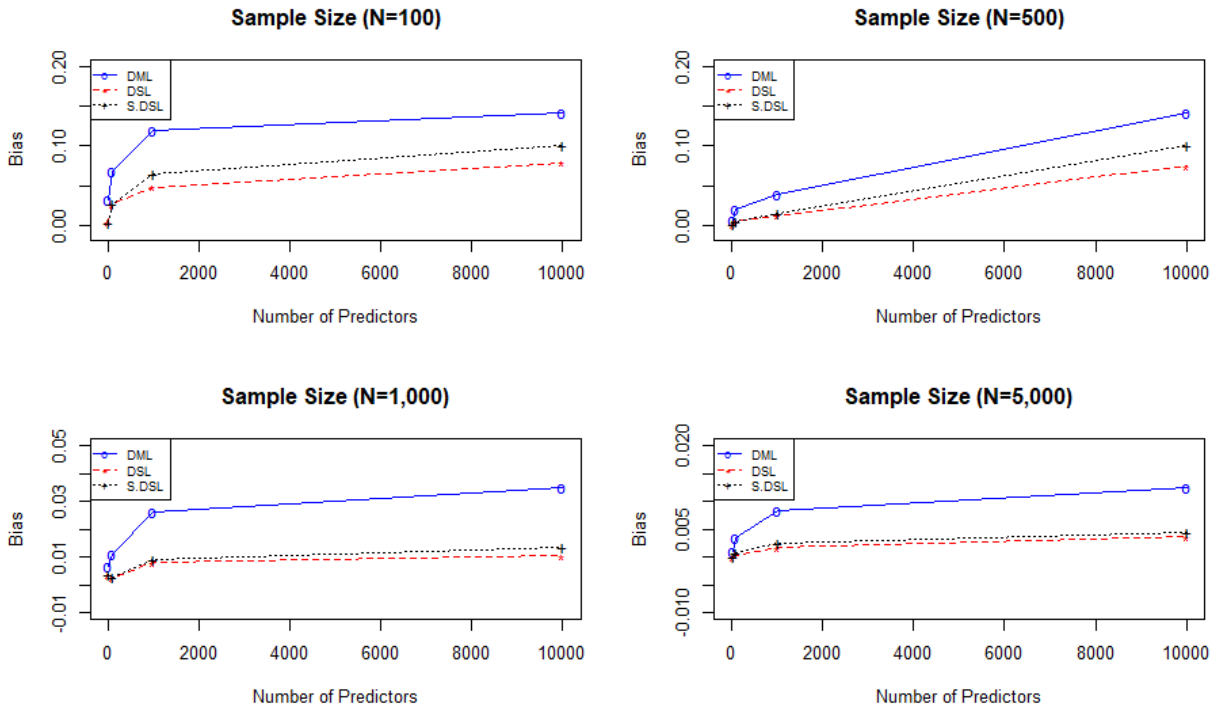
Note. Number of replications is 264 when N equals to 5,000, and 504 otherwise.

In reviewing the simulation analysis results (displayed in Figures 19 and 20 and Tables 17-20) that investigated the performance of the three methods used in estimating the effect of a binary treatment variable in the presence of 1,000 and 10,000 covariates, respectively, across different sample sizes, few interesting findings can be concluded. Unlike in the previous analysis for datasets with continuous treatment variables, the DML method failed to produce valid confidence intervals when the dimension of the datasets increased for datasets with binary treatment variables. More specifically, when replicating the DML method over 500 times, there were 4 occasions where the associated confidence intervals failed to include the true value of the targeted parameter: once, when p = 1,000 and N = 5,000; and three times, when p = 10,000 and N = {500, 1,000, 5,000}. These shortcomings of the DML method, at least when incorporating Random Forests in these datasets, can be observed in Tables 18 and 20. These results are also supported by Figures 19 and 20, where the densities of the DML estimates shifted toward the left as the sample size increased and were not centered around the 0 as they should.

On the other hand, the DSL methods showed great performance in terms of bias reduction while providing valid confidence intervals where the true value of the targeted parameter was always contained within the confidence limits. Furthermore, the densities of the estimated parameters (Figures 19 and 20) clearly showed that the DSL methods had better performance in terms of bias and distribution of estimates' densities compared with the DML method. The following figure compares the three estimation methods in term of bias as the number of associated covariates increased across different sample sizes.

Figure 21

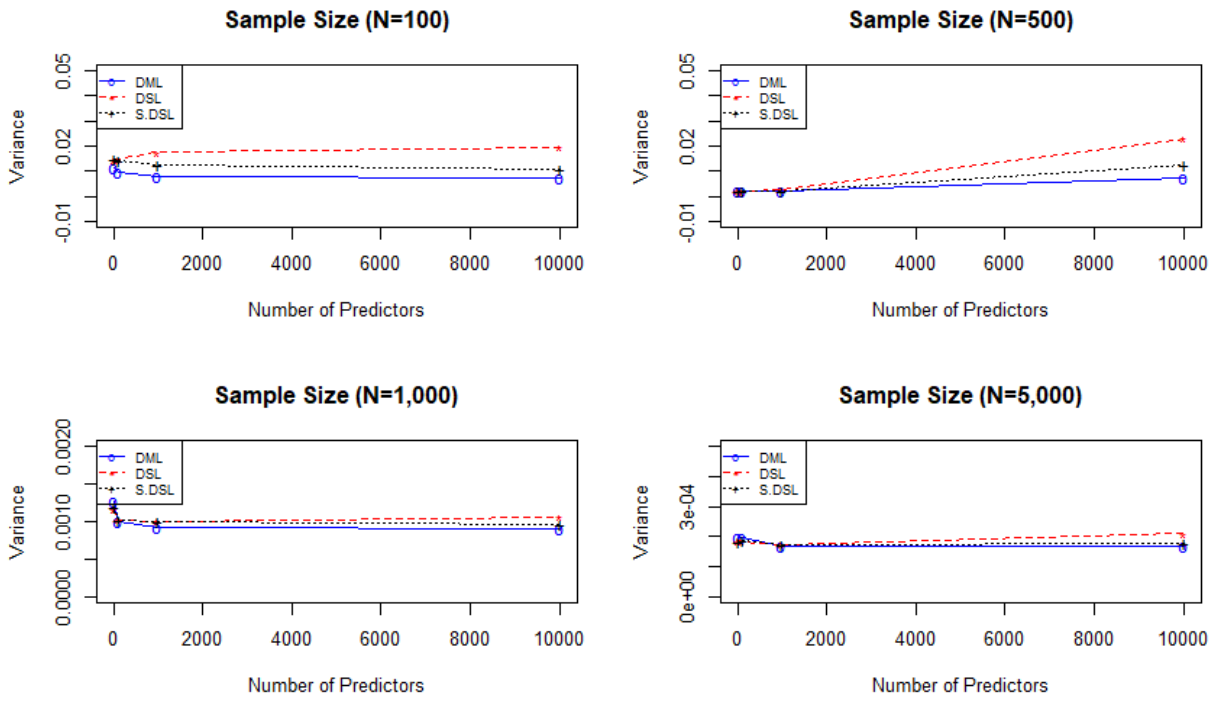
Assessing the Bias of the Three Methods Across Different Sample Sizes as p Increases when the Treatment Variable is Binary



The results presented in Figure 21 show that the DSL methods (DSL in red line and selected DSL in black line) produced lower bias in every setting compared to the DML method (the blue line) by a fair margin. These results aligned with the previous analysis when the treatment variable was considered continuous. Furthermore, Figure 22 shows that the associated variance of the estimated targeted parameter using the three methods almost matched as the sample size increased.

Figure 22

Assessing the Variance of the Three Methods across Different Sample Sizes as p Increases when the Treatment Variable is Binary



Implementing the Double Super Learner Function Using R package

Now that the DML, DSL, and the selected DSL methods have been introduced, it is appropriate to introduce a function in R that can perform the analysis without the need for creating long syntax in order to perform the analysis. After testing the algorithms of these estimation methods in the previous simulation sections, I compacted them into a package that I named DoubleSL, which is hosted on my Github account, SamiSaadAlanazi. The idea behind building an R package was to allow others to be able use the method correctly and to allow the analysis to be replicated easily.

The DoubleSL package carries many advantages that makes it very useful, the first of which is that there is no need to install the necessary packages to implement these proposed estimation methods as with the packages related to the ML algorithms discussed earlier, the packages needed for the data generating process, or the packages needed for performing parallel computations. Another advantage when using the DoubleSL package is that it provides complementary functions for performing the DML method and for simulating datasets according to the data-generating process described earlier in the simulation scheme section. In addition, the DoubleSL package includes two datasets, used in later sections to serve as empirical examples when performing the DSL estimation method. Finally, the use of the DoubleSL package provides a neat and short syntax in R, making it easier for researchers to use and build on. The following table gives a summary about the available functions and datasets that are included in the DoubleSL package.

