Double/Debiased Machine Learning for Inference in Regression Discontinuous Designs under Local Randomization

Advised by:Jelena Bradic
Qianyi Wang

May 29, 2025

Abstract

Regression discontinuity designs (RDDs) are common quasi-experiment designs in economics and statistics. RDDs rely on discontinuous treatment assignment mechanisms to identify causal effects. Units are assigned a treatment based on whether their value of an observed covariate is above or below a fixed cutoff. The most popular methodologies for analyzing RDDs utilize continuity-based assumptions and local polynomial regression. However, an alternative framework, the local randomization framework, has repeatedly proven its usefulness in practice. On the other hand, the available data grows fast, such as the features of the units. To benefit from the flexibility of the machine learning methods to control for high-dimensional confounding and keep the validity of our statistical inference, we need double/debiased machine learning (DML). In this thesis, we apply DML on the inference in RDDs under local randomization. We illustrate our proposed methodology with a simulation study.

1 Introduction

Average treatment effect estimation is a crucial problem in causal inference and has been the topic of a considerable amount of recent literature Bradic, Wager, and Zhu (2019). Regression discontinuity designs are a popular approach to causal inference that rely on known discontinuous treatment assignment mechanisms to identify causal effects. Thistlethwaite and Campbell (1960). The basic idea behind the RD design is that the assignment to the treatment is determined, either completely or partially, by the value of a predictor (the covariate Z_i) being on either side of a fixed threshold Imbens and Lemieux (2008). In RDDs, we assume that there is a running variable and a cutoff (threshold), such that if the running variable is above the cutoff, we regard it as "assigned treatment", vice versa Villamizar-Villegas, Pinzon-Puerto, and Ruiz-Sanchez (2021). Since we do not take the treatment assignment to be random, approaches in random controlled trials, such as IPW, do not apply to RDD Rubin (2008). This means we have to develop a new algorithm to do the estimation.

The traditional inference approach in RDDs estimates treatment effects using local nonparametric methods and observations near the known cutoff. The key assumption is that the conditional expectation of a potential outcome is continuous at the threshold.

The analogy between RD designs and randomized experiments was first formalized without continuity conditions by Cattaneo et al. (2015). Rather than relying on limits as the score tends to the cutoff and on heuristic analogies between units barely above and barely below the cutoff, this framework considers assumptions under which the RD design would produce conditions equivalent to the conditions that would have occurred if a randomized experiment had been conducted in a neighborhood around the cutoff Matias and Rocio (2022). Thus, many methods for analyzing randomized experiments can be used. Motivated by this idea, we develop a methodological framework for analyzing RDDs under local randomization inference setup with another recently introduced method-double/debiased machine learning Chernozhukov et al. (2018).

The remainder of the paper is as follows. In Section 2, we review the local randomization framework for RDDs, specifically for Sharp RDDs. In Section 3, we review methods for choosing the window around the cutoff for which units are deemed as-if randomized. In Section 4, we review the framework for double/debiased machine learning. In Section 5, we discuss the application of DML in local randomization framework. In

Section 6, we compare the original local randomization approach and the DML local randomization approach through simulation studies. In section 7, we conclude the paper.

2 The Local Randomization Framework for Sharp Regression Discontinuity Designs

The key idea behind local randomization methods is that we assume that there is some window around the cutoff in an RDD such that units are as-if randomized to treatment and control. Therefore, methods for analyzing randomized experiments can be applied to estimate treatment effects within this window. After declaring the necessary notations, we will review the assumptions that the local randomization framework needs to estimate the treatment effects in RDDs. We will also introduce some assignment mechanism in RDDs and the analytic form of estimands.

2.1 Notations

We follow Imbens and Lemieux (2008) to discuss the framework of sharp RDD formalized using potential outcomes. Consider the setting with N units, indexed $i = 1, 2, ..., N$, we have potential outcomes $(Y_i(1), Y_i(0))$, where $Y_i(1)$ denotes the outcome of unit i under treatment, and $Y_i(0)$ denotes the outcome of unit i under control. Let Z_i denote the running variable for unit i, and let c be a known cutoff or threshold. Let T_i denote the treatment assignment for unit i, where $T_i = 1$ if unit i is assigned to treatment and 0 otherwise, which means $T_i = \mathbb{1}_{\{Z_i \geq c\}}$. Let X_i be a d dimensional vector of other pretreatment covariates. The pretreatment covariates are required within the local randomization framework because these covariates are used to determine if units within a particular window are effectively randomized, which we will discuss further in Section 3. Without these additional covariates X_i, the assumptions discussed later are not testable.

The distribution of treatment assignment T for units with $Z < c$ is different from the distribution of T for units with $Z \geq c$ for some cutoff c. The local randomization framework for RDDs mainly focuses on units within a window around the cutoff for which units are effectively randomized to treatment and control. Thus, we define the window $W_h = [c-h, c+h]$ for some bandwidth h. For simplicity, we choose a symmetric window. We denote the number of observations inside of the window be N_w.

1. Running variable: $Z_i \in \mathbb{R}$
2. Pretreatment covariates: $X_i \in \mathbb{R}^d$
3. Outcome variable: $Y_i \in \mathbb{R}$
4. Binary treatment: $T_i = \mathbb{1}_{\{Z_i \geq c\}}$ (T_i is not randomized)
5. Data set: $(X_i, Z_i, Y_i, T_i) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R} \times \{0, 1\}$

2.2 Local Randomization Mechanism Assumptions

The key assumption behind the local randomization approach to RDDs is that units near the cutoff are as-if randomly assigned to treatment. We now begin the discussion of local randomization by specifying the assumptions within the window around the cutoff that allow us to analyze the RDDs as a randomized experiment.

The local randomization framework for RDDs is characterized by two features: a) a known treatment assignment mechanism in the window and b) an exclusion restriction on the potential outcomes.Cattaneo et all. (2016)

2.2.1 Example of Local Randomization mechanism

The first feature is analogous to the requirement of know assignment mechanism in classical randomized experiments. One natural randomization mechanism is Bernoulli trials. Since we have binary treatment, we can model each treatment assignment as independent Bernoulli trails. Units are assigned independently

to treated and control group within the window. Typically, $e(X_i)$ needs to be estimated(e.g., via logistic regression). One simple choice of $e(X_i)$ is $\frac{N_w^+}{N_w}$. Under the Local Bernoulli Trials assumption, any treatment assignment in $\{0,1\}^{|W_h|}$ could have plausibly occurred for units in W_h, including the cases where all units are assigned to treatment or all units are assigned to control. Thus, this is the least strict assumption.

Local Bernoulli Trials: For units $i \in W_h$,

$$\mathbb{P}(T = t \mid X) = \prod_{i \in W_h} e(X_i)^{T_i}[1 - e(X_i)]^{1-T_i}, \quad \text{where } 0 < e(\mathbf{X}_i) < 1 \tag{2.1}$$

where $e(\mathbf{X}_i) \equiv \mathbb{P}(Z_i = 1 \mid \mathbf{X}_i)$ is the propensity score for unit i.

Since the local Bernoulli trials allow the situation of all treatment can be equal to 0 or 1, we need more restricted assumptions to avoid the generalization of that possible treatment assignment. Local Complete Randomization assumes that the propensity scores for all units in W_h are equal, conditional on the number of units assigned to treatment. This means that the probability of each treatment assignment is $\binom{N_w}{N_w^+}^{-1}$, where N_w^+ is the number of treatment units within the window W.

Local Complete Randomization: For units $i \in W_h$,

$$\mathbb{P}(T = t \mid X) = \begin{cases} \left(\binom{N_w}{N_w^+}\right)^{-1} & \text{if } \sum_{i \in W_h} T_i = N_w^+, \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

2.2.2 SUTVA and Local Unconfoundedness and Overlap

In general, the running variable is often correlated with the potential outcomes in RDDs. Such a relationship between the score and the potential outcomes would hinder the comparability of unis above and below the cutoff within the window because of the lack of common support in the score Matias and Rocio (2022). Thus, we need to explicitly give assumptions on the exclusion restriction.

Assumption 2.1 (Local SUTVA)**.** *There exist a window $W_h = [c - h, c + h]$ such that for each $i \in W_h$, consider two values Z_i' and Z_i'', where $Z_i' \neq Z_i''$, corresponding to treatment assignments $T_i' = \mathbb{1}(Z_i' > c)$ and $Z_i'' = \mathbb{1}(Z_i'' > c)$, where $\mathbb{1}$ denotes the indicator function for event A. If $T_i' = T_i''$, then $Y_i(T_i') = Y_i(T_i'')$.*

This means that for units $i \in W_h$, the treatment assignment of a unit depends on the running variable only through its being above or below c, and that the potential outcomes of each unit do not depend on other units treatment assignment.

We also need one additional assumption to make the average treatment effect identifiable. We need the potential outcomes are independent of treatment assignment given covariates and there is non-zero probability of units receiving treatment or control, which are very basic assumptions in causal inference. Under local randomization framework, we need these assumptions to be true for all units in the window $W_h = [c - h, c + h]$.

Assumption 2.2 (Local Unconfoundedness and Overlap)**.** *There exist a window $W_h = [c - h, c + h]$ such that for all $i \in W_h$,*

$$(Y_i(1), Y_i(0)) \perp\!\!\!\perp T_i \mid X_i \quad \textit{and} \quad 0 < \mathbb{P}(T_i = 1 \mid X_i) < 1 \tag{2.3}$$

The assumption that $0 < \mathbb{P}(T_i = 1 \mid X_i) < 1$ for all $i \in W_h$ ensures that there is a non-zero probability on each of the $2^{|W_h|}$ possible treatment assignments. The local unconfoundedness assumes that T_i is independent of the potential outcomes and holds conditional on X_i.

