Comparing Two Bootstrapping Methods for Calculating Standard Errors in Regression Discontinuity Designs

John Deke and Lisa Dragoset
Mathematica Policy Research

Abstract

Calculating impacts in a study that utilizes a regression discontinuity design (RDD) can be a technically complex affair. Bandwidth estimation, calculating “fuzzy” impacts, accounting for clustering of observations by unique values of the assignment variable, and incorporating multiple assignment variables or cutoffs all add to the complexity of estimation. When confronted with technically complex estimation methods researchers often turn to non-parametric bootstrapping as a simple and robust way to calculate standard errors. In this paper we use Monte Carlo simulations to examine the performance of non-parametric bootstrapping for RDD impact estimation scenarios that incorporate all of the estimation complexities described above. We also propose and examine a residual bootstrapping algorithm that is conceptually motivated by the idea that the RDD is akin to random assignment of residuals conditional on the assignment variable. The data generating processes used for the simulations are taken from previous Monte Carlo examinations of RDD estimation techniques found in the literature, as well as data generating processes that mimic data from our own empirical work in education. We find that both non-parametric bootstrapping and the proposed residual bootstrapping algorithm successfully control the type 1 error rate at the desired level, but that the residual bootstrapping algorithm can provide greater statistical power (that is, non-parametric bootstrapping can yield standard error estimates that are too conservative).
I. Introduction

The regression discontinuity design (RDD) is an appealing nonexperimental alternative to the randomized controlled trial (RCT) because it involves a mechanism for assignment to treatment and control conditions that is observed by the researcher. Because the assignment mechanism is observed, it is often possible to make a credible adjustment for the relationship between the assignment mechanism and outcomes, yielding an asymptotically unbiased impact estimate.

Under RDD, individuals are selected to receive an intervention based on where they fall relative to a cutoff value on a continuous assignment variable. If the relationship between the assignment variable and the outcome is continuous in the absence of the intervention, then a discontinuity in the outcome-assignment relationship at the cutoff value can be interpreted as the impact of the intervention, so long as the researcher is able to appropriately adjust for the relationship between the outcome and the assignment variable. See Cook (2008) for a thorough account of the multidisciplinary history of RDD.

While studies employing RDD have the potential to generate credible impacts, calculating credible RDD impacts can be a technically complex affair. Bandwidth estimation (for example, Imbens & Kalyanaraman [IK], 2012), calculating “fuzzy” impacts (Hahn, Todd, and Van Der Klaauw 2001), accounting for clustering of observations by unique values of the assignment variable (Lee and Card 2008), and incorporating multiple assignment variables or cutoffs (Reardon and Robinson 2010; Wong, Steiner, and Cook 2013) all add to the complexity of estimation.

Calculating the standard error of an RDD impact for the purpose of hypothesis testing can be even more challenging, particularly when several of the estimation complexities described above interact in the same study. When confronted with technically complex estimation methods researchers often turn to bootstrapping as a simple and robust way to calculate standard errors (Efron and Tibshirani 1993). Perhaps the most widely applicable approach to bootstrapping is the non-parametric bootstrap in which (1) observations in the data are resampled with replacement, (2) the estimate of interest is calculated on the resampled data, (3) steps 1 and 2 are repeated a large number of times, and (4) the standard deviation of the repeated estimates is used as the estimate of the standard error of the original estimate. Another approach to bootstrapping that is specific to regression analysis is the residual bootstrap in which only regression residuals (rather than entire cases) are resampled with replacement in step (1).

In this paper we use Monte Carlo simulations to examine the performance of nonparametric bootstrapping for RDD impact estimation scenarios that incorporate all of the estimation complexities described above. We also propose and examine a residual bootstrapping algorithm that is conceptually motivated by the idea that the RDD is akin to random assignment of residuals conditional on the assignment variable. The data generating processes used for the simulations are taken from previous Monte Carlo examinations of RDD estimation techniques found in the literature, as well as data generating processes that mimic data from our own empirical work in education. We find that both non-parametric bootstrapping and the proposed residual bootstrapping algorithm successfully control the type 1 error rate at the desired level, but
that the residual bootstrapping algorithm can provide greater statistical power because non-parametric bootstrapping can yield standard error estimates that are too conservative.