Table 21

Summary of Functions and Datasets Included in the DoubleSL Package

Function	Description
DATA1	Data generating process: Simulates data set that can used in DML and DSL analysis when the treatment variable is considered to be continuous.
DATA2	Data generating process: Simulates data set that can used in DML and DSL analysis when the treatment variable is considered to be binary.
DML1	The double machine-learning method (DML): Estimates the treatment effect using the DML method using Random Forests when the treatment variable is considered to be continuous.
DML2	The double machine-learning method (DML): Estimates the treatment effect using the DML method using Random Forests when the treatment variable is considered to be binary.
DSL1	The double super learner method (DSL): Estimates the treatment effect using the DSL method when the treatment variable is considered to be continuous.
DSL2	The double super learner method (DSL): Estimates the treatment effect using the DSL method when the treatment variable is considered to be binary.
S.DSL1	The selected double super learner method (SDSL): Estimates the treatment effect using the selected DSL method when the treatment variable is considered to be continuous.
S.DSL2	The selected double super learner method (SDSL): Estimates the treatment effect using the selected DSL method when the treatment variable is considered to be binary.
Dataset	**Description**
Example1	Student's Math and Portuguese Scores Dataset: A dataset for low-dimensional example with 382 rows and 36 variables to perform the DML and DSL methods.
Example2	Communities and Crime Dataset: A dataset for high-dimensional example with 123 rows and 127 variables to perform the DML and DSL methods.

Empirical Example: Student's Performance Dataset

In previous sections, the DSL method was applied on 32 different settings for datasets that were simulated using the data-generating process explained in the simulation scheme section. In this section, however, the DSL method is applied on a real-life dataset that is available in the public domain. The dataset selected for this demonstration is about the student performance, which was originally obtained from UCI Machine Learning Repository. This dataset is also included in the DoubleSL package introduced in the previous section under the Example1 dataset.

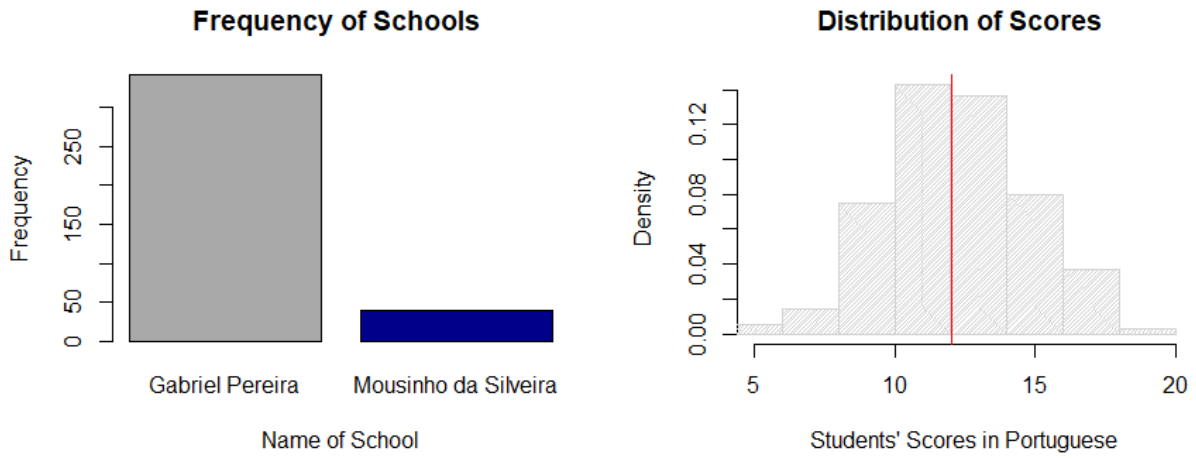
The student performance dataset was collected for 382 secondary school students in Portugal. In this dataset, 36 variables were recorded for each student including school name, sex, family size, grades in mathematics and Portuguese, and so many other attributes. Full information about the associated variables can be found in the DoubleSL package under the Example1 dataset. The literature associated with this dataset can be found in the study by Cortez and Silva (2008), where business intelligence and data mining techniques were incorporated to predict a student's performance in mathematics and Portuguese based on relative variables.

The goal of using this dataset was to demonstrate the application of the proposed DSL method on real-life datasets using the DoubleSL package. For this reason, the analysis of the student performance dataset had a different research objective compared to that of Cortez and Silva (2008). In this analysis, the objective was to investigate the effect of school on students' performance in Portuguese in the presence of other relevant variables. The school variable was a binary variable coded as follows: 0 for students in the Gabriel Pereira school, and 1 for students in the Mousinho da Silveira school. The response variable, on the other hand, the Portuguese final grade, is a numerical variable that ranged between 0 and 20. Descriptive analysis found that

final Portuguese scores had a mean of 12.52 and a standard deviation of 2.94. Moreover, 342 students studying at the Gabriel Pereira school, while only 40 students in the sample studying at the Mousinho da Silveira school. The following figure presents the descriptive statistics of those two variables that are of interest in this analysis.

Figure 23

Descriptive Statistics of School and Final Portuguese Scores Variables



To investigate the research objective, which was to estimate the true effect of school on students' final Portuguese scores while accounting for the presence of the remaining variables, three methods were implemented: (a) the double machine learning method (DML); (b) the double super learner method (DSL); and (c) the selected DSL method. By using the functions provided in the DoubleSL package for the case of binary treatment variables, the estimated effects of the treatment variable are reported in Table 22 for the three methods along with their estimated variance and the respective confidence intervals. The same DSL settings used in the

previous analysis in terms of candidate ML algorithms and the cross-validation folds were also applied in this analysis.

Table 22

Summary Statistics of School Effect on Students' Final Portuguese Scores

Method	Estimates	Variance	95% CI	
DML	-0.3960	47.5004	-1.0872	0.2951
DSL	-0.5586	34.6672	-1.1490	0.0319
Selected DSL	-0.4775	38.6778	-1.1012	0.1462

The results displayed in Table 22 show that the estimated effect of school is around -0.5. Specifically, the DML method estimated the school effect to be -0.39 point in students' final Portuguese scores, while the effect of the school is estimated to be -0.55 and -0.47 using the DSL and the selected DSL methods, respectively. This indicates that when a student chose to attend the Mousinho da Silveira school, which was coded as 1, his performance in the Portuguese final was expected to be lower by roughly half a point. In addition, it appears that the DSL method resulted in the lowest estimated variance of 34.66, which leads to the narrower confidence interval for the estimated effect of school. However, since the confidence intervals of the estimated targeted parameter using the three estimation methods included the 0, it is fair to say that effect of school was not statistically significant. It is worth mentioning that the best performing ML algorithms are shown to be LASSO and Random Forests as it appears in Table 23.

Table 23

Candidate Machine-Learning Algorithms' Performance in Estimating Nuisance Functions

Nuisance	LASSO	GLM	KNN	RF	Boosting
$\hat{g}_0(\boldsymbol{X}_{i\in T})$	0.716	0.171	0.000	0.000	0.113
$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	0.964	0.000	0.000	0.000	0.036
$\hat{m}_0(\boldsymbol{X}_{i\in T})$	0.000	0.000	0.000	1.000	0.000
$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	0.133	09a.048	0.000	0.819	0.000

Findings

In this dissertation, four research questions regarding the proposed DSL method were investigated. These research questions were concerned with how the estimators of the proposed DSL method were constructed, how the proposed method would perform in terms of bias reduction, how the proposed method could be improved in terms of computational efficiency, and, finally, how the proposed method could be implemented using R software.

To answer the first research question, nine steps were introduced in Chapter III that showed how the estimator of the targeted parameter could be constructed along with the estimated variance. More specifically, several equations were introduced within these nine steps, equations (38)–(49) and (56)-(57). These sets of equations were developed theoretically in parallel with the partially linear model of Robinson (1988) and the double ML approach of Chernozhukov et al. (2018). By following these nine steps and applying the equations mentioned using R, the DSL estimator of the targeted parameter can be numerically calculated, which has been verified using simulation.