2.3 Estimand and Estimation

Under the local SUTVA, the regression functions $E[Y_i(1)|Z_i = z]$ and $E[Y_i(0)|Z_i = z]$ are constant for all values of the running variable inside the window W_h. The average treatment effect is then the difference between $E[Y_i(1)|Z_i = z]$ and $E[Y_i(0)|Z_i = z]$ inside W_h. For the N_w units with $Z_i \in W_h$, the sharp local randomization RD treatment effect cam be defined as

$$\theta_{SLR} = \frac{1}{N_w} \sum_{i:Z_i \in W_h} \mathbb{E}_w[Y_i(1) - Y_i(0)] \tag{2.4}$$

$$= \frac{1}{N_w} \sum_{i:Z_i \in W_h} \mathbb{E}[\frac{T_i Y_i}{\mathbb{P}_w(T_i = 1)}] - \frac{1}{N_w} \sum_{i:Z_i \in W} \mathbb{E}[\frac{(1 - T_i)Y_i}{1 - \mathbb{P}_w(T_i = 1)}] \tag{2.5}$$

where $\mathbb{E}_w$ and $\mathbb{P}_w$ denote expectation and probability computed conditionally for all units with $Z_i \in W_h$ Matias and Rocio (2022).

Similar to most works on causal inference, a common choice for the estimation of average treatment effect is difference-in-means.

$$\hat{\theta}_{SLR} = \bar{Y}^+ - \bar{Y}^- \tag{2.6}$$

$$= \frac{1}{N_w} \sum_{i:Z_i \in W_h} \frac{T_i Y_i}{\mathbb{P}_w(T_i = 1)} - \frac{1}{N_w} \sum_{i:Z_i \in W_h} \frac{(1 - T_i)Y_i}{1 - \mathbb{P}_w(T_i = 1)} \tag{2.7}$$

where W_i denotes appropriate weights that are chosen according to the assumptions and the framework employed.

3 Bandwidth selection

In order for local randomization methods to yield trustworthy causal inferences, we need to find a bandwidth h such that, within the window $W_h = [c-h, c+h]$, it is plausible that Local SUTVA, Local Unconfoundedness and overlap and a particular assignment mechanism hold. In this section, we will mainly utilize the method in rdlocrand package. Motivated by the idea that the treatment assignment is as-if random inside the window, Cattaneo et al. (2015) propose that distribution of preintervention covariates(before treatment assignment) and postintervention covariates(after treatment assignment) should be the same between treated and control units. The distribution of these covariates for control and treatment units should be unaffected by the treatment within W_0 but should be affected by the treatment outside the window.

Define X be the $n \times k$ matrix with k covariates. For an arbitrary window W_i, let X_{W_i} be the subvector corresponding to units with running variable inside the window W_i. Then, the window selection algorithm is the following:

1. Choose an initial small window, $\hat{W}_1$.

2. For each of the k covariates, conduct a test of the null hypothesis of no effect of the treatment on the covariate using some test statistic $T(X_{\hat{W}_1}, Z_{\hat{W}_1})$. Take the minimum p-value from the k test.

3. If the minimum p-value obtained in step 2, p_1, is less than some prespecified level (0.15 by default), the initial window was too large. Then, we decrease the initial window and start over. If the window cannot be decreased (for example, because a smaller window would contain too few data points), we conclude that the window cannot be found.

4. If $p_1 \geq 0.15$, then we choose a larger window $\hat{W}_1 \subset \hat{W}_2$, and go back to step 2 to calculate p_2. Repeat the process until the minimum p-value is less than 0.15. The selected window is the largest window such that the minimum p-value is larger than or equal to 0.15 in that window and in all windows contained in it.

The resulting window, $\hat{W}$, is the estimate of W_0. Intuitively, this algorithm would choose the largest window in which the distributions of all covariates are not affected by the treatment assignment.

The p-value can only tell us whether the distribution of covariats are affected inside the window. If the window $\hat{W}_i$ haves p-value is greater or equal to 0.15 and the window $\hat{W_{i+1}}$ have p-value less than 0.15, we can conclude that the treatment assignmenet does not affect the distribution of covariates inside $\hat{W}_i$ but affect the distribution of covariates outside $\hat{W}_i$. If the smallest window has p-value less than 0.15, then we can only conclude that the distribution of covariates is affected in all windows, which invalidate our assumption. Thus, we reach to the result in step 3.

Usually, researchers are concerned about controlling Type I error to avoid rejecting the null hypothesis too often when it is true. However, in our study, our goal is to learn whether the data support the existence of a window around the cutoff where our null hypothesis(treatment assignment does not affect covariate distribution) fails to be rejected. We actually focus on controlling Type II error. And by power calculations, the window selection method recommends 0.15 as the significance value, instead of the conventional 0.05.

4 Double/Debiased Machine Learning Methods

To estimate and construct confidence intervals for a parameter of interest when having a high-dimensional set of covariates, researchers introduced machine learning methods. However, naively plugging data in these methods would cause two problems—overfitting and regularization bias. Chernozhukov et al. (2018) showed these two problems can be vanished by using two adjustments–cross-fitting and Neyman-orthognoal scores. In terms of estimating treatment effects, Chernozhukov et al. (2018) combined machine learning techniques with interactive model, introducing a robust methodology for estimating treatment effects in high-dimensional settings. They propose an extension on classical literature under unconfoundedness using machine learning methods. They consider the estimation of average treatment effects when treatment effects are fully heterogeneous and the treatment variable is binary.

4.1 Inference on Treatment Effects in the Interactive Model

Let $T_i \in \{0, 1\}$. Let potential outcomes be $(Y_i(1), Y_i(0))$, where $Y_i(1)$ denotes the outcome of unit i under treatment, and $Y_i(0)$ denotes the outcome of unit i under control. Let $X_i \in \mathbb{R}^d$ be covariates. With the vector $(X_i, Y_i, T_i) \in \mathbb{R}^d \times \mathbb{R} \times \{0, 1\}$, we have

$$Y_i = g_0(T_i, X_i) + U_i, \quad \mathbb{E}[U_i|X_i, T_i] = 0 \tag{4.1}$$

$$T_i = m_0(X_i) + V_i, \quad \mathbb{E}[V_i|X_i] = 0 \tag{4.2}$$

A common true parameter of interest in this model is the average treatment effect:

$$\theta = \mathbb{E}_p[g_0(1, X) - g_0(0, X)] \tag{4.3}$$

The covatiates X_i affect the treatment variable via the propensity score m_0 and the potential outcome Y_i via the function g_0. Since both of these functions are unknown and potentially complicated, we can take advantage of machine learning methods to learn them.

For estimation of the ATE, we use

$$\psi(W; \theta, \eta) := (g(1, X) - g(0, X)) + \frac{T(Y - g(1, X))}{m(X)} - \frac{(1 - T)(Y - g(0, X))}{1 - m(X)} - \theta, \tag{4.4}$$

where the nuisance parameter is $\eta = (m, g)$ consists of P-square-integrable functions g and m mapping the support of (T, X) to $\mathbb{R}$ and the support of X to $(\epsilon, 1 - \epsilon)$, respectively, for some $\epsilon \in (0, 1/2)$. The true value of η is $\eta_0 = (m_0, g_0)$.

4.2 Assumption

To make the interactive model useful, Chernozhukov et al. (2018) raised the regularity condition for ATE estimation 4.1 and the DML inference on ATE theorem 4.1. The this ensures the estimator converges to the true parameter at rate of $\sqrt{n}$ and has asymptotic normality.

Using the score, it can be easily seen that true parameter values θ_0 for ATE obey the moment condition $E_P\psi(W;\theta_0,\eta_0)=0$, and also that the orthogonality condition $\partial_\eta E_P\psi(W;\theta_0,\eta_0)[\eta-\eta_0]=0$ holds.

Let $(\delta_N)_{N=1}^\infty$ and $(\Delta_N)_{N=1}^\infty$ be sequences of positive constants approaching 0. Also, let c,ϵ,C and q be fixed strictly positive constants such that $q>2$, and let $K\geq 2$ be a fixed integer. Moreover, for any $\eta=(\ell_1,\ldots,\ell_l)$, denote $\|\eta\|_{P,q}=\max_{1\leq j\leq l}\|\ell_j\|_{P,q}$. For simplicity, assume that N/K is an integer.

Assumption 4.1 (Regularity Conditions for ATE Estimation)**.** *For all probability laws $P\in\mathcal{P}$ for the triple (Y,T,X) the following conditions hold: (a) equations 4.1-4.2 hold, with $T\in\{0,1\}$, (b) $\|Y\|_{P,q}\leq C$, (c) $P_P\{\epsilon\leq m_0(X)\leq 1-\epsilon\}=1$, (d) $\|U\|_{P,2}\geq c$, (e) $\|E_P[U^2\mid X]\|_{P,\infty}\leq C$, and (f) given a random subset I of $[N]$ of size $n=N/K$, the nuisance parameter estimator $\hat{\eta}_0=\hat{\eta}_0((W_i)_{i\in I^c})$ obeys the following conditions: with P-probability no less than $1-\Delta_N$:*

$$\|\hat{\eta}_0-\eta_0\|_{P,q}\leq C,\quad \|\hat{\eta}_0-\eta_0\|_{P,2}\leq\delta_N,\quad \|\hat{m}_0-1/2\|_{P,\infty}\leq 1/2-\epsilon,\quad \textit{and}$$

for the score ψ in 4.4, where $\eta_0=(g_0,m_0)$ and the target parameter is ATE,

$$\|\hat{m}_0-m_0\|_{P,2}\times\|\hat{g}_0-g_0\|_{P,2}\leq\delta_N N^{-1/2}.$$

Theorem 4.1 (DML Inference on ATE)**.** *Suppose that $\theta_0=E_P[g_0(1,X)-g_0(0,X)]$ and the score ψ in 4.4 is used. In addition, suppose that 4.1 holds. Then the DML estimators $\hat{\theta}$ obey*

$$\sigma^{-1}\sqrt{N}(\hat{\theta}_0-\theta_0)\rightsquigarrow N(0,1), \tag{4.5}$$

uniformly over $P\in\mathcal{P}$, where $\sigma^2=E_P[\psi^2(W;\theta_0,\eta_0)]$. Consequently, confidence regions based upon the DML estimators $\hat{\theta}$ have uniform asymptotic validity:

$$\lim_{N\to\infty}\sup_{P\in\mathcal{P}}\left|P_P\left(\theta_0\in[\hat{\theta}_0\pm\Phi^{-1}(1-\alpha/2)\hat{\sigma}/\sqrt{N}]\right)-(1-\alpha)\right|=0.$$

4.3 Machine Learning Method

Machine Learning method primarily focused on pattern recognition. The goal is to build models under fewer distributional assumptions. So ML methods are more like algorithms instead of starting with a relatively simple, predefined equation. Chernozhukov et al. (2018) apply Lasso, Random Forest, and Regression Trees in empirical studies. For our study, we use these three machine learning methods as well. To make these methods more clear, we give a short description about these methods.