II. Two Bootstrap Algorithms

We examine two bootstrap algorithms – (1) the standard non-parametric bootstrap and (2) a customized version of the residual bootstrap. First we describe how we apply the standard nonparametric bootstrap in the RDD context. Second we describe our customized residual bootstrap algorithm.

Applying the Nonparametric Bootstrap in the RDD Context

The algorithm we use to apply the nonparametric bootstrap is a straightforward implementation of the usual approach to nonparametric bootstrapping with clustered data:

1. For all observations above the RDD cutoff value (that is, not just observations within a bandwidth), select a random sample (with replacement) of unique values of the RDD assignment variable. Include all observations with the sampled values of the assignment variable in the selected sample.\(^1\) Repeat this sampling approach for observations below the RDD cutoff value.\(^2\)

2. Apply all RDD estimation methods to the selected sample. This includes all steps in RDD impact estimation -- the selection of the bandwidth (we use the 1K bandwidth), accounting for multiple assignment variables, and estimating fuzzy impacts. Save the RDD impact.

3. Repeat steps 1 and 2 N times.

4. Calculate the standard error of the impact estimate as the standard deviation of the N impact estimates.

Customized Residual Bootstrap for the RDD Context

The residual bootstrap algorithm we use is customized to the RDD context:

1. Aggregate all variables by unique values of the assignment variable. For example, to create the aggregated version of the outcome variable we calculate the average value of the outcome for each unique value of the assignment variable. The number of observations in the aggregated data set is equal to the number of unique values of the assignment variable. We do this to address clustering of observations within unique values of the assignment variable.

---

\(^1\) We sample unique values of the assignment variable to address the clustering of observations within unique values of the assignment variable (Lee and Card 2009).

\(^2\) We sample separately above and below the cutoff so that in small samples we are assured of sample being selected both above and below the cutoff.
2. Using all of the aggregated observations (that is, not just observations within a bandwidth), estimate a regression of the outcome of interest on the following variables:

- An indicator (or indicators, in the case of multiple assignment variables) of whether an observation is above or below the cutoff value of the assignment variable(s)
- The assignment variable(s)
- The square of the assignment variable(s)
- Interactions of the assignment variable(s) and the square of the assignment variable(s) with indicator(s) for whether an observation is above or below the cutoff value of the assignment variable(s).
- Any additional covariates that will be included in impact analysis (for example, to increase precision)

In the case of a fuzzy analysis also estimate a linear regression of the treatment participation indicator on the same variables as above.3

3. Calculate predicted outcomes using the coefficient estimates from step 2. These predicted outcomes will be held constant across bootstrap replications. In the case of a fuzzy analysis also calculate predicted participation probabilities.

4. Calculate rescaled residuals from the regression estimated in step 2 (both the outcome regression and the participation regression). The raw regression residuals are too small and need to be rescaled (Johnston and DiNardo 1997). To rescale the residuals, we divide each residual $\hat{e}_i$ by $\sqrt{1 - h_i}$, where $h_i = X_i(X'X)^{-1}X_i'$, and then re-center the rescaled residuals at zero. These residuals will be randomly sampled in each bootstrap replication.

5. Randomly sample the residuals from step 4. For each observation, calculate a new outcome variable equal to the sum of the randomly sampled residual plus the fixed predicted outcome from step 3. Do the same for participation.

6. Apply all RDD estimation methods using the outcome constructed in step 5 and the actual (fixed) values of all other variables. This includes all steps in RDD impact estimation -- the selection of the bandwidth (we use the IK bandwidth), accounting for multiple assignment variables, and estimating fuzzy impacts. Save the RDD impact.

7. Repeat steps 5 and 6 N times.

8. Calculate the standard error of the impact estimate as the standard deviation of the N impact estimates.

The primary consideration in developing this algorithm was the selection of the functional form used to estimate the relationship between the outcome and the assignment variable in the regression in step 2. In a typical residual bootstrap, the functional form of the regression used to...