Constructing the DSL estimator of the targeted parameter is one thing, but assessing the performance of the resultant estimates in terms of bias reduction and the validity of their

respective confidence intervals is another, which leads to the second research question of this dissertation. Since the DSL method was developed theoretically in parallel with the DML method, it was logical to compare the two methods in terms of bias and confidence intervals to see if the proposed method achieved any improvements in these areas. To do so, simulation was performed to investigate whether the proposed method led to improvements in the sense of bias reduction and whether the associated confidence intervals contained the true value of the targeted parameter. In the process of verifying the performance of the proposed method, 32 different settings for datasets were introduced based on the three distinctions: the number of associated covariates in the dataset, p = {20, 100, 1,000, 10,000}; the sample size in the dataset, N = {100, 500, 1,000, 5,000}; and the type of the treatment variable D as continuous or binary. Each of these datasets were created using the data-generating process using the partially linear model approach as in the study by Chernozhukov et al. (2018). Over 500 replications were then performed for each setting, and the resultant estimates of the targeted parameter using the DML and DSL were retained. The results showed that the DSL methods achieved improvement in bias reduction over the DML method in each setting. In addition, the associated confidence interval always contained the true value of the targeted parameter when using the DSL method, while this was not the case for the DML, at least when Random Forests algorithm was incorporated, when the treatment variable was binary, especially once the number of the associated covariates (p) in the datasets grew large.

Since the DSL method incorporates a set of ML algorithms, the computational intensity issue when using this method was present in the third research question. Three steps were introduced in the methodology chapter to choose the best performing algorithm, and a variant of the DSL method was introduced under the term selected DSL method. In short, this method used

the information retained by the SL function on a small sample to learn about the best performing ML algorithm, and then incorporated it alone in the DSL method. Simulation showed that when selecting only the best performing ML algorithm using the SL function, the estimation bias was always lower than when using the DML method. In addition, the respective confidence intervals using the selected DSL method always remained valid since the true value of the targeted parameter continued to fall within the confidence limits.

Finally, the fourth research question investigated the implementation of the proposed DSL method in R. For this purpose, a package named DoubleSL was created to perform the DSL method and to make analysis in this dissertation easy to replicate and verify. The package was made available in the public domain hosted by Github.com under my profile name SamiSaadAlanazi. This package includes several functions for the DSL and selected DSL methods as well as a number of other complimentary functions for the DML method and the data-generating process used to simulate datasets based on the sample size, the number of associated covariates, and the type of treatment variable. In addition, the DoubleSL package contains two real-life datasets to serve as empirical examples for the use of the DSL method. To install the DoubleSL package, one can simply use the following command in R:

install_github('SamiSaadAlanazi/DoubleSL')

CHAPTER V

DISCUSSION AND CONCLUSIONS

So far in this dissertation, the concept of the proposed DSL algorithm has been introduced with the goal of reducing estimation bias and allowing causal inference to be drawn using confidence intervals. Earlier than that, the semi-parametric modeling concepts, which the proposed DSL method is based on, were visited as well as a review of the historic developments of the ML algorithms, which the DSL method incorporates. After that, the formulas for estimating the targeted parameter and the variance using the DSL method were introduced. In addition, another version of the DSL method was introduced to improve the computational efficiency of the DSL algorithm, which is referred to as the selected DSL method. The DSL method was compared with the existing concept, the DML, which was introduced by Chernozhukov et al. (2018), in terms of bias, variance, and the validity of the associated confidence intervals.

To investigate the performance of the proposed DSL methods in comparison with the DML method, 32 different simulation settings for datasets were introduced according to the data-generating process described earlier. These simulated datasets were varied in terms of their dimensions by using a grid of values for the number of associated covariates (p) and the sample sizes (N) as well as varied in terms of the type of the treatment variable, continuous and binary. Once the simulation analysis was concluded, an R package was created and made available on Github that includes functions of the DSL methods as well as several other complementary functions so the analysis can be replicated easily by others. In addition, an empirical analysis of

real-life data was performed to demonstrate the application of employing the DSL method using the developed package.

In this chapter, I will be concluding this dissertation under two sections. The first section addresses the limitations that were observed during the analysis and during the development of the proposed method. In addition, the first section of this chapter includes some suggestions about future studies where there is room for further development and research. Finally, the last section of this chapter summarizes the overall conclusions found in this dissertation.

Limitations and Future Studies

The concept of the DSL method highlights a number of advantages over the DML method in terms of bias reduction and valid confidence intervals, but was not without its limitations. These limitations may seem as such, but they could also be viewed as opportunities for future research. In this section, some of the limitations are listed in this dissertation and a few research problems are proposed to be investigated by researchers for future research of the DSL method.

The first obvious limitation when using the DSL method is the computational intensity that this method requires. Since the DSL method requires the incorporation of multiple ML algorithms, it requires larger computing capabilities and consumes more time for execution in comparison with the DML. The lack of computational efficiency could be related to the algorithm engineering in the way the SL function was created. In addition, the SL function in R does not support data with missing values, which is also a clear disadvantage since missing values are present in most real-life data. One area for future research could be to attempt to reengineer the SL function by using tools and functions that are more efficient to help improve the computational efficiency and allow for data with missing values.

The second limitation of the DSL method when running the analysis over a number of replications is the phenomenon of producing a set of estimates with outliers, which was not the case with using the DML method. Although trimming treatment at a 1% level was found to be an effective solution for this issue, this phenomenon could inspire other researchers to investigate why it happened in the first place. Another limitation found in the DML and the DSL methods is that estimation is not stable. Since this dissertation was limited to the cross-fitting technique, where the samples are split equally and randomly into testing and training sets, this led to receiving different estimates every time these methods were replicated. One area of research could be invested in investigating these methods under a larger number of splits. Although there is literature on the DML method with multiple splits, the idea of investigating the number of splits using the DSL methods to improve estimation stability is worth looking into.

It is worth pointing out that the DSL method presented in this dissertation was limited to certain types of variables in terms of the response and the treatment variable. The response variable investigated in the simulation was continuous, while the treatment variables were limited to continuous and binary cases. Investigating the performance of the DSL method using other forms of variables could be an interesting research topic.

Conclusion

Earlier in this dissertation, a new proposed method I referred to as the DSL was introduced. This method was developed in parallel with the well-known method of Chernozhukov et al. (2018) called the DML. The DML method with Random Forests is used to make causal inference about the targeted parameter in the presence of high-dimensional nuisance function. The rationale behind proposing the DSL method was that instead of using a single ML algorithm to estimate the targeted parameter, the incorporation of multiple learners can not only

reduce the risk of selecting an inappropriate ML algorithm, but also can result in bias reduction and valid confidence intervals for the targeted parameter. The foundations of the proposed method were introduced and investigated using numerical simulation, and the estimates of the proposed method were compared with the DML approach across various data settings. The results showed that the use of the DSL method produced estimates of the targeted parameter that were better in terms of bias reduction compared with the DML method using Random Forests. Furthermore, the corresponding confident intervals of the DSL method remained valid over all the different settings for the simulated datasets, unlike those using the DML method with Random Forests where the respective confidence intervals failed in four occasions when the dimension of the simulated datasets grew dimensionally large. A new variant of the DSL method referred to as the selected DSL was then introduced, which is based on a single ML algorithm chosen by the SL function. Numerical analysis also showed that this new variant of the DSL method produced better results compared with the DML method in terms of bias and confidence intervals, but not in comparison with the DSL method that incorporates multiple ML algorithms. A package that contains the DSL functions was then introduced and made publicly available for researchers who are interested in using the proposed method and to help in replicating the results presented in this dissertation.

REFERENCES

Andrews, D. W. (1994). Asymptotics for semiparametric econometric models via stochastic equicontinuity. *Econometrica: Journal of the Econometric Society*, *62*(1), 43-72.

Bach, P., Chernozhukov, V., Kurz, M. S., & Spindler, M. (2021). DoubleML--An object-oriented implementation of double machine learning in R. *arXiv preprint arXiv:2103.09603*.

Belloni, A., Chernozhukov, V., & Hansen, C. (2014). High-dimensional methods and inference on structural and treatment effects. *Journal of Economic Perspectives*, *28*(2), 29-50.

Bickel, P. J. (1982). On adaptive estimation. *The Annals of Statistics*, *10*(3), 647-671.

Bzdok, D., Altman, N., & Krzywinski, M. (2018). Points of significance: Statistics versus machine learning. *Nature Methods*, *15*(04), 233-234.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., & Newey, W. (2017). Double/debiased/neyman machine learning of treatment effects. *American Economic Review*, *107*(5), 261-65.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *Econometrics Journal*, *21*(1), C1–C68. doi: 10.1111/ectj.12097.

Cortez, P., & Silva, A. M. G. (2008). Using data mining to predict secondary school student performance. *Proceedings of 5th Annual Future Business Technology Conference.* https://hdl.handle.net/1822/8024.

Donsker, M. D. (1951). An invariance principle for certain probability limit theorems. *Memoirs of the American Mathematical Society*.

Frisch, R., & Waugh, F. V. (1933). Partial Time Regressions as Compared with Individual Trends. *Econometrica*, *1*(4), 387–401. https://doi.org/10.2307/1907330.