4.3.1 Lasso

Least Absolute Selection and Shrinkage Operator Tibshirani (1996) is a machine learning method that can be applied when having a linear regression with many regressors. It minimizes the sum of squared residuals with an additional term:

$$\min_\beta\sum_{i=1}^N(Y_i-X_i\beta)^2+\lambda\cdot\|\beta\| \tag{4.6}$$

where $\|\beta\|=\sum_{k=1}^K|\beta_k|$. Rewrite (4.6) as follows to choose the penalty parameter

$$\min_\beta\sum_{i=1}^N(Y_i-X_i\beta)^2\quad\text{s.t.}\quad\sum_{k=1}^K|\beta_k|\leq t\cdot\sum_{k=1}^K|\beta_k^{\text{ols}}| \tag{4.7}$$

where t is a scalar between zero and one. When t equals zero, it is easy to see that all estimates shrinks to zero. When t is equal to one, the estimates are not shrinking and it is just OLS. The penalty parameter λ in (4.6) or t in (4.7) are chosen through cross-validation.

4.3.2 Random Forests

When applying Random Forests, we draw several bootstrap samples from the data and start to build a tree for each bootstrap sample. First, we consider all the samples in a single root node. Then, we recursively split nodes. We randomly select L covariates out of the K total available covariates. Among the subset consisting of L covariates, we select the optimal covariate and split threshold to create child nodes. Then, we repeat this splitting procedure for any resulting node that contains more units than a predefined minimum leaf size. If a node meets this minimum criterion or other stopping rules, it becomes a terminal leaf. Finally, we take the average of the predictions from all the individual trees grown on the bootstrap samples to get the final Random Forest prediction.

4.3.3 Neural Network

A neural network algorithm processes input data through interconnected layers of nodes (neurons). Input data are fed into the first layer. Each neuron receives inputs, multiplies them by associated weights, sums these weighted inputs, adds a bias, and then passes the result through an activation function to produce its output. The output is then fed forward as input to the neurons in the subsequent layer. This input-output process repeats until the final layer produces the network's prediction. The network's prediction is compared to the actual target value from the training data using a loss function. This measures how wrong the prediction was. The error calculated by the loss function is propagated backward through the network. This step calculates the gradient of the loss function with respect to each weight and bias, essentially determining how much each parameter contributed to the error. Then, an optimization algorithm uses these gradients to update the weights and biases in a direction that minimizes the loss. Finally, these processes are repeated iteratively for many data samples and multiple passes through the entire training dataset until the network's performance converges.

5 Application of DML in Local Randomization Framework

5.1 Motivation

5.1.1 Theoretical feasibility

Because the key idea behind local randomization methods is that we assume that units are as-if randomized to treatment and control inside some window around the cutoff in an RDD. Therefore, methods for analyzing randomized experiments can be applied to estimate treatment effects within this window. This means we can utilize double/debiased machine learning method inside the window. The other necessary step in the inference of RDDs is the local randomization mechanism. This can also be solved by DML because we have propensity score function $m_0(X)$ in the DML framework. Since the confounding factors affect the treatment assignment via the propensity score, we can interchange the original local randomization mechanism, for example, local bernoulli trails and local complete randomization above, by function $m_0(X)$. To represent the treatment assignment is completely determined by the running variable Z, we define $m_0(X)$ maps from the support of Z to $(\epsilon, 1-\epsilon)$, respectively, for some $\epsilon \in (0, 1/2)$.

5.1.2 Benefit

We notice that the window selection procedure would dramatically decrease the usable observations in inference. Thus, though we may have a large number of observations in the beginning, we may have a data set where the number of features is large relative to the number of observations after the window selection. That is, window selection may shift the problem from the setting where classical statistical inference methods are typically appropriate to one that requires the techniques of high-dimensional statistical inference. In this case, double machine learning offers significant benefits in solving high-dimensional causal statistical problems. The most important benefit of DML is that DML enables valid statistical inference under high-dimensional settings. Getting reliable standard errors, confidence intervals, and p-values for a parameter θ_0 is difficult when high-dimensional machine learning methods are used in the estimation process. However,

DML yields the estimates of θ_0 that are asymptotically normal and centered around the true parameter θ_0. This allows us to construct valid confidence intervals and run hypothesis testing.

5.2 Assumption

The assumptions of the application of DML in local randomization framework should follow the assumption from the local randomization framework and the local regularity conditions for ATE estimation.

First, the structure of interactive model requires two parts: propensity score and overall regression function. By defining the propensity score here, we can fulfill the requirement of the treatment assignment mechanism of the local randomization framework. The key difference in our assumption and the original interactive model is that, the argument of propensity score and the regression function involves running variable Z and covariates X, instead of X only.

Assumption 5.1 (Local Propensity Score)**.** *Let c be the cutoff. There exists a window $W_h = [c-h, c+h]$ such that there exist a function m_0 mapping the support of running variable Z and covariates X to $(\epsilon, 1-\epsilon)$ such that*

$$T = m_0(Z, X) + V, \quad \mathbb{E}[V|Z, X] = 0 \tag{5.1}$$

where V is the error term.

Assumption 5.2 (Local Regression Function)**.** *Let c be the cutoff. There exists a window $W_h = [c-h, c+h]$ such that there exist a function g_0 mapping the support of (T, Z, X) to $\mathbb{R}$ such that*

$$Y = g_0(T, Z, X) + U, \quad \mathbb{E}[U|Z, X] = 0 \tag{5.2}$$

where U is the error term.

Similarly, we need to define the exclusion restriction on potential outcomes for the local randomization framework. Under this assumption, the treatment assignment of a unit depends on the running variable only through its being above or below the cutoff. And the potential outcomes of each unit do not depend on other units' treatment assignment inside the window. Meanwhile, we do not have different version of treatment assignment for differenct values of the running variable within the window.

Assumption 5.3 (Local DML SUTVA)**.** *Let c be the cutoff. There exists a window $W_h = [c-h, c+h]$ such that for each $i \in W_h$, consider two values Z_i' and Z_i'', where $Z_i' \neq Z_i''$, corresponding to treatment assignments $T_i' = \mathbb{1}(Z_i' > c)$ and $T_i'' = \mathbb{1}(Z_i'' > c)$, where $\mathbb{1}$ denotes the indicator function for event A. If $T_i' = T_i''$, then $Y_i(T_i') = Y_i(T_i'')$.*

In order for the parameter to be identifiable, we still need unconfoundedness and overlap inside the window. That says, the potential outcomes are independent of treatment assignment given covriates. And there is a non zero probability of units receiving treatement or control.

Assumption 5.4 (Local DML Unconfoundedness and Overlap)**.** *Let c be the cutoff. There exists a window $W_h = [c-h, c+h]$ such that for all $i \in W_h$,*

$$(Y_i(1), Y_i(0)) \perp\!\!\!\perp T_i \mid X_i \quad \textit{and} \quad 0 < \mathbb{P}(T_i = 1 \mid X_i) < 1 \tag{5.3}$$

Then, to make DML valid, we need the regularity conditions for ATE inside the window.

Assumption 5.5 (Regularity Conditions for ATE Estimation)**.** *There exists a window $W_h = [c-h, c+h]$. Let N_w be the number of observations in W_h. Let $(\delta_{N_w})_{N_w=1}^{\infty}$ and $(\Delta_{N_w})_{N_w=1}^{\infty}$ be sequences of positive constants approaching 0. Also, let c, ϵ, C and q be fixed strictly positive constants such that $q > 2$, and let $K \geq 2$ be a fixed integer. Moreover, for any $\eta = (\ell_1, \ldots, \ell_l)$, denote $\|\eta\|_{P,q} = \max_{1 \leq j \leq l} \|\ell_j\|_{P,q}$. For simplicity, assume that N/K is an integer.*

For all probability laws $P \in \mathcal{P}$ for the quadruple (Y, T, Z, X) the following conditions hold: (a) equations 5.1-5.2 hold, with $T \in \{0, 1\}$, (b) $\|Y\|_{P,q} \leq C$, (c) $\|U\|_{P,2} \geq c$, (d) $\|E_P[U^2 \mid Z, X]\|_{P,\infty} \leq C$, and (e) given a random subset I of $[N_w]$ of size $n = N_w/K$, the nuisance parameter estimator $\hat{\eta}_0 = \hat{\eta}_0((T_i)_{i \in I^c})$ obeys the following conditions: with P-probability no less than $1 - \Delta_{N_w}$:

$$\|\hat{\eta}_0 - \eta_0\|_{P,q} \leq C, \quad \|\hat{\eta}_0 - \eta_0\|_{P,2} \leq \delta_{N_w}, \quad \|\hat{m}_0 - 1/2\|_{P,\infty} \leq 1/2 - \epsilon, \quad \textit{and}$$

for the score ψ in 4.4, where $\eta_0 = (g_0, m_0)$ and the target parameter is ATE,

$$\|\hat{m}_0 - m_0\|_{P,2} \times \|\hat{g}_0 - g_0\|_{P,2} \leq \delta_{N_w} {N_w}^{-1/2}.$$

5.3 Bandwidth Selection

The algorithm to find the window is similar as before. The key idea is the same: the distribution of covariates before treatment assignment and covariates after treatment assignment should be the same between treated and control group. But the distribution should be affected by the treatment assignment outside the window.