---

3 We use the linear probability model rather than a logit or probit because this is the convention in fuzzy RDD impact analysis. See, for example, Imbens and Lemieux (2008).
generate residuals (step 2) would be the same as the functional form for the regression used to estimate impacts (step 6). In this case, however, that approach could be overly conservative since the residuals estimated in a global linear regression (i.e., a regression using all the observations) will be larger than the residuals from a local linear regression, unless the relationship between the assignment variable and the outcome is the same both locally and globally. Alternatively, choosing a polynomial that provides the best fit to the data globally has two drawbacks – (1) if we over-fit the data then the algorithm would generate standard errors that are too small and (2) employing a best-fit criterion to select the functional form for the regression in step 2 would take place outside of the bootstrap procedure which means that it would introduce a source of variability that is not reflected in the standard error, which might also lead to standard errors that are too small. As a compromise between the overly conservative approach of using a linear functional form in step 2 and the anti-conservative approach of using a best-fit polynomial, we chose to use a fixed quadratic functional form in step 2. We chose a quadratic functional form based on the guidance from Gelman and Imbens (2014) to avoid higher order polynomials in RDD estimation. Our simulations (see below) indicate that this is a reasonable compromise that allows this algorithm to control type 1 errors at the desired rate in most cases while offering more power than the nonparametric bootstrap.

III. Monte Carlo Experiment

We use Monte Carlo experiments to assess the two bootstrap algorithms described in the previous section. We begin with five experiments involving a “single site” study (that is, a single impact estimated using one cutoff on one assignment variable). The single site experiments consist of the following steps:

1. Use a data generating process (DGP) to create an artificial data set that includes an outcome, an assignment variable, and a treatment participation indicator such that there is a discontinuity in participation at the cutoff value on the assignment variable. The data generating processes we use to relate outcomes to assignment variables are shown in the first 5 rows of Table 1. The cutoff value in all cases is the median of the assignment variable. All data are generated under the null hypothesis of no impact on the outcome variable. The DGP used to relate participation (P) to the assignment variable is the same for all of the single site experiments and is shown in the last row of Table 1. Participation data is generated such that there is an impact of 0.75 on the participation rate at the cutoff value of the assignment variable.

2. Estimate RDD impacts and standard errors using the data generated in step 1. We use four different approaches to calculating standard errors: (1) conventional least squares standard errors, (2) re-estimating the local linear impact regression including the square of the assignment variable in order to estimate “bias-corrected” standard errors (Calonico, Cattaneo, and Titiunik 2014), (3) the non-parametric bootstrap described in section II, and (4) the residual bootstrap described in section II.

3. Repeat steps 1 and 2 10,000 times. For each replication, record the estimated impact, the estimated standard errors using the four methods described above, and the p-values based on the impact and standard error estimates.
Table 1. Single Site Data Generating Processes