Funk, M. J., Westreich, D., Wiesen, C., Stürmer, T., Brookhart, M. A., & Davidian, M. (2011). Doubly robust estimation of causal effects. *American Journal of Epidemiology*, *173*(7), 761-767.

Grossman, R., Kasif, S., Moore, R., Rocke, D., & Ullman, J. (1999). A report of three NSF workshops on mining large, massive, and distributed data. *Data Mining Research: Opportunities and Challenges, National Science Foundation from the Information and Data Management Program, the Algebra and Number Theory Program, from the Statistics and Probability Program*. https://papers.rgrossman.com/misc-001.pdf.

Ho, T. K. (1995, August). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition* (Vol. 1, pp. 278-282). IEEE.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55-67.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer.

Johnstone, I. M., & Titterington, D. M. (2009). Statistical challenges of high-dimensional data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *367*(1906), 4237-4253.

Kang, H. (2013). The prevention and handling of the missing data. *Korean Journal of Anesthesiology*, *64*(5), 402.

Knaus, M. C. (2021). A double machine learning approach to estimate the effects of musical practice on student's skills. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *184*(1), 282-300.

Li, X., & Shen, C. (2020). Doubly robust estimation of causal effect: Upping the odds of getting the right answers. *Circulation: Cardiovascular Quality and Outcomes*, *13*(1), e006065.

Luedtke, A. R., Sofrygin, O., Van Der Laan, M. J., & Carone, M. (2017). Sequential double robustness in right-censored longitudinal models. *arXiv Preprint arXiv:1705.02459*. https://doi.org/10.48550/arXiv.1705.02459.

Mansour, Y. (1997, July). Pessimistic decision tree pruning based on tree size. *In Machine learning-international workshop then conference* (pp. 195-201). Morgan Kaufmann Publishers.

Marr, B. (2018). *How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read*. Forbes. https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/?sh=7b81736760ba

Mitchell, T. M. (1997). *Machine learning*. Mcgraw-Hill.

Mittal, K., Khanduja, D., & Tewari, P. C. (2017). An insight into "Decision Tree Analysis." *World Wide Journal of Multidisciplinary Research and Development*, *3*(12), 111-115.

Newey, W. K. (1994). The Asymptotic Variance of Semiparametric Estimators. *Econometrica, 62*(6), 1349–1382. https://doi.org/10.2307/2951752

Neyman, J. (1959). Optimal asymptotic tests of composite hypotheses. *Probability and Statistics*, 213-234.

O'Driscoll, D., & Ramirez, D. E. (2016). Limitations of the least squares estimators; A teaching perspective. *Athens: ATINER'S Conference Paper Series*, No: STA2016-2074. http://hdl.handle.net/10395/2537.

Polley, E. (2010). *Super learner*. (Doctoral dissertation, University of California, Berkeley). https://escholarship.org/uc/item/4qn0067v.

Polley, E. C., Rose, S., & van der Laan, Mark J. (2011). Super learning. *Targeted learning* (pp. 43-66). Springer New York. https://doi.org/10.1007/978-1-4419-9782-1_3

Powell, J. L. (1994). Estimation of semiparametric models. *Handbook of Econometrics, 4*, 2443-2521.

Robins, J. M., Rotnitzky, A., & Zhao, L. P. (1994). Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association, 89*(427), 846-866.

Robinson, P. M. (1988). Root-N-Consistent Semiparametric Regression. *Econometrica, 56*(4), 931–954. https://doi.org/10.2307/1912705

Schmetterer, L. (1960). On a Problem of J. Neyman and E. Scott. *The Annals of Mathematical Statistics, 31*(3), 656–661. http://www.jstor.org/stable/2237575

Skelly, A., Dettori, J., & Brodt, E. (2012). Assessing bias: The importance of considering confounding. *Evidence-Based Spine-Care Journal, 3*(01), 9–12.

Stein, C. (1956). Efficient nonparametric testing and estimation. *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 187-195).

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological), 58*(1), 267–288. http://www.jstor.org/stable/2346178

Tsiatis, A. A. (2006). *Semiparametric theory and missing data*. Springer.

Urminsky, O., Hansen, C., & Chernozhukov, V. (2019, September 13). The Double-Lasso Method for Principled Variable Selection. https://doi.org/10.31234/osf.io/2pema

Van der Laan, M. J., Polley, E. C., & Hubbard, A. E. (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology*, *6*(1). https://doi.org/10.2202/1544-6115.1309.

Van der Vaart, A. W. (1998). *Asymptotic statistics.* Cambridge.

Wald, A. (1950). Note on Zero Sum Two Person Games. *Annals of Mathematics, 52*(3), 739–742. https://doi.org/10.2307/1969446

Wang, J., He, X., & Xu, G. (2020). Debiased inference on treatment effect in a high-dimensional model. *Journal of the American Statistical Association*, *115*(529), 442-454.

Wise, B. M., & Geladi, P. (2000). A brief introduction to multivariate image analysis (MIA). *Eigenvector Research.* http://www.eigenvector.com/Docs/MIA Intro. pdf

Yang, J. C., Chuang, H. C., & Kuan, C. M. (2020). Double machine learning with gradient boosting and its application to the Big N audit quality effect. *Journal of Econometrics*, *216*(1), 268-283.

Yu, H., Jiang, S., & Land, K. (2015). Multicollinearity in Hierarchical Linear Models. Social Science Research. 53, 118-136.

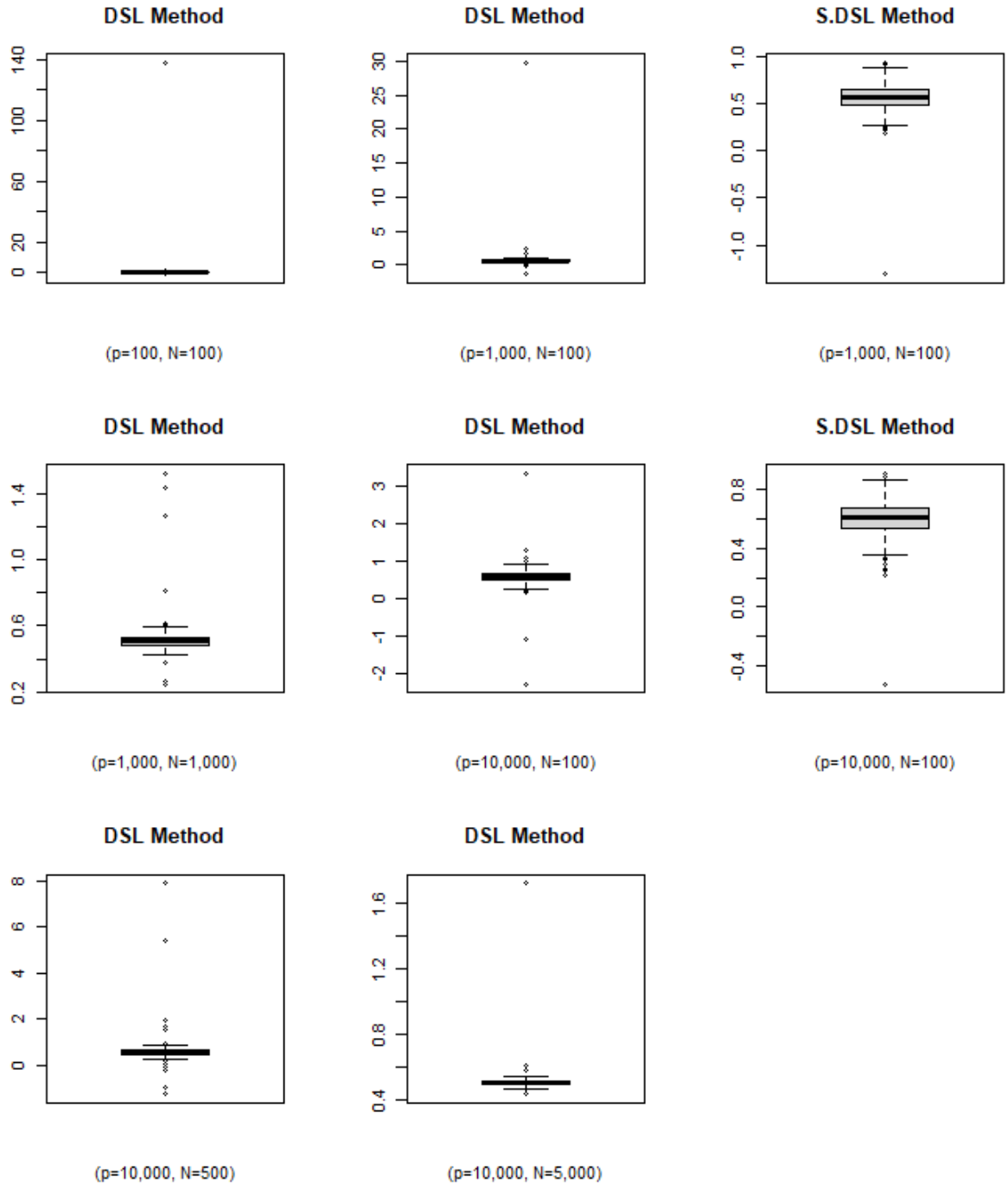
APPENDIX A

OUTLIERS DETECTION

Outliers Detection

Figure 24

Outliers for Estimating the Targeted Parameter when the Treatment D is Continuous