1. Choose an initial small window, $\hat{W}_1$.
2. For each of the k covariates, we conduct a test of the null hypothesis of no effect of the treatment on the covariate using some test statistic $T(X_{\hat{W}_1}, Z_{\hat{W}_1})$. Take the minimum p-value from the k test.
3. If the minimum p-value obtained in step 2, p_1, is less than 0.15, the initial window was too large. Then, we decrease the initial window and start over. If the window cannot be decreased (for example, because a smaller window would contain too few data points), we conclude that the window cannot be found. Otherwise, if $p_1 \geq 0.15$, then we choose a larger window $\hat{W}_1 \subset \hat{W}_2$, and go back to step 2 to calculate p_2.
4. Repeat the process until the minimum p-value is less than 0.15. The selected window is the largest window such that the minimum p-value is larger than or equal to 0.15 in that window and in all windows contained in it.

5.4 Estimand

Our estimand is similar to the original estimand. But the regression function and the propensity score are different.

$$\theta_{DMLSLR} = \frac{1}{N_w} \sum_{i:Z_i \in W_h} \mathbb{E}_w[Y_i(1) - Y_i(0)] \tag{5.4}$$

$$= \frac{1}{N_w} \sum_{i:Z_i \in W} \mathbb{E}_w[g(1, Z_i, X_i) - g(0, Z_i, X_i)] \tag{5.5}$$

$$= \frac{1}{N_w} \sum_{i:Z_i \in W} \mathbb{E}_w[\frac{T_i g(1, Z_i, X_i)}{m(Z_i, X_i)}] - \frac{1}{N_w} \sum_{i:Z_i \in W} \mathbb{E}_w[\frac{(1-T_i) g(0, Z_i, X_i)}{1 - m(Z_i, X_i)}] \tag{5.6}$$

The running variable Z_i affects the treatment variable via the propensity score m_0. And both Z_i and covariates X_i affect the potential outcome Y_i via the function g_0. Since both of these functions are unknown and potentially complicated, we can take advantage of machine learning methods to learn them.

For estimation of the ATE, we use

$$\psi(W; \theta, \eta) := (g(1, Z, X) - g(0, Z, X)) + \frac{T(Y - g(1, Z, X))}{m(Z, X)} - \frac{(1-T)(Y - g(0, Z, X))}{1 - m(Z, X)} - \theta, \tag{5.7}$$

where the nuisance parameter is $\eta = (m, g)$ consists of P-square-integrable functions g and m mapping the support of (T, Z, X) to $\mathbb{R}$ and the support of Z to $(\epsilon, 1-\epsilon)$, respectively, for some $\epsilon \in (0, 1/2)$. The true value of η is $\eta_0 = (m_0, g_0)$.

6 Simulation Study

To illustrate the methods developed in section 5, we consider four simulation examples. The first data set represents the data with clear discontinuity. The second data set represents the data without clear discontinuity. The third data set represents the high-dimensional data with clear discontinuity. And the

last data set represents the high-dimensional data without clear discontinuity. Here, high-dimensional data refers to datasets with a large number of features (large number of covariates X).

We first show the data generating process. We claim the distribution of each covariate and the regression function. Then, we show the process of window selection. We report results based on the conventional local randomization method and double machine learning methods. We label the conventional local randomization method as"OG", Lasso as "Lasso", random forest as "RF", and neural network as "NN". For "Lasso", we set $\lambda = 0.01$. The results in the "RF" column are obtained by estimating each nuisance function with a random forest with averages over 100 trees. To estimate the nuisance functions using the neural networks, we use 5 neurons, a decay parameter of 0.01, and a maximum number of iterations of 5. Besides the selected window, we also show the inference in a smaller window and a larger window to test the robustness of all the methods. To show which method is more robust in each window, we calculate the Mean Squared Error (MSE) for each estimator. We run the Monte Carlo simulation with 1000 repetitions for the calculation of MSE.

6.1 Dataset 1 - Data with clear discontinuity

This model employs the similar regression function form described in Imbens and Kalyanaraman (2012), which was generated using data from Lee (2008). Lee studies the incumbency advantage in elections, and thus his identification strategy was based on the discontinuity generated by the rule that the party with a majority vote share wins. The running variable is the difference in vote share between the Democratic candidate and his/her strongest opponent (usually Republican) in a given election. In this model, we have the cutoff $c = 0$. The regression function is obtained by fitting a fifth-order global polynomial with different coefficients for running variable below and above the cutoff. The resulting coefficients estimated on the Lee (2008) data, after discarding observations with past vote share differences greater than 0.99 and less than -0.99.

6.1.1 Data Generating Process

The data are generated as i.i.d. draws $i = 1, 2, ..., n$ with $n = 5000$ as follows:

$$Y_i = \mu_1(Z_i) + 0.50 * X_{1,i} - 0.10 * X_{2,i} + 0.20 * X_{3,i} - 0.30 * X_{4,i} + 0.66 * X_{5,i} - 0.05 * X_{6,i} + 0.25 * X_{7,i} - 0.15 * X_{8,i} + \epsilon_i \quad (6.1)$$

$$\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2) \quad (6.2)$$

$$X_{1,i} \sim \mathcal{N}(5, 3.14) \quad (6.3)$$

$$X_{2,i} \sim \mathcal{U}(-2.5, 2) \quad (6.4)$$

$$X_{3,i} \sim \mathcal{B}(5, 2) \quad (6.5)$$

$$X_{4,i} \sim Gamma(2, 0.5) \quad (6.6)$$

$$X_{5,i} \sim Poisson(2) \quad (6.7)$$

$$X_{6,i} \sim \mathcal{N}(0, 1.3) \quad (6.8)$$

$$X_{7,i} \sim Exp(1.2) \quad (6.9)$$

$$X_{8,i} \sim \mathcal{U}(-1, 3) \quad (6.10)$$

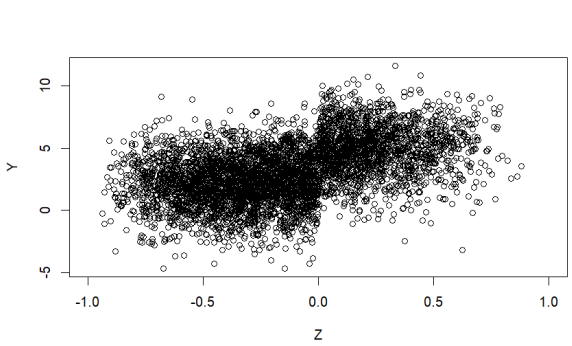
$$Z_i \sim 2\mathcal{B}(2, 4) - 1 \quad (6.11)$$

where $\mathcal{B}(\alpha, \beta)$ denotes a beta distribution with parameters α and β, $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2, $\mathcal{U}(a, b)$ denotes a uniform distribution with parameters a and b, Gamma(k, λ) denotes a Gamma distribution with shape parameter k and rate parameter λ, Poisson(λ) denotes a Poisson distribution with rate parameter λ, Exp(λ) denotes an Exponential distribution with rate

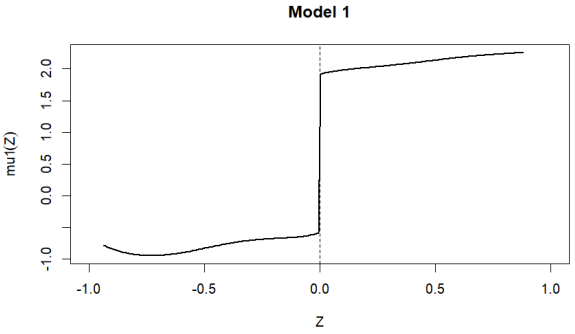
parameter λ, and $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ with $\sigma_\epsilon^2 = 0.1295$. The running variable Z_i is generated from a Beta$(3, 4)$ distribution, scaled and shifted to have support $[-1, 1]$.And the regression function follows:

$$\mu_1(z) = \begin{cases} -0.58 + 1.27z + 7.18z^2 + 20.21z^3 + 21.54z^4 + 7.33z^5, & \text{if } z < 0, \\ 1.92 + 0.84z - 3.00z^2 + 7.99z^3 - 9.01z^4 + 3.56z^5, & \text{if } z \geq 0. \end{cases} \tag{6.12}$$

This function introduces a discontinuity at the cutoff $c = 0$, with the true parameter θ_0 of 2.5.



(a) The distribution of generated dateset 1.



(b) A smooth curve of the distribution of generated dateset 1.

Figure 1: Generated data set 1 - data with clear discontinuity.

Figure 1 summarizes the distribution of the data set. We can clearly see that the data have a discontinuity on $Z = 0$. And according to the graph of smooth curve of the distribution of the generated data, we can see that the regression function has a big jump on $Z = 0$

6.1.2 Bandwidth Selection

We select out window using the method based on the rdlocarand package, rdwinselect function. The largest window we considered is [-0.87585, 0.87585], covering almost the entire support of our running variable. The smallest window is [-0.05, 0.05]. We analyze all symmetric windows around the cutoff between the [-0.05, 0.05] and [-0.87585, 0.87585] in increments of 0.00415 on each side of the cutoff. In each window, we perform randomization-based test of the sharp null hypothesis of no treatment effect for each of the predetermined covariates X_1 through X_8. As the default setting in rdwinselect, we set the minimum accepted value of the p-value from the covariate balance tests to be 0.15. And we use difference in means statistic as the test statistics in our randomization-based tests. The test is based on 1000 replications. For each window, we choose the minimum p-value across X_1 through X_8.

Figure 2 summarizes graphically the results of our window selector. The x-axis represents the upper limit of symmetric window [-w, w] around the cutoff, which is the absolute value of our running variable. For every symmetric window considered (x-axis), we plot the minimum p-value found in that window (y-axis). For example, the point 0.5 on the x-axis corresponds to the [-0.5, 0.5] window. The figure also shows the conventional significance level of 0.05 and the significance level of 0.15 that we use for implementation. Using significance level = 0.15, our chosen window is [-0.15790, 0.15790], since this is the largest window where the minimum p-value exceeds 15% in that window and all the windows contained in it.

The selected window is $[-0.15790, 0.15790]$. Table 1 shows the minimum p-value for the first three consecutive windows, $[-0.15790, 0.15790]$ and the next largest window, and last three consecutive windows. The minimum p-value of our chosen window is 0.187, and the minimum p-value of next largest window, $[-0.16205, 0.16205]$, is 0.116. From the Table 1 and Figure 2, we can see that p-value decreases rapidly after the selected window and keeps relatively almost surely.

