rdlasso: A Stata command for high-dimensional regression discontinuity designs

Marianna Nitti, Marco Ventura
Sapienza University of Rome

25 September 2025

The 19th Italian STATA Conference

Milan, Italy

Outline

Motivations

- Including many pre-determined covariates can improve the precision of treatment effect estimates in RDD (Calonico et al., 2014, 2019; Kreiss & Rothe, 2023).
- However, in high-dimensional settings (number of covariates p greater than number of observations n), **direct inclusion** of all covariates without selection can lead to:
  - Overfitting
  - Distorted inference
- Kreiss & Rothe (2023) propose a two-step approach:
  - Step 1: Local Lasso selection of covariates within the bandwidth
  - Step 2: Post-Lasso RD estimation using the selected covariates
- From a theoretical perspective, under an **approximate sparsity** condition (i.e., only a small subset of covariates is relevant near the cutoff), the estimator behaves like the standard local linear estimator in terms of bias and variance.

Our Contribution

- We develop `rdlasso`, a new Stata command for Regression Discontinuity Designs with high-dimensional covariates.
- **Main features:**
  - Implements the two-step procedure of Kreiss & Rothe (2023)
  - Support both sharp and fuzzy RD designs
  - Relies on Python Machine Learning libraries through the Stata Function Interface (sfi)
- **Advantages for users:**
  - No need to switch to R or Python
  - Results available as standard Stata variables and scalars
  - Easy incorporation in Stata do-files for further analysis

The econometric set up (1)

- **Sample:** (Y_i, X_i, W_i), with Y_i outcome, X_i running variable, W_i covariates.
- **Treatment:** $T_i = I(X_i \geq 0)$.
- **Observed outcome:** $Y_i = Y_i(T_i)$.
- **Parameter of interest (ATE at threshold):**

$$\tau_Y = \mathbb{E}[Y_i(1) - Y_i(0) \mid X_i = 0]$$

- **Identification via continuity of conditional expectations:**

$$\tau_Y = \lim_{x \downarrow 0} \mathbb{E}[Y \mid X = x] - \lim_{x \uparrow 0} \mathbb{E}[Y \mid X = x]$$

The econometric set up (2)

- Fit separate local linear regressions on each side of cutoff.
- Baseline estimator:

$$\hat{\tau}_{h,Base} = e_2^\top \arg\min_\theta \sum_{i=1}^{n} K_h(X_i) \left[Y_i - V_i^\top \theta \right]^2$$

- **Definitions:**
  - $K_h(x) = \frac{1}{h} K(x/h)$: kernel function with bandwidth h
  - $V_i = (1, T_i, X_i/h, T_i X_i/h)^\top$
  - $e_2 = (0, 1, 0, 0)^\top$ selects treatment effect coefficient

The econometric set up (3)

- Researchers often augment the baseline with covariates:

$$\hat{\tau}_{h,CCFT} = e_2^\top \arg\min_{(\theta,\gamma)} \sum_{i=1}^{n} K_h(X_i) \left[Y_i - V_i^\top \theta - W_i^\top \gamma \right]^2$$

- Consistent under standard RDD identification assumptions.
- **High-dimensional challenges:** When the number of covariates p is large relative to n (even $p > n$), the estimator may become undefined and there is a risk of overfitting.
- This motivates the use of regularization and variable selection (next section).

The econometric set up (4)

Step 1: Local Lasso Selection

- Weighted Lasso within preliminary bandwidth b and penalty λ:

$$(\hat{\theta}_n, \hat{\gamma}_n) = \arg\min_{\theta,\gamma} \sum_{i=1}^{n} K_b(X_i) \left[Y_i - V_i^\top \theta - (W_i - \hat{\mu}_{Z,n})^\top \gamma \right]^2 + \lambda \sum_{k=1}^{p_n} \hat{w}_{n,k} |\gamma_k|$$

- Covariates implicitly standardized; selected set $\hat{J}_n = \{k : \hat{\gamma}_k \neq 0\}$
- Local Lasso: selection focuses on variables predictive of outcome within the bandwidth

Step 2: Post-Lasso Estimation

- Final estimation on selected covariates with bandwidth h:

$$\hat{\tau}_h(\hat{J}_n) = e_2^\top \arg\min_{\theta,\gamma} \sum_{i=1}^{n} K_h(X_i) \left[Y_i - V_i^\top \theta - W_i(\hat{J}_n)^\top \gamma \right]^2$$

- Asymptotic distribution does not depend on b or λ; standard RDD inference applies.

RDD with high-dimensional covariates in practice

- **Fuzzy RD:** treatment not perfectly assigned → probability jump at cutoff.

$$\tau_{\text{fuzzy}} = \frac{\tau_Y^+ - \tau_Y^-}{\tau_T^+ - \tau_T^-}$$

- Estimation: run post-Lasso twice (once with Y_i, once with T_i) and take ratio.
- Asymptotic normality via delta method if τ_T bounded away from zero.

rdlasso

- b: preliminary bandwidth chosen by standard baseline RD methods
- λ: tuning parameter via cross-validation or plug-in
- h: final bandwidth as in covariate-adjusted RD
- Post-Lasso is flexible, robust and efficient.