APPENDIX B

ESTIMATION OF BINARY TREATMENT EFFECTS USING EQUATIONS (44) – (46)

Estimation of Binary Treatment Effects Using Equations (44) – (46)

Figure 25

The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 20 when the Treatment Variable is Binary

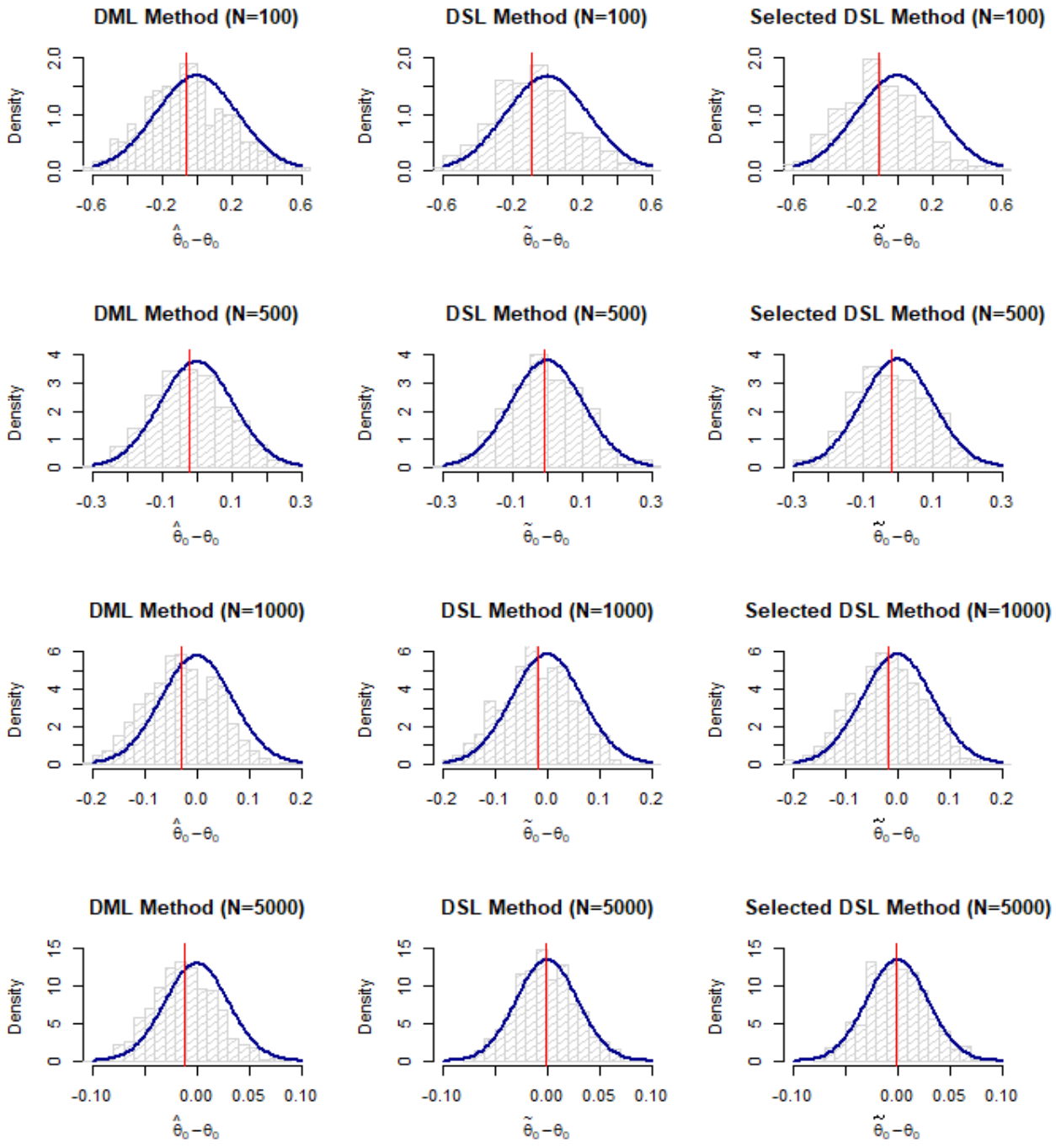


Table 24

P*erformance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 20 when the Treatment Variable is Binary*

Sample N	Nuisance	Best Candidate	LASSO	GLM	KNN	RF	Boosting
N=100	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	187	6	169	117	25
		Ratio	0.333	0.072	0.298	0.206	0.091
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	192	10	162	115	25
		Ratio	0.347	0.068	0.291	0.208	0.086
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	120	5	50	99	230
		Ratio	0.214	0.076	0.105	0.185	0.419
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	115	4	56	103	226
		Ratio	0.195	0.068	0.106	0.197	0.434
N=500	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	407	11	23	60	3
		Ratio	0.655	0.077	0.096	0.134	0.038
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	380	28	26	68	2
		Ratio	0.616	0.100	0.100	0.146	0.038
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	452	9	8	30	5
		Ratio	0.738	0.081	0.044	0.085	0.053
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	451	13	5	33	2
		Ratio	0.727	0.082	0.041	0.093	0.057
N=1,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	448	17	1	38	0
		Ratio	0.711	0.097	0.060	0.106	0.026
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	451	17	3	33	0
		Ratio	0.720	0.091	0.053	0.107	0.030
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	481	15	1	7	0
		Ratio	0.801	0.085	0.031	0.050	0.033
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	485	12	0	6	1
		Ratio	0.814	0.080	0.028	0.045	0.033
N=5,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	495	9	0	0	0
		Ratio	0.836	0.094	0.018	0.039	0.013
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	484	17	0	3	0
		Ratio	0.822	0.098	0.020	0.047	0.014
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	494	10	0	0	0
		Ratio	0.860	0.090	0.012	0.020	0.018
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	495	9	0	0	0
		Ratio	0.875	0.079	0.012	0.016	0.018

Table 25

Summary Statistics for Datasets with p = 20 when the Treatment Variable is Binary

N	Method	Estimates	Bias	Variance	95% CI	
100	DML	0.4350	0.0650	0.0559	-0.0283	0.8982
	DSL	0.4153	0.0847	0.0565	-0.0507	0.8813
	Selected DSL	0.3966	0.1034	0.0555	-0.0650	0.8582
500	DML	0.4771	0.0229	0.0112	0.2702	0.6841
	DSL	0.4904	0.0096	0.0109	0.2857	0.6952
	Selected DSL	0.4820	0.0180	0.0107	0.2788	0.6852
1,000	DML	0.4707	0.0293	0.0047	0.3364	0.6050
	DSL	0.4824	0.0176	0.0046	0.3501	0.6148
	Selected DSL	0.4826	0.0174	0.0046	0.3498	0.6153
5,000	DML	0.4879	0.0121	0.0009	0.4281	0.5477
	DSL	0.4990	0.0010	0.0009	0.4414	0.5566
	Selected DSL	0.4990	0.0010	0.0009	0.4414	0.5567

Note. Number of replications is 504.