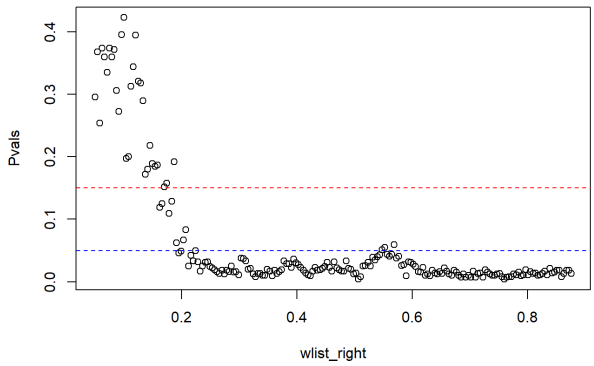


Figure 2: Window selector based on predetermined covariates.

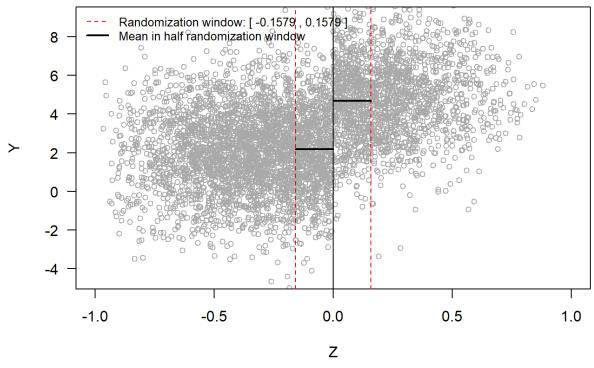


Figure 3: Figure of randomization-based estimation within the selected window

Table 1: Randomization-based p-value from the test for different windows

Window	**Minimum p-value**	**Number of observations inside the window**
$[-0.05000, 0.05000]$	0.296	462
$[-0.05415, 0.05415]$	0.368	503
$[-0.05830, 0.05830]$	0.256	549
$[-0.15790, 0.15790]$	0.187	1481
$[-0.16205, 0.16205]$	0.116	1508
$[-0.86755, 0.86755]$	0.018	4992
$[-0.87170, 0.87170]$	0.018	4992
$[-0.87585, 0.87585]$	0.013	4963

Table 2: Estimated Average Treatment Effect under Different Window

Window Property	**Window**	**OG**	**Lasso**	**RF**	**NN**
Smaller Window	[-0.0500, 0.0500]	2.56952	2.52611	2.57872	2.51248
Selected Window	[-0.1579, 0.1579]	2.49790	2.55973	2.56041	2.50989
Larger Window	[-0.8717, 0.8717]	2.75186	2.55425	2.70658	2.51781

6.1.3 Inference within the Window

Figure 3 shows the actual data we used in the inference of selected window. Table 2 lists out the estimation results. We noticed that all four methods produce ATE estimates that are relatively close to each other and generally near the likely true value of 2.5. Meanwhile, since the difference in bandwidth of Smaller Window and Selected Window is small, we received similar estimation across all 4 methods when inference in Selected Window. As the window grows larger, the estimates diverge. Lasso and NN estimates remain significantly stable and close to the true ATE of 2.5. RF estimate increases to 2.70658. And OG estimate remains higher at 2.75186.

Table 3 shows the result of MSE. The results of MSE support what we describe above. The MSE's of all three DML estimators are similar in the selected window, which the MSE of the conventional local randomization method is slightly higher. By the small difference between Smaller Window and Selected Window, MSE's of all four estimators in "smaller window" also follows the same concludsion in Selected Window. However, we can find a noticeable difference in MSE of all four estimators in "larger window". Overall, Lasso and NN consistently show the best performance in terms of MSE across all windows. The conventional method's accuracy deteriorates as the window size increases. RF performs better than the conventional method but its accuracy deteriorates significantly in larger window. Thus, the results suggest that the robustness of the DML methods does not affected by the choice of window size. While the conventional method's MSE increases with a misspecified window, the DML methods maintain low MSE, indicating their estimates remain accurate and precise even when the window is misspecified. This proves an advantage of using these DML techniques in RDD analysis.

6.2 Dataset 2 - Data without clear discontinuity

This dataset keeps the distribution of all covariates X_i identical to data set 1. We ensure that any observed differences in the outcome Y in the performance of models fitted to the data are directly contributed by the

Table 3: Monte Carlo MSE of each estimator

Window Property	**Window**	**OG**	**Lasso**	**RF**	**NN**
Smaller Window	[-0.0500, 0.0500]	0.01605	0.00234	0.00445	0.00012
Selected Window	[-0.1579, 0.1579]	0.02289	0.00624	0.00719	0.00171
Larger Window	[-0.8717, 0.8717]	0.04877	0.00422	0.04228	0.00197

change in the running variable Z. Thus, the only change in the data generating process is the constant term in the regression function. This setup mimics a controlled experiment and allows was for a direct comparison between the clear discontinuity and the unclear discontinuity.

6.2.1 Data Generating Process

The data are generated as i.i.d. draws $i = 1, 2, ..., n$ with $n = 5000$ as follows:

$$Y_i = \mu_2(Z_i) + 0.50 * X_{1,i} - 0.10 * X_{2,i} + 0.20 * X_{3,i} - 0.30 * X_{4,i} + 0.66 * X_{5,i} - 0.05 * X_{6,i} + 0.25 * X_{7,i} - 0.15 * X_{8,i} + \epsilon_i \quad (6.13)$$

$$\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2) \quad (6.14)$$

$$X_{1,i} \sim \mathcal{N}(5, 3.14) \quad (6.15)$$

$$X_{2,i} \sim \mathcal{U}(-2.5, 2) \quad (6.16)$$

$$X_{3,i} \sim \mathcal{B}(5, 2) \quad (6.17)$$

$$X_{4,i} \sim Gamma(2, 0.5) \quad (6.18)$$

$$X_{5,i} \sim Poisson(2) \quad (6.19)$$

$$X_{6,i} \sim \mathcal{N}(0, 1.3) \quad (6.20)$$

$$X_{7,i} \sim Exp(1.2) \quad (6.21)$$

$$X_{8,i} \sim \mathcal{U}(-1, 3) \quad (6.22)$$

$$Z_i \sim 2\mathcal{B}(2, 4) - 1 \quad (6.23)$$

where $\mathcal{B}(\alpha, \beta)$ denotes a beta distribution with parameters α and β, $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2, $\mathcal{U}(a, b)$ denotes a uniform distribution with parameters a and b, Gamma(k, λ) denotes a Gamma distribution with shape parameter k and rate parameter λ, Poisson(λ) denotes a Poisson distribution with rate parameter λ, Exp(λ) denotes an Exponential distribution with rate parameter λ, and $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ with $\sigma_\epsilon^2 = 0.1295$. The running variable Z_i is generated from a Beta$(3, 4)$ distribution, scaled and shifted to have support $[-1, 1]$. And the regression function follows:

$$\mu_2(z) = \begin{cases} 0.4 + 1.27z + 7.18z^2 + 20.21z^3 + 21.54z^4 + 7.33z^5, & \text{if } z < 0, \\ 0.42 + 0.84z - 3.00z^2 + 7.99z^3 - 9.01z^4 + 3.56z^5, & \text{if } z \geq 0. \end{cases} \quad (6.24)$$

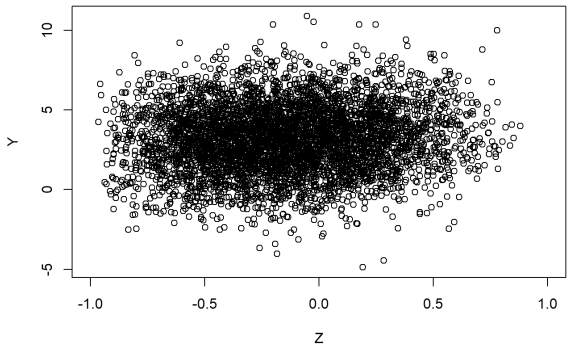
This function introduces a discontinuity at the cutoff $c = 0$, with a the true parameter θ_0 of 0.02.

Figure 4 summarizes the distribution of the data set. Compared to data set 1, we cannot see the discontinuity at $Z = 0$. In terms of the smooth curve of the distribution of generated data set 2, the curve is relatively smooth at $Z = 0$.

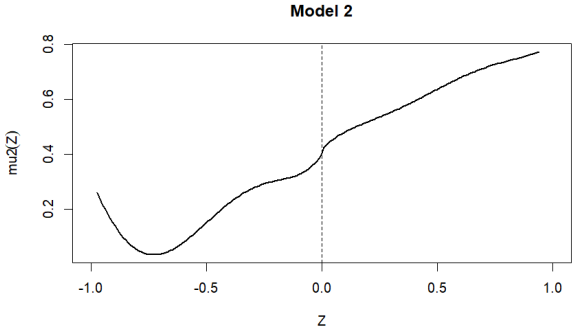
6.2.2 Bandwidth Selection

We select out window using the method based on the rdlocarand package, rdwinselect function. The largest window we considered is [-0.816, 0.816], covering almost the entire support of our running variable. The smallest window is [-0.02, 0.02]. We analyze all symmetric windows around the cutoff between the [-0.02, 0.02] and [-0.816, 0.816] in increments of 0.004 on each side of the cutoff. In each window, we perform randomization-based test of the sharp null hypothesis of no treatment effect for each of the predetermined covariates X_1 through X_8. As the default setting in rdwinselect, we set the minimum accepted value of the p-value from the covariate balance tests to be 0.15. And we use difference in means statistic as the test statistics in our randomization-based tests. The test is based on 1000 replications. For each window, we choose the minimum p-value across X_1 through X_8.

Figure 5 summarizes graphically the results of our window selector. The selected window is [-0.036, 0.036]. Notably, the mean of the potential outcome Y calculated for observations immediately to the left of the cutoff $Z = 0$ is very close to the mean calculated immediately to the right. This aligns with the regression



(a) The distribution of generated dateset 2.



(b) A smooth curve of the distribution of generated dateset 2.

Figure 4: Generated dataset 2 - data with clear discontinuity.