The Stata syntax of `rdlasso` (1)

`rdlasso` *varlist* [*if*] [*in*], [,`t` (*varname*)
`z` (*varname*) `c` (*real*) `fuzzy` (*string*)`level` (*real*)
`b` (*string*)`bfactor` (*real*)
`h` (*string*)`tpc` (*string*) *pypackagesnocheck*]

Where:

varlist is a list of numerical variables, it must includes in order: the outcome variables (Y), the running variable(X), one or more covariates (D) to be used for high-dimensional adjustment.

The Stata syntax of `rdlasso` (2)

`t` (*varname*) specifies the treatment variable, `t ()` is required for fuzzy designs.

`z` (*varname*) specifies the instrumental variable, `z ()` is required for fuzzy designs.

`c` (*real*) indicates the cutoff value of the running variable.

`fuzzy` (*string*) activates fuzzy design logic (default is "off"), `fuzzy ("on")` is required for fuzzy designs.

`level` (*real*) indicates the confidence level for inference, between 0 and 1. (default value is 0.95).

`b` (*string*) indicates the bandwidth for the model selection. If it is not specified, the default, bandwidth selection from rdrobust without covariates is used to find b.

`bfactor` (*real*) indicates the scaling factor applied to the selected bandwidth for the model selection (default value is 1.0).

The Stata syntax of `rdlasso` (3)

`h` (*string*) indicates the bandwidth for the actual RD estimation. If it is not specified, the default, bandwidth selection from rdrobust with the selected covariate set is used

`tpc` (*string*) specifies the algorithm used to select the tuning parameter in the Lasso-based model selection step. The default is CV (cross-validation). Available methods are: CV (cross-validation with sample weights) and BCH (plug-in rule by Belloni et al., 2014)

pypackagesnocheck skips package dependency checks in Python.

kernel Specifies which kernel to be used for both model selection and estimation, possible choices are triangular (the default), epanechnikov, uniform)

Example 1: Sharp RD Design

Does Islamic political control affect women's empowerment?[1]

- **Source:** Cattaneo, Idrobo & Titiunik (2020), *A Practical Introduction to RDD*.
- **Setting:** Turkish municipal elections, 1994.
- **Running variable (X):** Margin of victory of the Islamic party over its strongest secular opponent.
- **Cutoff:** 0 (Islamic mayor if margin > 0; secular mayor otherwise).
- **Treatment (T):** Electoral victory of the Islamic party in 1994.
- **Outcome (Y):** Share of women aged 15–20 in 2000 who had completed high school.

[1] Meyersson (2014), *Econometrica*.

Example 1: Estimation and Results

```
. rdrobust Y X

Sharp RD estimates using local polynomial regression.

      Cutoff c = 0 | Left of c  Right of c            Number of obs =       1908
-------------------+----------------------            BW type       =      mserd
     Number of obs |      1683         225            Kernel        = Triangular
Eff. Number of obs |       263         147            VCE method    =         NN
    Order est. (p) |         1           1
    Order bias (q) |         2           2
       BW est. (h) |    11.514      11.514
       BW bias (b) |    19.031      19.031
         rho (h/b) |     0.605       0.605

Outcome: Y. Running variable: X.
--------------------------------------------------------------------------------
            Method |   Coef.    Std. Err.    z     P>|z|    [95% Conf. Interval]
-------------------+------------------------------------------------------------
      Conventional |  1.3602      2.015   0.6750   0.500   -2.58914      5.3095
            Robust |     -          -     0.3423   0.732   -3.80618     5.41699
--------------------------------------------------------------------------------
```

Figure: rdrobust

Example 1: Estimation and Results

```
. rdlasso Y X _3-_211

Selected covariates (Y): 83 175 179 182

Sharp RD estimates using local polynomial regression.
```

Cutoff c = 0	Left of c	Right of c
Number of obs	1683	225
Eff. Number of obs	260	145
Order est. (p)	1	1
Order bias (q)	2	2
BW est. (h)	11.410	11.410
BW bias (b)	11.514	11.514
rho (h/b)	0.991	0.991

Number of obs	=	1908
BW type	=	Manual
Kernel	=	Triangular
VCE method	=	NN

```
Outcome: Y. Running variable: X.
```

Method	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
Conventional	1.3145	2.0236	0.6496	0.516	-2.65175	5.28072
Robust	-	-	-0.1914	0.848	-6.27723	5.16007

Figure: rdlasso

Further developments and suggestions

- **Tuning parameter choice:** Currently only the plug-in method of Belloni et al. (2014) is implemented. $\rightarrow$ Next step: add cross-validation and other selection methods.
- **Alternative algorithms:** According to Kreiss & Rothe (2023), the theoretical properties of the estimator remain valid with other regularization methods. $\rightarrow$ Possible extensions: Elastic Net, Post-Selection OLS, Boosting.
- **Additional functionality:**
  - **Integration with `rdrobust`:** Current version already passes key options (cutoff, bandwidths, kernel, confidence level). $\rightarrow$ Future work: extend to additional
  - **Documentation and replication:** Development of a full documentation and extended replication materials.

Thank you for your attention!