Figure 26

The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 100 when the Treatment Variable is Binary

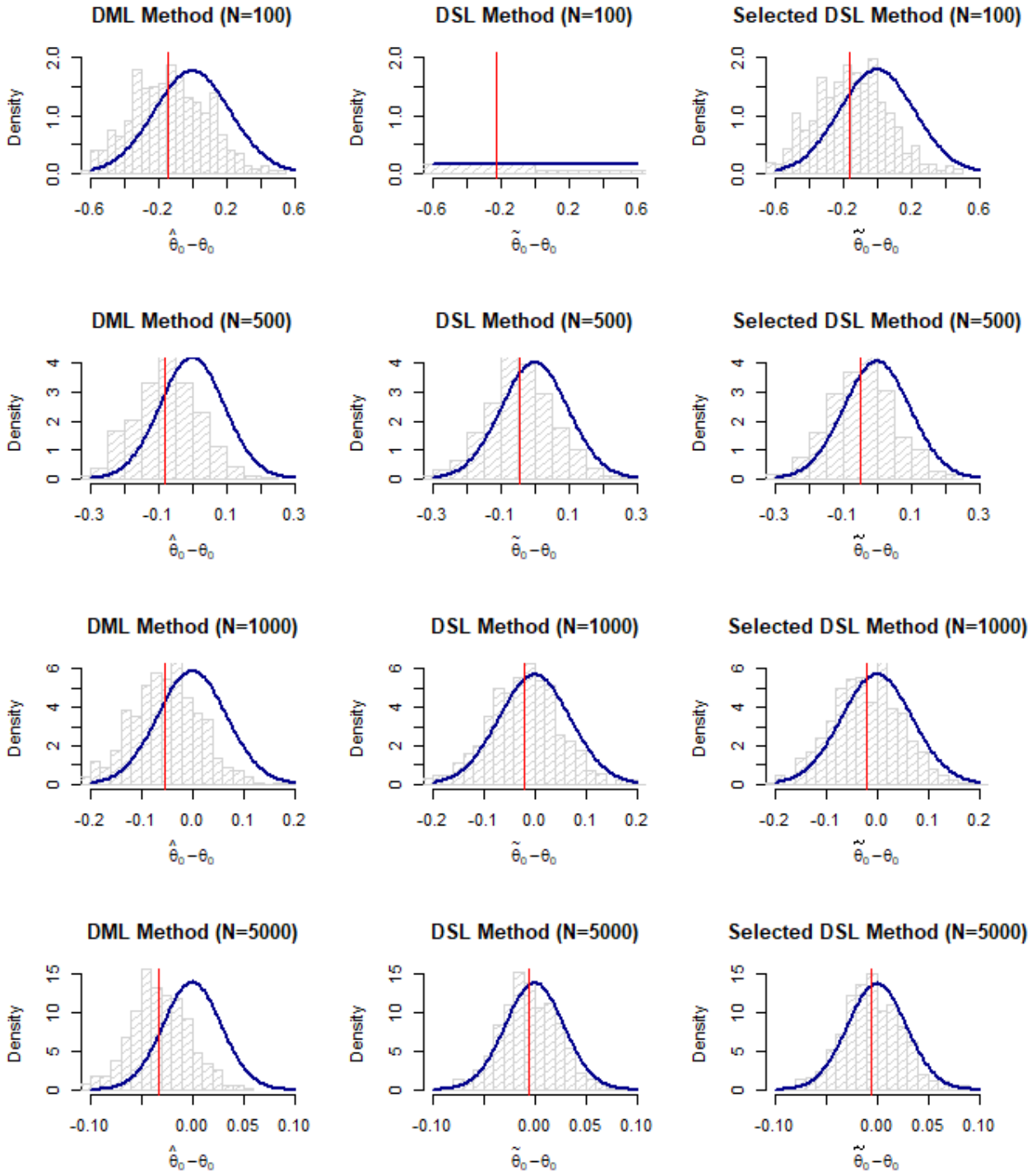
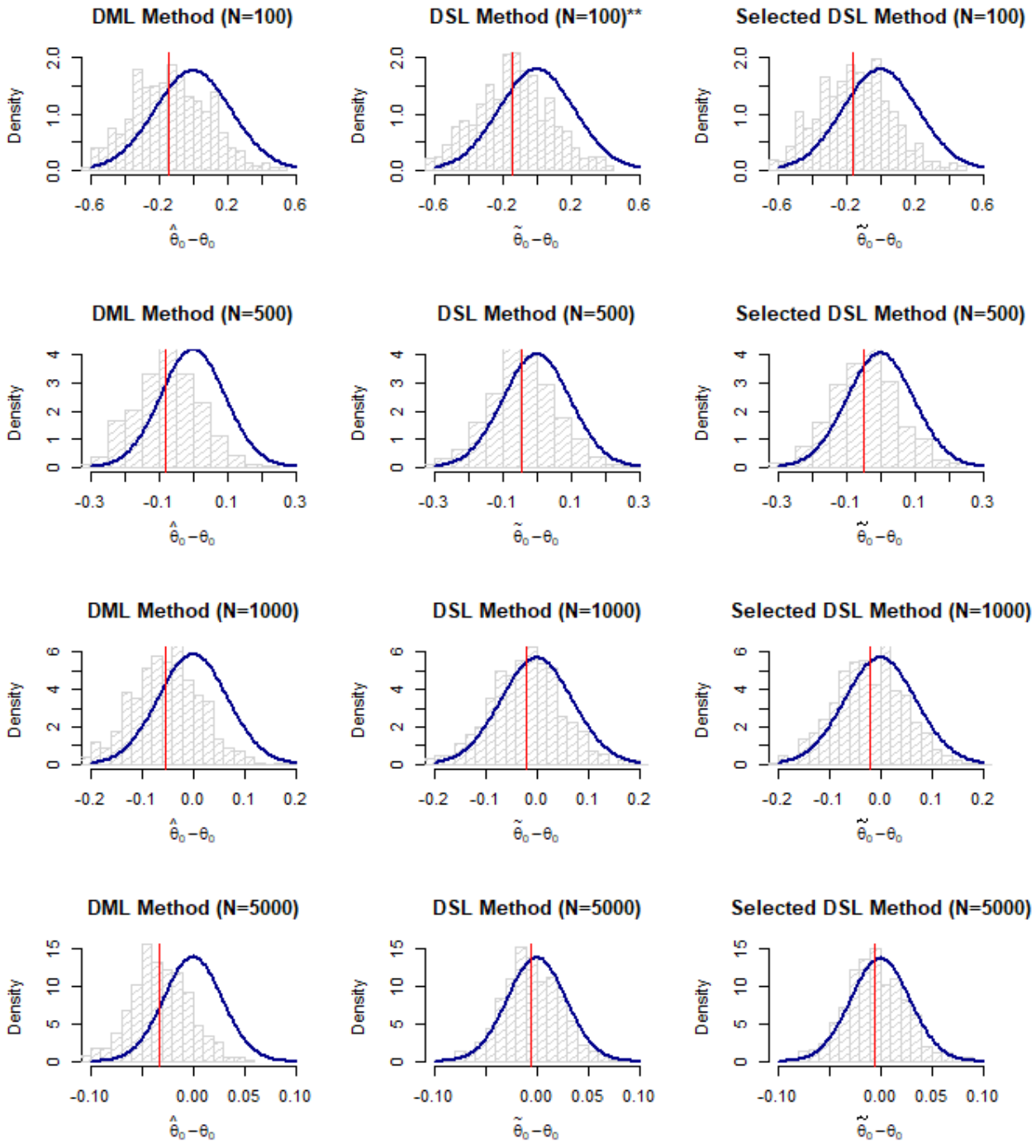


Figure 27

The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 100 when the Treatment Variable is Binary after Applying the Necessary Trimming



**trimming was applied.

Table 26

Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 100 when the Treatment Variable is Binary

Sample N	Nuisance	Best Candidate	LASSO	GLM	KNN	RF	Boosting
100	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	186	0	117	157	44
		Ratio	0.350	0.004	0.232	0.295	0.118
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	173	0	127	154	50
		Ratio	0.341	0.005	0.248	0.290	0.115
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	88	3	62	86	265
		Ratio	0.181	0.053	0.129	0.151	0.487
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	110	2	59	88	245
		Ratio	0.201	0.068	0.121	0.165	0.446
500	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	338	0	21	135	10
		Ratio	0.583	0.026	0.083	0.250	0.058
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	354	0	16	128	6
		Ratio	0.599	0.027	0.083	0.230	0.061
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	487	0	2	14	1
		Ratio	0.847	0.025	0.028	0.041	0.059
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	489	0	2	12	1
		Ratio	0.852	0.021	0.032	0.039	0.056
1,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	448	0	0	55	1
		Ratio	0.758	0.034	0.040	0.128	0.040
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	439	0	0	64	1
		Ratio	0.734	0.032	0.046	0.144	0.043
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	503	0	0	1	0
		Ratio	0.901	0.028	0.013	0.014	0.044
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	503	0	0	1	0
		Ratio	0.901	0.030	0.014	0.014	0.042
5,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	502	0	0	2	0
		Ratio	0.902	0.042	0.009	0.028	0.019
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	502	0	0	2	0
		Ratio	0.893	0.041	0.012	0.035	0.020
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.935	0.035	0.004	0.002	0.025
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	504	0	0	0	0
		Ratio	0.937	0.031	0.004	0.002	0.026

Table 27

Summary Statistics for Datasets with p = 100 when the Treatment Variable is Binary

N	Method	Estimates	Bias	Variance	95% CI	
100	DML	0.3583	0.1417	0.0507	-0.0832	0.7998
	DSL	0.2731	0.2269	5.7159	-4.4129	4.9590
	DSL *	0.3560	0.1440	0.0491	-0.0783	0.7902
	Selected DSL	0.3370	0.1630	0.0493	-0.0980	0.7720
500	DML	0.4207	0.0793	0.0089	0.2354	0.6061
	DSL	0.4553	0.0447	0.0098	0.2612	0.6493
	Selected DSL	0.4529	0.0471	0.0096	0.2606	0.6451
1,000	DML	0.4466	0.0534	0.0046	0.3141	0.5791
	DSL	0.4787	0.0213	0.0049	0.3417	0.6158
	Selected DSL	0.4783	0.0217	0.0049	0.3418	0.6149
5,000	DML	0.4676	0.0324	0.0008	0.4117	0.5235
	DSL	0.4950	0.0050	0.0008	0.4387	0.5512
	Selected DSL	0.4947	0.0053	0.0008	0.4380	0.5515

Note: Number of replications is 504.