<table>
<thead>
<tr>
<th>DGP</th>
<th>Assignment Variable Distribution</th>
<th>Error Distribution</th>
<th>Functional Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( x \sim U \left( -\frac{\sqrt{12}}{2}, \frac{\sqrt{12}}{2} \right) )</td>
<td>( u \sim N(0,1) )</td>
<td>Linear: ( y_{i,j} = \sqrt{0.5} x_j + \sqrt{0.5}(\sqrt{0.1} u_j + \sqrt{0.9} e_{i,j}) )</td>
</tr>
<tr>
<td>2</td>
<td>( x \sim U \left( -\frac{\sqrt{12}}{2}, \frac{\sqrt{12}}{2} \right) )</td>
<td>( u \sim N(0,1) )</td>
<td>Quadratic: ( y_{i,j} = \sqrt{0.5} \cdot \text{sign}(x_j) \cdot</td>
</tr>
<tr>
<td>3</td>
<td>( x \sim U \left( -\frac{\sqrt{12}}{2}, \frac{\sqrt{12}}{2} \right) )</td>
<td>( u \sim N(0,1) )</td>
<td>Cubic: ( y_{i,j} = \sqrt{0.5} \cdot x_j^3 + \sqrt{0.5}(\sqrt{0.1} u_j + \sqrt{0.9} e_{i,j}) )</td>
</tr>
<tr>
<td>4</td>
<td>( z \sim \text{Beta}(2,4) ) ( x = 2z - 1 )</td>
<td>( e \sim N(0,0.1295) )</td>
<td>IK1: ( y_{i,j}[x_j &lt; 0] = 0.50 + 1.27 x_j + 7.18 x_j^2 + 20.21 x_j^3 + 21.54 x_j^4 + 7.33 x_j^5 + e_{i,j} ) ( y_{i,j}[x_j \geq 0] = 0.50 + 0.84 x_j - 3 x_j^2 + 7.99 x_j^3 - 9.01 x_j^4 + 3.56 x_j^5 + e_{i,j} )</td>
</tr>
<tr>
<td>5</td>
<td>( z \sim \text{Beta}(2,4) ) ( x = 2z - 1 )</td>
<td>( e \sim N(0,0.1295) )</td>
<td>IK2: ( y_{i,j}[x_j &lt; 0] = 3 x_j^2 + e_{i,j} ) ( y_{i,j}[x_j \geq 0] = 4 x_j^2 + e_{i,j} )</td>
</tr>
<tr>
<td>P</td>
<td>Same as outcome</td>
<td>( u \sim N(0,1) ) ( e \sim N(0,1) )</td>
<td>( P^<em><em>{i,j} = \sqrt{0.5} \left( \frac{x_j - \text{median}(x)}{\sigma_x} \right) + \sqrt{0.5}(\sqrt{0.1} u_j + \sqrt{0.9} e</em>{i,j}) ) ( P_{i,j} = 1 ) if ( P^</em><em>{i,j} &lt; F^{-1}(0.85) ) and 0 otherwise ( P</em>{i,j} = 1 ) if ( P^*_{i,j} &lt; F^{-1}(0.10) ) and 0 otherwise</td>
</tr>
</tbody>
</table>

Source: For DGPs 1 through 3 the assignment variable was constructed to have a mean of zero and a standard deviation of 1. DGPs 4 and 5 are taken from Imbens and Kalyanaraman (2012). Unique values of the assignment variable are indexed by \( j \), assigned units are indexed by \( i \). The assignment variable is \( x_j \), \( y \) is the outcome, \( P^* \) is the continuous latent propensity to participate, \( P \) is a dichotomous participation variable, \( u \) is a random effect at the level of unique values of the assignment variable, \( e \) is a random effect at the level of assigned units, and \( F \) is the cumulative distribution function for the normal distribution with mean 0 and standard deviation \( \sqrt{0.5} \).

[not yet complete] In addition to the single site experiments, we also conduct a multi-site experiment. The multi-site experiment generates data in step 1 using \(<N>\) different DGPs. The DGPs are based on data from an RDD evaluation of School Improvement Grants (SIG), in which cutoffs on school-level achievement and (in the case of high schools) graduation rates are used as eligibility criteria for receipt of SIG funds. The data come from elementary, middle, and high schools in \(<X>\) states. Each DGP involves a distinct sample with a different relationship between the assignment variable(s) and the outcome. The relationship between the outcome (student-level math test scores) and assignment variable in each DGP is a quartic polynomial. The distributions of the assignment variables and the coefficients for each polynomial are based on empirical estimates using data from the SIG evaluation (the outcome variable used to form the empirical estimates comes from before the implementation of SIG, meaning that there is no impact of the intervention). \(<M>\) of the \(<N>\) DGPs involve two assignment variables, the rest involve a single assignment variable.

IV. Findings

Single Site Experiments

---

Findings from the single site experiments are shown in Table 2. We report the bias, variance ratio, and error rate for four different approaches to calculating standard errors: (1) conventional (meaning least squares standard errors), (2) bias-corrected (least squares standard errors from an impact regression that includes the square of the assignment variable), (3) nonparametric bootstrap, and (4) residual bootstrap. Bias is the mean impact estimate across Monte Carlo replications and is invariant to the method used to estimate standard errors. The variance ratio is the mean (averaging across Monte Carlo replications) estimated variance divided by the variance in impacts across Monte Carlo replications (ideally the variance ratio would be 1). The error rate is the proportion of Monte Carlo replications in which the estimated p-value is less than 0.05 (ideally the error rate would equal 0.05). We report all of these values for both reduced form impact estimation and fuzzy impact estimation.