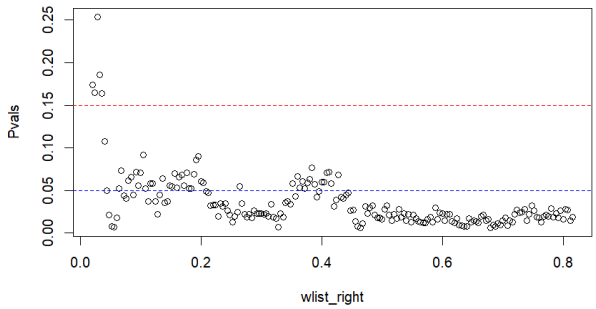


Figure 5: Window selector based on predetermined covariates.

Table 4: Randomization-based p-value from the test for different windows

Window	**Minimum p-value**	**Number of observations inside the window**
$[-0.020, 0.020]$	0.174	183
$[-0.024, 0.024]$	0.165	223
$[-0.028, 0.028]$	0.254	261
$[-0.036, 0.036]$	0.164	331
$[-0.040, 0.040]$	0.108	357
$[-0.808, 0.808]$	0.027	4926
$[-0.812, 0.812]$	0.015	4931
$[-0.816, 0.816]$	0.019	4935

function $\mu_2(z)$ whose values of the constant differs slightly on both side of the cutoff. Therefore, since the true jump in the regression function at the cutoff is small, the estimated treatment effect is expected to be correspondingly small.

The selected window is $[-0.036, 0.036]$. Table 4 shows the minimum p-value for the first three consecutive windows, $[-0.036, 0.036]$ and the next largest window, and last three consecutive windows. The trend of p-value is similar to the trend of $p$value in data set 1.

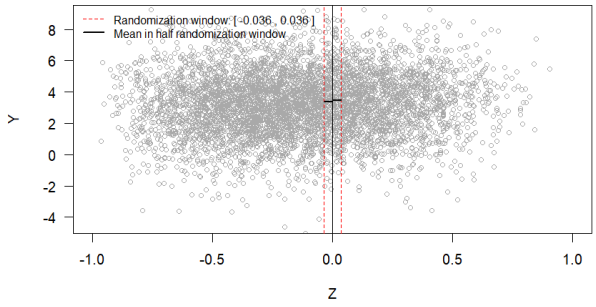


Figure 6: Figure of randomization-based estimation within the selected window

6.2.3 Inference within the Window

Table 5: Estimated Average Treatment Effect under Different Window

Window Property	Window	OG	Lasso	RF	NN
Smaller Window	[-0.020, 0.020]	0.32828	0.00453	0.09842	0.02788
Selected Window	[-0.036, 0.036]	0.12335	0.04190	0.08255	0.06316
Larger Window	[-0.816, 0.816]	0.24490	0.07644	0.27288	0.05307

Table 6: Monte Carlo MSE of each estimator

Window Property	Window	OG	Lasso	RF	NN
Smaller Window	[-0.020, 0.020]	0.11395	0.00067	0.07039	0.05726
Selected Window	[-0.036, 0.036]	0.03685	0.00109	0.01823	0.02945
Larger Window	[-0.816, 0.816]	0.06238	0.00403	0.03175	0.00084

Figure 6 shows the actual data we used in inference of the selected window. Table 5 lists out the estimation results. The results of this data set vary. Under Smaller Window, the Lasso estimator yields a point estimate 0.00453 very close to the true value, which is 0.02. The NN estimate is also close. The RF estimate is farther from the true value with small bias. However, the OG estimate shows a big bias. In Selected Window, all methods now produce estimates somewhat higher than the true ATE. Lasso, NN, and RF are moderately biased , while the OG estimate still remains the furthest from the true value. In Larger Window, all estimates still appear biased relative to 0.02. Lasso and NN show less bias than RF and OG. In terms of MSE, Lasso achieves the lowest MSE, showing the best overall performance. NN performs very well in the Larger Window but less well in the smaller windows compared to Lasso. RF has relatively high MSE, particularly in the smallest window. The OG estimator generally has the highest or second-highest MSE. The results highlight the potential benefits of using DML methods in this RDD without a clear discontinuity scenario.

6.3 Dataset 3 - High dimensional Data with clear discontinuity

We generate a high-dimensional dataset for the RDD scenario with a clear discontinuity. We introduce some correlation structure among covariates, making the dataset more realistic than independent covariates. To reduce dimensionality, we implemented different variable selection strategies for each machine learning method. For Lasso, we use its built-in L1 regularization to automatically identify non-zero coefficients through cross-validated lambda selection. With RF, we calculate permutation-based variable importance measures and establish a threshold to identify covariates with meaningful predictive power. For NN, we use

a permutation approach that measured each variable's contribution to quantify each coefficient's impact on the outcome prediction.

6.3.1 Data Generating Process

The data are generated as i.i.d. draws for $i = 1, 2, ..., n$, with sample size $n = 5000$, total number of covariates $p = 100$, and number of active covariates $p_{\text{active}} = 10$, as follows:

$$Y_i = \mu_3(Z_i) + \sum_{k=1}^{p} X_{k,i}\beta_k + \epsilon_i \tag{6.25}$$

$$\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2) \tag{6.26}$$

$$\mathbf{X}_i = (X_{1,i}, \ldots, X_{p,i}) \sim \mathcal{N}_p(\mathbf{0}, \Sigma) \tag{6.27}$$

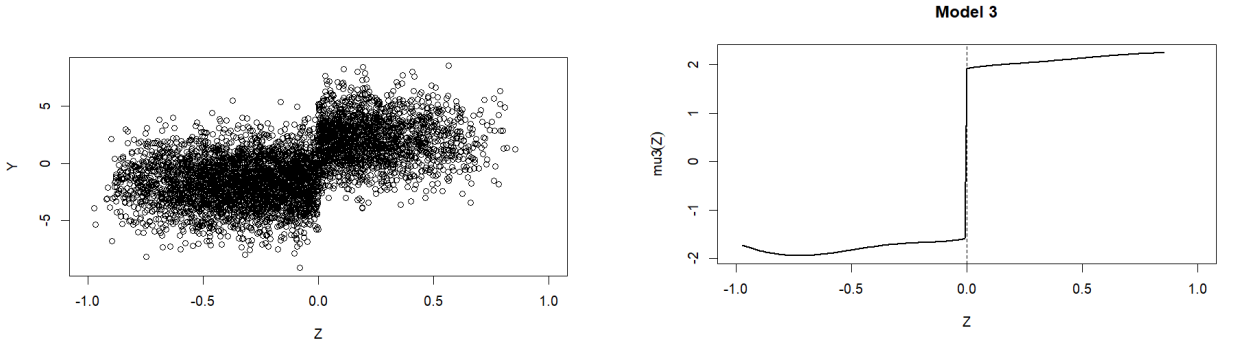
$$Z_i \sim 2\mathcal{B}(2, 4) - 1 \tag{6.28}$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2, $\mathcal{N}_p(\mu, \Sigma)$ denotes a p-variate normal distribution with mean vector μ and $p \times p$ covariance matrix Σ, and $\mathcal{B}(\alpha, \beta)$ denotes a beta distribution with shape parameters α and β. The error ϵ_i has distribution $\mathcal{N}(0, \sigma_\epsilon^2)$ with $\sigma_\epsilon^2 = 0.1295$. The running variable Z_i is generated from a Beta$(3, 4)$ distribution, scaled and shifted to have support $[-1, 1]$.

The $p \times p$ covariance matrix Σ (with $p = 100$) has an autoregressive AR(1) structure with elements $\Sigma_{jk} = \rho^{|j-k|}$, where the correlation parameter is $\rho = 0.5$. The p-dimensional coefficient vector $\beta = (\beta_1, \ldots, \beta_p)$ is sparse. The first $p_{\text{active}} = 10$ coefficients are drawn independently from a uniform distribution, $\beta_k \sim U(-1, 1)$ for $k = 1, \ldots, 10$. The remaining $p - p_{\text{active}} = 90$ coefficients are set to zero, $\beta_k = 0$ for $k = 11, \ldots, 100$. And the regression function follows:

$$\mu_3(z) = \begin{cases} -1.58 + 1.27z + 7.18z^2 + 20.21z^3 + 21.54z^4 + 7.33z^5, & \text{if } z < 0, \\ 1.92 + 0.84z - 3.00z^2 + 7.99z^3 - 9.01z^4 + 3.56z^5, & \text{if } z \geq 0. \end{cases} \tag{6.29}$$

This function introduces a discontinuity at the cutoff $Z = 0$, with a the true parameter θ_0 of 3.5.



(a) The distribution of generated dateset 3.

(b) A smooth curve of the distribution of generated dateset 3.

Figure 7: Generated dataset 3 - data with clear discontinuity.

Figure 7 summarizes the distribution of the data set. In terms of the smooth curve of the distribution of generated dataset 3, the curve is relatively smooth at $Z = 0$.

6.3.2 Bandwidth Selection

Before window selection, we use random forest feature importance to reduce dimensionality. We create a function that fits a random forest model with 500 trees to identify which covariates have the strongest

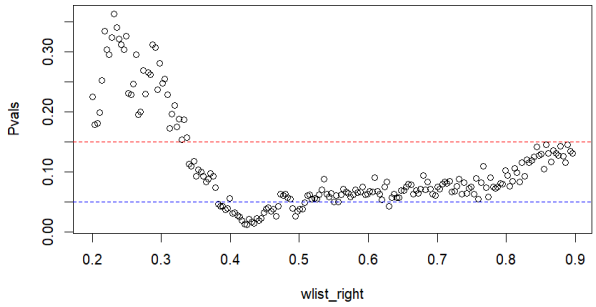


Figure 8: Window selector based on predetermined covariates.

predictive relationship with the outcome variable Y. Then, we calculate importance scores for all covariates X and select the top 15 most influential features based on their importance rankings. Finally, we use these 15 selected covariates for the hypothesis test in window selection. We select out window using the method based on the rdlocarand package, rdwinselect function. The largest window we considered is [-0.8965, 0.8965], covering almost the entire support of our running variable. The smallest window is [-0.200, 0.200]. We analyze all symmetric windows around the cutoff between the [-0.200, 0.200] and [-0.8965, 0.8965] in increments of 0.0035 on each side of the cutoff. We perform randomization-based test of the sharp null hypothesis of no treatment effect for each of the predetermined covariates for selected covariates in each window. As the default setting in rdwinselect, we set the minimum accepted value of the p-value from the covariate balance tests to be 0.15. And we use difference in means statistic as the test statistics in our randomization-based tests. The test is based on 1000 replications. For each window, we choose the minimum p-value across all selected covariates.