*1% trimming has been applied.

Figure 28

The Distribution of ($\tilde{\theta}_0 - \theta_0$) for p = 1,000 when the Treatment Variable is Binary

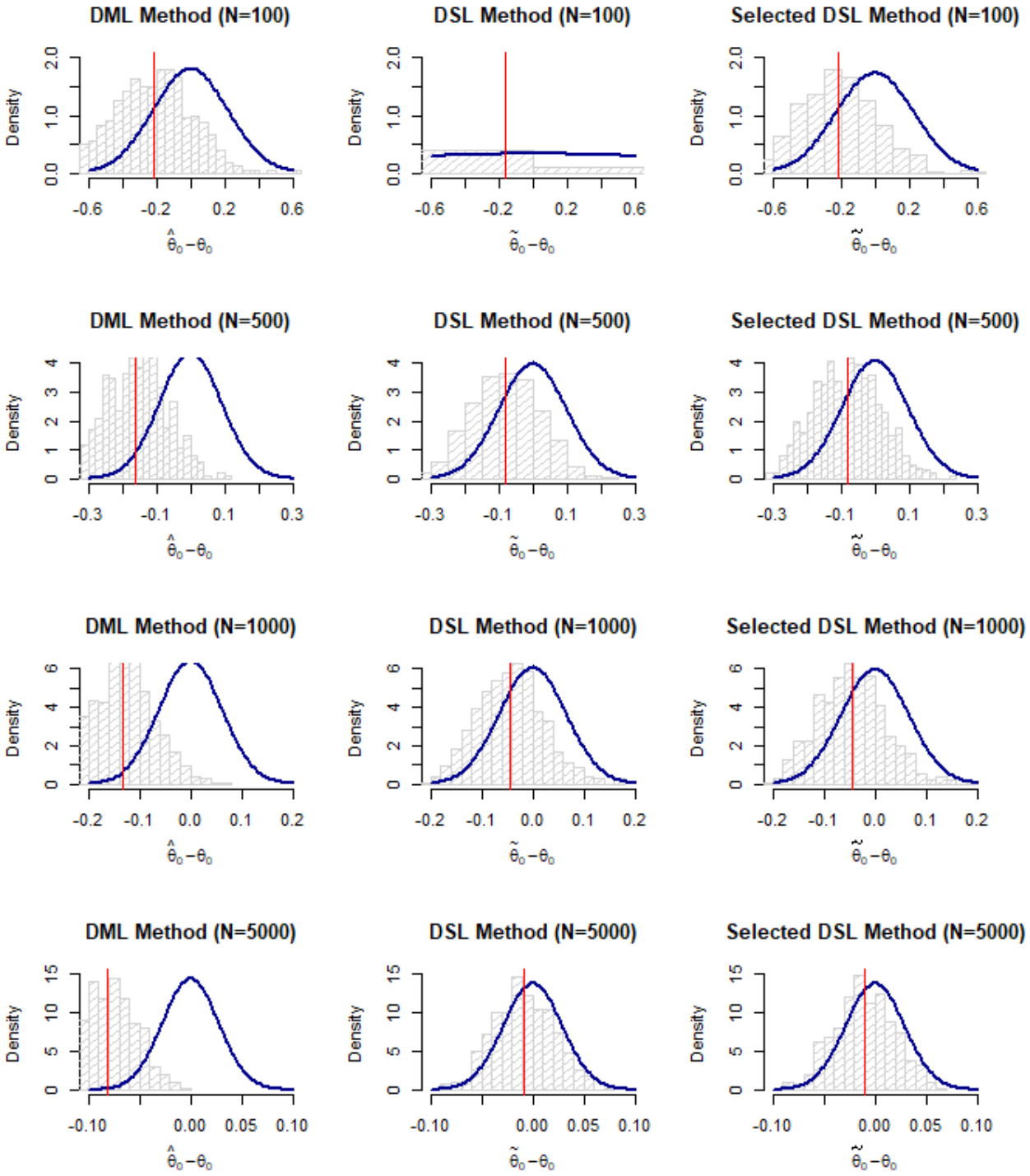
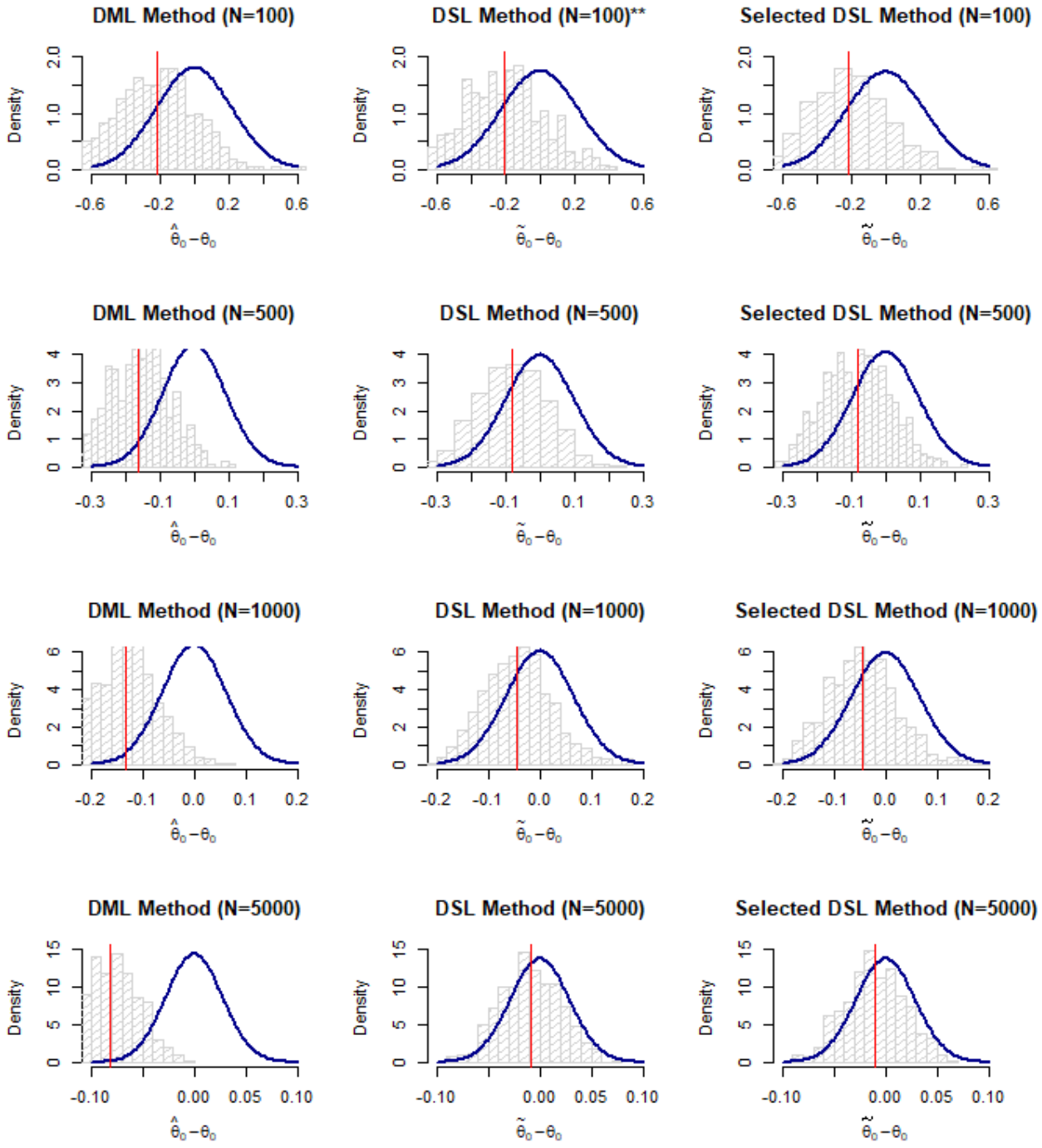