Conventional standard errors are always too small, resulting in an error rate that is too high. The variance ratio ranges from 0.385 to 0.840 and the error rate ranges from 0.064 to 0.210. The variance ratio is lowest (and the error rate highest) for the first three DGPs because they include a random effect at the level of unique values of the assignment variable while the last two DGPs do not.

Bias-corrected standard errors are always too large, often (but not always) resulting in an error rate that is too low. The smallest variance ratio is 1.732, and in some cases the variance ratio is very large due to some individual Monte Carlo replications returning standard error estimates exploding towards infinity\(^5\). The error rate ranges from 0.000 to 0.063, with 0.063 being the only error rate above 0.05 for this method.

Standard errors calculated using the nonparametric bootstrap are always too large, resulting in an error rate that is always too small. The variance ratio ranges from 1.114 to 20.812 and the error rate ranges from 0.017 to 0.046.

Standard errors calculated using the residual bootstrap are closest to being correct, resulting in error rates that are also closest to being correct. The variance ratio ranges from 0.896 to 1.139 and the error rate ranges from 0.036 to 0.064.

*Multi-Site Experiments*

[not yet complete]

---

\(^5\) In calculating bias and the variance ratio we used a trimmed mean, in which the top and bottom 0.1% of values across Monte Carlo replications were omitted. Despite this trimming, bias-corrected standard errors still led to some very large variance estimates.
Table 2. Bias, Variance Ratios, and Error Rates by Estimation Approach

<table>
<thead>
<tr>
<th></th>
<th>Reduced Form Impact</th>
<th>Fuzzy Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Conventional Bias-Corrected</td>
<td>Nonparametric Bootstrap</td>
</tr>
<tr>
<td>Bias</td>
<td>-0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Variance Ratio</td>
<td>0.385</td>
<td>2.164</td>
</tr>
<tr>
<td>Error Rate</td>
<td>0.191</td>
<td>0.036</td>
</tr>
<tr>
<td>DGP: Linear</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>0.071</td>
<td></td>
</tr>
<tr>
<td>Variance Ratio</td>
<td>0.385</td>
<td>6.811</td>
</tr>
<tr>
<td>Error Rate</td>
<td>0.210</td>
<td>0.049</td>
</tr>
<tr>
<td>DGP: Quadratic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>0.028</td>
<td></td>
</tr>
<tr>
<td>Variance Ratio</td>
<td>0.437</td>
<td>803372</td>
</tr>
<tr>
<td>Error Rate</td>
<td>0.182</td>
<td>0.038</td>
</tr>
<tr>
<td>DGP: Cubic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>-0.004</td>
<td></td>
</tr>
<tr>
<td>Variance Ratio</td>
<td>0.835</td>
<td>1.764</td>
</tr>
<tr>
<td>Error Rate</td>
<td>0.071</td>
<td>0.011</td>
</tr>
<tr>
<td>DGP: IK2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>-0.001</td>
<td></td>
</tr>
<tr>
<td>Variance Ratio</td>
<td>0.840</td>
<td>1.732</td>
</tr>
<tr>
<td>Error Rate</td>
<td>0.072</td>
<td>0.012</td>
</tr>
</tbody>
</table>

Source: Monte Carlo simulations, 10,000 replications. 50 unique values of the assignment variable, 500 units assigned. The cutoff value is the median of the assignment variable. The bias and variance ratio are calculated using truncated means, such that the top and bottom 0.1% of replications are excluded. The error rate is calculated using all replications (no truncation).
V. Conclusion

Based on the simulations conducted so far, we conclude that the residual bootstrap algorithm described here performs much better than the alternatives considered, which included conventional standard errors, bias-corrected standard errors, and standard errors calculated using the nonparametric bootstrap. Specifically the residual bootstrap algorithm is best able to control the Type 1 error rate at the desired level – other methods are either too conservative (bias-corrected standard errors and nonparametric bootstrapping) or anti-conservative (conventional RDD standard errors). However we have more work to complete (which we intend to add to a future version of this paper) before reaching final conclusions. The additional work includes:

1. Running single-site simulations with larger sample sizes
2. Running multiple-site simulations
3. Calculating bootstrap (both nonparametric and residual) $p$-values that incorporate asymptotic refinement (as in Cameron, Gelbach, and Miller 2008)