Table 7: Randomization-based p-value from the test for different windows

Window	**Minimum p-value**	**Number of observations inside the window**
$[-0.2000, 0.200]$	0.174	1905
$[-0.2035, 0.2035]$	0.165	1926
$[-0.2070, 0.2070]$	0.254	1960
$[-0.3365, 0.3365]$	0.164	2942
$[-0.3400, 0.3400]$	0.108	2968
$[-0.8895, 0.8895]$	0.145	4985
$[-0.8930, 0.8930]$	0.134	4986
$[-0.8965, 0.8965]$	0.130	4986

The selected window is $[-0.3365, 0.3365]$. Table 7 shows the minimum p-value for the first three consecutive windows, $[-0.3365, 0.3365]$ and the next largest window, and last three consecutive windows. The minimum p-value of our chosen window is 0.164, and the minimum p-value of next largest window, $[-0.3400, 0.3400]$, is 0.108. From the Table 7 and Figure 8, we can see that p-value decreases rapidly after the selected window. But the p-values increase gradually after the window around $[-0.43, 0.43]$. This suggest that the conventional window selection method would not be the desired window selection method in this case.

6.3.3 Inference within the Window

Figure 9 shows the actual data we used in inference of the selected window. Table 8 lists out the estimation results. The results of this data set vary. Under Smaller Window, the Lasso estimator yields a point estimate 3.56538, which is very close to the true value. The RF estimate is farther from the true value with small

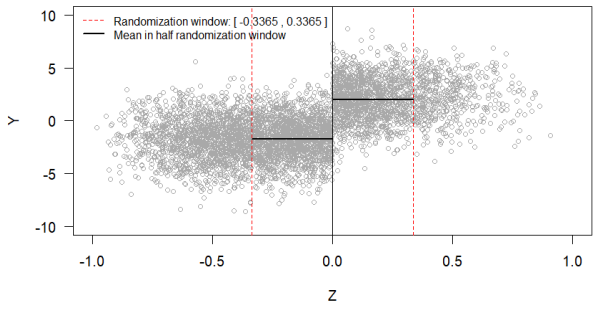


Figure 9: Figure of randomization-based estimation within the selected window

Table 8: Estimated Average Treatment Effect under Different Window

Window Property	**Window**	**OG**	**Lasso**	**RF**	**NN**
Smaller Window	[-0.2000, 0.2000]	3.69378	3.57532	3.65952	3.40026
Selected Window	[-0.3365, 0.3365]	3.74410	3.58470	3.68385	3.70978
Larger Window	[-0.8965, 0.8965]	3.85901	3.57524	3.80864	3.89268

Table 9: Monte Carlo MSE of each estimator

Window Property	**Window**	**OG**	**Lasso**	**RF**	**NN**
Smaller Window	[-0.2000, 0.2000]	0.03763	0.00405	0.01909	0.00201
Selected Window	[-0.3365, 0.3365]	0.05762	0.00532	0.03320	0.01863
Larger Window	[-0.8965, 0.8965]	0.12980	0.00407	0.08046	0.47929

bias. This aligns the results in the previous two dataset. The OG and NN estimate show larger biases. In Selected Window, all methods now produce estimates higher than the true ATE. Notably, the bias across all four methods get larger than that in Smaller window. This result aligns the conclusion in window selection– the conventional window selection method may not be desired under high-dimensional dataset. In Larger Window, all estimates still appear biased relative to 3.5. But we can see a decrease in bias for Lasso. In terms of MSE, Lasso achieves the lowest MSE, showing the best overall performance. RF performs very well in the Smaller Window but less well in the Larger Window. The OG and NN estimator generally have the highest MSE. The results show that we need to pay attention to method selection under high dimensional dataset since not all DML methods have a better bias-variance tradeoff than the conventional method. Though there are the potential benefits of using DML methods in high-dimensional RDDs, we have to always remind ourselves that method properties are more important than naively choosing any high-dimensional approach.

6.4 Dataset 4 - High dimensional Data without clear discontinuity

Similar to the relationship between Dataset 1 and Dataset2, this dataset keeps the distribution of all covariates X_i identical to data set 3. We ensure that any observed differences in the outcome Y in the performance of models fitted to the data are directly contributed by the change in the running variable Z. Thus, the only change in the data generating process is the constant term in the regression function. We use the same dimensionality reduction method as we used in dataset 3.

Though NN performs well in the previous three datasets, it fails in this scenario. Despite their theoretical power, neural networks struggle with overfitting and fail to distinguish the weak signal, which is the unclear discontinuity in this dataset, from the surrounding noise dimensions. On the other hand, Lasso's feature selection capabilities and Random Forest's balanced flexibility proved crucial for isolating the small true treatment effect in high-dimensional noise.

6.4.1 Data Generating Process

The data are generated as i.i.d. draws for $i = 1, 2, ..., n$, with sample size $n = 5000$, total number of covariates $p = 100$, and number of active covariates $p_{\text{active}} = 10$, as follows:

$$Y_i = \mu_4(Z_i) + \sum_{k=1}^{p} X_{k,i}\beta_k + \epsilon_i \tag{6.30}$$

$$\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2) \tag{6.31}$$

$$\mathbf{X}_i = (X_{1,i}, \ldots, X_{p,i}) \sim \mathcal{N}_p(\mathbf{0}, \Sigma) \tag{6.32}$$

$$Z_i \sim 2\mathcal{B}(2, 4) - 1 \tag{6.33}$$

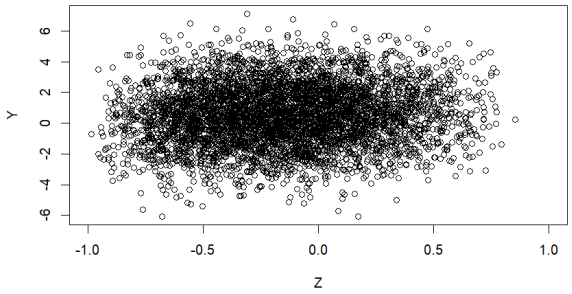
where $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2, $\mathcal{N}_p(\mu, \Sigma)$ denotes a p-variate normal distribution with mean vector μ and $p \times p$ covariance matrix Σ, and $\mathcal{B}(\alpha, \beta)$ denotes a beta distribution with shape parameters α and β. The error ϵ_i has distribution $\mathcal{N}(0, \sigma_\epsilon^2)$ with $\sigma_\epsilon^2 = 0.1295$. The running variable Z_i is generated from a Beta$(3, 4)$ distribution, scaled and shifted to have support $[-1, 1]$.

The $p \times p$ covariance matrix Σ (with $p = 100$) has an autoregressive AR(1) structure with elements $\Sigma_{jk} = \rho^{|j-k|}$, where the correlation parameter is $\rho = 0.5$. The p-dimensional coefficient vector $\beta = (\beta_1, \ldots, \beta_p)$ is sparse. The first $p_{\text{active}} = 10$ coefficients are drawn independently from a uniform distribution, $\beta_k \sim U(-1, 1)$ for $k = 1, \ldots, 10$. The remaining $p - p_{\text{active}} = 90$ coefficients are set to zero, $\beta_k = 0$ for $k = 11, \ldots, 100$. And the regression function follows:

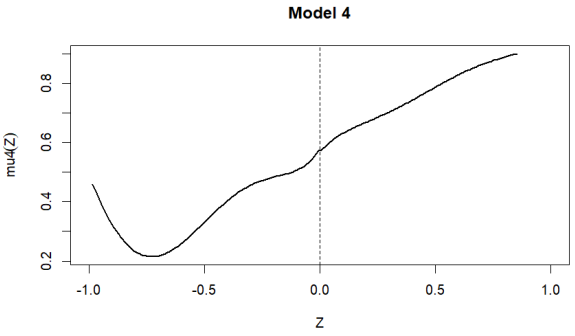
$$\mu_4(z) = \begin{cases} 0.58 + 1.27z + 7.18z^2 + 20.21z^3 + 21.54z^4 + 7.33z^5, & \text{if } z < 0, \\ 0.57 + 0.84z - 3.00z^2 + 7.99z^3 - 9.01z^4 + 3.56z^5, & \text{if } z \geq 0. \end{cases} \tag{6.34}$$

This function introduces a discontinuity at the cutoff $Z = 0$, with a the true parameter θ_0 of 0.01. In terms of the smooth curve of the distribution of generated data set 2, the curve is relatively smooth at $Z = 0$.

Figure 10 summarizes the distribution of the data set. In terms of the smooth curve of the distribution of generated dataset 4, the curve is relatively smooth at $Z = 0$.



(a) The distribution of generated dateset 4.



(b) A smooth curve of the distribution of generated dateset 4.

Figure 10: Generated dataset 4 - data with clear discontinuity.

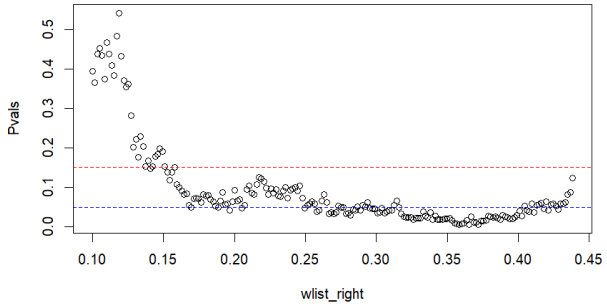


Figure 11: Window selector based on predetermined covariates.