Figure 29

The Distribution of $(\tilde{\theta}_0 - \theta_0)$ for p = 1,000 when the Treatment Variable is Binary after Applying the Necessary Trimming



**trimming was applied.

Table 28

Performance of Each Candidate Learner in the Super Learner Analysis for Datasets with p = 1,000 when the Treatment Variable is Binary

Sample N	Nuisance	Best Candidate	LASSO	GLM	KNN	RF	Boosting
N=100	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	160	0	126	167	51
		Ratio	0.301	0.005	0.237	0.327	0.130
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	165	0	95	181	63
		Ratio	0.325	0.005	0.194	0.328	0.148
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	58	4	70	44	328
		Ratio	0.106	0.066	0.143	0.079	0.606
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	54	0	81	43	326
		Ratio	0.103	0.060	0.160	0.082	0.595
N=500	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	314	0	9	155	26
		Ratio	0.534	0.001	0.075	0.275	0.115
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	304	0	16	162	22
		Ratio	0.536	0.001	0.085	0.277	0.101
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	470	0	4	18	12
		Ratio	0.820	0.018	0.026	0.032	0.104
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	472	0	3	20	9
		Ratio	0.815	0.019	0.025	0.041	0.101
N=1,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	451	0	4	46	3
		Ratio	0.781	0.000	0.044	0.093	0.081
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	454	0	1	44	5
		Ratio	0.788	0.000	0.038	0.091	0.083
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.915	0.008	0.007	0.002	0.068
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	504	0	0	0	0
		Ratio	0.916	0.008	0.004	0.001	0.070
N=5,000	$\hat{g}_0(\boldsymbol{X}_{i\in T})$	Frequency	503	0	0	1	0
		Ratio	0.961	0.010	0.004	0.002	0.024
	$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	503	0	0	1	0
		Ratio	0.952	0.010	0.004	0.002	0.033
	$\hat{m}_0(\boldsymbol{X}_{i\in T})$	Frequency	504	0	0	0	0
		Ratio	0.952	0.005	0.001	0.000	0.043
	$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	Frequency	504	0	0	0	0
		Ratio	0.952	0.005	0.000	0.000	0.043

Table 29

Summary Statistics for Datasets with p = 1,000 when the Treatment Variable is Binary

N	Method	Estimates	Bias	Variance	95% CI	
100	DML	0.2837	0.2163	0.0487	-0.1489	0.7163
	DSL	0.3396	0.1604	1.3974	-1.9773	2.6565
	DSL *	0.2916	0.2084	0.0519	-0.1548	0.7380
	Selected DSL	0.2816	0.2184	0.0527	-0.1682	0.7313
500	DML	0.3380	0.1620	0.0084	0.1584	0.5176
	DSL	0.4195	0.0805	0.0099	0.2241	0.6150
	Selected DSL	0.4189	0.0811	0.0095	0.2281	0.6096
1,000	DML	0.3678	0.1322	0.0038	0.2463	0.4893
	DSL	0.4559	0.0441	0.0043	0.3273	0.5844
	Selected DSL	0.4544	0.0456	0.0045	0.3235	0.5852
5,000	DML	0.4191	0.0809	0.0008	0.3651	0.4731
	DSL	0.4908	0.0092	0.0008	0.4343	0.5473
	Selected DSL	0.4895	0.0105	0.0008	0.4331	0.5460

Note: Number of replications is 504.

*1% trimming has been applied.

APPENDIX C

EMPIRICAL EXAMPLE: COMMUNITIES AND CRIMES DATASET

Empirical Example: Communities and Crimes Dataset

An application of the DSL method using the dataset Example2 dataset under the DoubleSL package. The response variable is the total number of violent crimes per 100K population, while the treatment variable is percentage of people in the labor force, and unemployed. The sample size of this data set is 123 neighborhoods measured on 127 variables (high-dimensional dataset). The following are the results using the three methods:

Table 30

Summary Statistics about the Employment Effect on the Total Number of Violent Crimes

Method	Estimates	Variance	95% CI	
DML	0.3733	2.8694	0.0740	0.6727
DSL	0.4948	3.3267	0.1725	0.8172
Selected DSL	0.4346	4.3620	0.0655	0.8037

Table 31

Candidate Machine-Learning Algorithms' Performance in Estimating Nuisance Functions

Nuisance	LASSO	GLM	KNN	RF	Boosting
$\hat{g}_0(\boldsymbol{X}_{i\in T})$	0.663	0.010	0.000	0.181	0.147
$\hat{g}_0(\boldsymbol{X}_{i\in Tr})$	0.211	0.000	0.000	0.569	0.221
$\hat{m}_0(\boldsymbol{X}_{i\in T})$	0.940	0.015	0.000	0.000	0.045
$\hat{m}_0(\boldsymbol{X}_{i\in Tr})$	0.920	0.000	0.000	0.000	0.080

APPENDIX D

R SYNTAX

R Syntax

This section of the appendix includes directions about how to import the DoubleSL R package that was created for applying the DSL method. The package is hosted on Github.com under my profile name SamiSaadAlanazi. To install the package, use the following R syntax example:

```
install_github('SamiSaadAlanazi/DoubleSL')
library(DoubleSL)
Dat = DATA1(500, 20, 0.5)
DML1(Dat, 1, 2)
# Specify a library of candidate machine learning algorithms #
SL.library <- c("SL.biglasso", "SL.glm", "SL.kernelKnn", "SL.ranger",
"SL.xgboost")
DSL1(Dat, 1, 2, SL.library, 5L)
```

For more details, refer to chapter IV on page 95.

APPENDIX E

PERMISSION TO REPRODUCE FIGURE 3

Permission to Reproduce Figure 3

SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

Nov 28, 2022

This Agreement between University of Northern Colorado -- Sami Alanazi ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5437861105106
License date	Nov 28, 2022
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Super Learning
Licensed Content Author	Eric C. Polley, Sherri Rose, Mark J. van der Laan
Licensed Content Date	Jan 1, 2011
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	electronic
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
Will you be translating?	no
Circulation/distribution	1 - 29

Author of this Springer Nature content	no
Title	Graduate Student
Institution name	University of Northern Colorado
Expected presentation date	Nov 2022
Portions	Figure 3.2 on page 51
Requestor Location	University of Northern Colorado Greeley GREELEY, CO 80639 United States Attn: University of Northern Colorado
Total	0.00 USD

Terms and Conditions

Springer Nature Customer Service Centre GmbH
Terms and Conditions

This agreement sets out the terms and conditions of the licence (the **Licence**) between you and **Springer Nature Customer Service Centre GmbH** (the **Licensor**). By clicking 'accept' and completing the transaction for the material (**Licensed Material**), you also confirm your acceptance of these terms and conditions.

1. Grant of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, world-wide licence to reproduce the Licensed Material for the purpose specified in your order only. Licences are granted for the specific use requested in the order and for no other use, subject to the conditions below.

1. 2. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

1. 3. If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Scope of Licence

2. 1. You may only use the Licensed Content in the manner and to the extent permitted by these Ts&Cs and any applicable laws.

2. 2. A separate licence may be required for any additional use of the Licensed Material, e.g. where a licence has been purchased for print only use, separate permission must be obtained for electronic re-use. Similarly, a licence is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence. Any content owned by third parties are expressly excluded from the licence.

2. 3. Similarly, rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com/bookpermissions@springernature.com for these rights.

2. 4. Where permission has been granted **free of charge** for material in print, permission may also be granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.

2. 5. An alternative scope of licence may apply to signatories of the STM Permissions Guidelines, as amended from time to time.

3. Duration of Licence

3. 1. A licence for is valid from the date of purchase ('Licence Date') at the end of the relevant period in the below table:

Scope of Licence	Duration of Licence
Post on a website	12 months
Presentations	12 months
Books and journals	Lifetime of the edition in the language purchased

4. Acknowledgement

4. 1. The Licensor's permission must be acknowledged next to the Licenced Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book's homepage. Our required acknowledgement format is in the Appendix below.

5. Restrictions on use

5. 1. Use of the Licensed Material may be permitted for incidental promotional use and minor editing privileges e.g. minor adaptations of single figures, changes of format, colour and/or style where the adaptation is credited as set out in Appendix 1 below. Any other changes including but not limited to, cropping, adapting, omitting material that affect the meaning, intention or moral rights of the author are strictly prohibited.

5. 2. You must not use any Licensed Material as part of any design or trademark.

5. 3. Licensed Material may be used in Open Access Publications (OAP) before publication by Springer Nature, but any Licensed Material must be removed from OAP