6.4.2 Bandwidth Selection

Similarly, we use random forest feature importance to reduce dimensionality. We select out window using the method based on the rdlocarand package, rdwinselect function. The largest window we considered is [-0.4383, 0.4383], covering only the half of the support of our running variable.The smallest window is [-0.1000, 0.1000]. We analyze all symmetric windows around the cutoff between the [-0.1000, 0.1000] and [-0.4383, 0.4383] in increments of 0.0017 on each side of the cutoff. We perform randomization-based test of the sharp null hypothesis of no treatment effect for each of the predetermined covariates for all selected covariates in each window. We set the minimum accepted value of the p-value from the covariate balance tests to be 0.15. And we use difference in means statistic as the test statistics in our randomization-based tests. The test is based on 1000 replications. For each window, we choose the minimum p-value across all selected covariates. If we consider windows larger than [-0.4383, 0.4383], the p-value of the tests would be greater than 0.15 and fails the whole process.

The selected window is $[-0.1391, 0.1391]$. Table 10 shows the minimum p-value for the first three consecutive windows, $[-0.1391, 0.1391]$ and the next largest window, and last three consecutive windows. The minimum p-value of our chosen window is 0.168, and the minimum p-value of next largest window, $[-0.1408, 0.1408]$, is 0.147. From the Table 10 and Figure 11, we can see that p-value decreases rapidly after the selected window and keeps relatively almost surely.

Table 10: Randomization-based p-value from the test for different windows

Window	Minimum p-value	Number of observations inside the window
$[-0.020, 0.020]$	0.394	940
$[-0.024, 0.024]$	0.365	958
$[-0.028, 0.028]$	0.438	977
$[-0.1391, 0.1391]$	0.168	1324
$[-0.1408, 0.1408]$	0.147	1339
$[-0.4349, 0.4349]$	0.81	3640
$[-0.4366, 0.4366]$	0.87	3650
$[-0.4383, 0.4383]$	0.124	3668

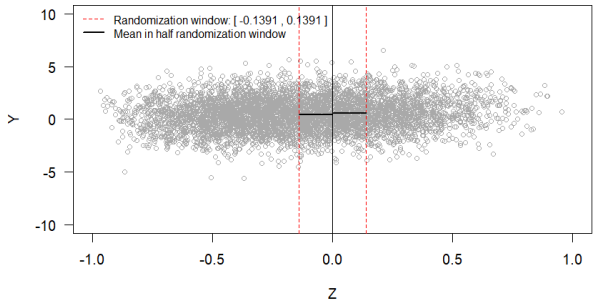


Figure 12: Figure of randomization-based estimation within the selected window

6.4.3 Inference within the Window

Figure 12 shows the actual data we used in inference of the selected window. Table 11 lists out the estimation results. The Lasso estimator demonstrated the highest accuracy. It has the closest estimated ATE to the true ATE of 0.01 across all three windows. The OG estimator consistently exhibited positive bias across all windows. The RF estimator also showed positive bias, but generally performing better than OG and worse than Lasso in terms of bias. Lasso significantly outperformed the other methods in terms of MSE, achieving the lowest MSE across all three window. This indicates Lasso provided the best combination of low bias and low variance. RF was the second-best performance with MSEs, which are lower than the MSEs' of conventional method. The OG estimator consistently yields the highest MSE, reflecting its relative inaccuracy and bias. Finally, the failure of NN serves as a practical reminder that method selection should be guided by problem structure rather than model complexity. The bias-variance tradeoff remains a fundamental consideration in causal inference settings where accurate effect estimation is the primary goal.

Table 11: Estimated Average Treatment Effect under Different Window

Window Property	Window	OG	Lasso	RF	NN
Smaller Window	[-0.1000, 0.1000]	0.09622	0.0453	0.04865	Fail
Selected Window	[-0.1391, 0.1391]	0.09171	0.044990	0.07329	Fail
Larger Window	[-0.4383, 0.4383]	0.15145	0.05884	0.15499	Fail

7 Discussion and Conclusion

Regression discontinuity designs (RDDs) are a common quasi-experiment in economics, education, political science, statistics, and biological statistics. Though the most popular methodologies for estimating casual

Table 12: Monte Carlo MSE of each estimator

Window Property	Window	OG	Lasso	RF	NN
Smaller Window	[-0.1000, 0.1000]	0.02461	0.00126	0.01098	Fail
Selected Window	[-0.1391, 0.1391]	0.02219	0.00213	0.01229	Fail
Larger Window	[-0.4383, 0.4383]	0.02703	0.00371	0.02539	Fail

effect is an RDD relying on continuity assumptions, the local randomization framework for RDDs has been more frequently discussed in recent literature. The local randomization framework views the running variable as stochastic, which introducing randomness to the assignment mechanism.

Double/Debiased Machine Learning (DML) is a powerful tool for the inference on high-dimensional parameters. It removes the impact of regularization bias and overfitting on estimation of the parameter of interest θ_0, which are caused by naively plugging ML estimators of nuisance parameter into estimation equations for θ_0. DML delivers point estimators that concentrated in a $n^{-1/2}$ neighborhood of the true parameter and are approximately unbiased and normally distributed, which allows construction of valid confidence statements.

In this paper, we provided a review of both local randomization framework for RDDs and DML. Then, we showed the application of DML in local randmization framework for RDDs. We declared the theoretical feasibility and the benefit of the application. We claimed the assumptions of the application. Finally, we provided the bandwidth selection algorithm and the estimand. We used four simulations on different datasets–data with clear discontinuity, data without clear discontinuity, high-dimensional data with clear discontinuity, and high-dimensional data without clear discontinuity–to show the advantage of our proposed method. The DML approach proved remarkably superior to the conventional local randomization method when estimating both small treatment effect and large treatment effect, under low-dimensional dataset or high-dimensional datasets. The application of DML on local randomization framework allowed us to effectively control the confounding bias while maintaining relatively low variance, resulting in a lower MSE than the conventional local randomization method. Our proposed method suggests a promising line for future research that explores a more robust method to estimate ATE under local randomization framework.

8 Acknowledgments

I would like to express my gratitude to Dr.Jelena Bradic, my honor graduation advisor. I would never complete my honor thesis without her support. Her guidance, help, and feedback have made me into a researcher, more than an undergraduate student, I am today. I also would like to acknowledge my friend, Xinghan, for his help.

References

Bradic, J., Wager, S., Zhu, Y. (2019). Sparsity double robust inference of average treatment effects (arXiv:1905.00744). arXiv. http://arxiv.org/abs/1905.00744

Branson, Zach, and Fabrizia Mealli. The Local Randomization Framework for Regression Discontinuity Designs: A Review and Some Extensions. arXiv:1810.02761, arXiv, 5 Nov. 2019. arXiv.org, https://doi.org/10.48550/arXiv.1810.02761.

Breiman, Leo, editor. Classification and Regression Trees. 1. CRC Press repr, Chapman & Hall/CRC, 1998.

Cattaneo, Matias D., Brigham R. Frandsen, et al. 'Randomization Inference in the Regression Discontinuity Design: An Application to Party Advantages in the U.S. Senate'. Journal of Causal Inference, vol. 3, no. 1, Mar. 2015, pp. 1–24. DOI.org (Crossref), https://doi.org/10.1515/jci-2013-0010.

Cattaneo, Matias D., et al. 'Inference in Regression Discontinuity Designs under Local Randomization'. The Stata Journal: Promoting Communications on Statistics and Stata, vol. 16, no. 2, June 2016, pp. 331–67. DOI.org (Crossref), https://doi.org/10.1177/1536867X1601600205.

Cattaneo, Matias D., and Rocío Titiunik. 'Regression Discontinuity Designs'. Annual Review of Economics, vol. 14, no. 1, Aug. 2022, pp. 821–51. DOI.org (Crossref), https://doi.org/10.1146/annurev-economics-051520-021409.

Chernozhukov, Victor, et al. 'Double/Debiased Machine Learning for Treatment and Structural Parameters'. The Econometrics Journal, vol. 21, no. 1, Feb. 2018, pp. C1–68. DOI.org (Crossref), https://doi.org/10.1111/ectj.12097.

Imbens, G., and K. Kalyanaraman. 'Optimal Bandwidth Choice for the Regression Discontinuity Estimator'. The Review of Economic Studies, vol. 79, no. 3, July 2012, pp. 933–59. DOI.org (Crossref), https://doi.org/10.1093/restud/rdr043.

Imbens, Guido W., and Thomas Lemieux. 'Regression Discontinuity Designs: A Guide to Practice'. Journal of Econometrics, vol. 142, no. 2, Feb. 2008, pp. 615–35. DOI.org (Crossref), https://doi.org/10.1016/j.jeconom.2007.05.001.

Lee, David S. 'Randomized Experiments from Non-Random Selection in U.S. House Elections'. Journal of Econometrics, vol. 142, no. 2, Feb. 2008, pp. 675–97. DOI.org (Crossref), https://doi.org/10.1016/j.jeconom.2007.05.004.

Matias and Rocio Cattaneo, Matias D., and Rocío Titiunik. 'Regression Discontinuity Designs'. Annual Review of Economics, vol. 14, no. 1, Aug. 2022, pp. 821–51. DOI.org (Crossref), https://doi.org/10.1146/annurev-economics-051520-021409.

Ripley, Brian D. Pattern Recognition and Neural Networks. 1st ed., Cambridge University Press, 1996. DOI.org (Crossref), https://doi.org/10.1017/CBO9780511812651.

Rubin, D. B. (2008). For objective causal inference, design trumps analysis. The Annals of Applied Statistics, 2(3). https://doi.org/10.1214/08-AOAS187

Tibshirani, Robert. 'Regression Shrinkage and Selection Via the Lasso'. Journal of the Royal Statistical Society Series B: Statistical Methodology, vol. 58, no. 1, Jan. 1996, pp. 267–88. DOI.org (Crossref), https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

Thistlethwaite, D. L., Campbell, D. T. (1960). Regression-discontinuity analysis: An alternative to the ex post facto experiment. Journal of Educational Psychology, 51(6), 309–317. https://doi.org/10.1037/h0044319

Villamizar-Villegas, M., Pinzon-Puerto, F. A., Ruiz-Sanchez, M. A. (2022). A comprehensive history of regression discontinuity designs: An empirical survey of the last 60 years. Journal of Economic Surveys, 36(4), 1130–1178. https://doi.org/10.1111/joes.12461