Package ‘rdlearn’

January 29, 2025

Type Package

Title Safe Policy Learning under Regression Discontinuity Design with Multiple Cutoffs

Version 0.1.1

Description Implements safe policy learning under regression discontinuity designs with multiple cutoffs, based on Zhang et al. (2022) <doi:10.48550/arXiv.2208.13323>. The learned cutoffs are guaranteed to perform no worse than the existing cutoffs in terms of overall outcomes. The 'rdlearn' package also includes features for visualizing the learned cutoffs relative to the baseline and conducting sensitivity analyses.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

URL https://github.com/kkawato/rdlearn

BugReports https://github.com/kkawato/rdlearn/issues

Imports nprobust, nnet, rdrobust, ggplot2, dplyr, glue, cli

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 3.5.0)

VignetteBuilder knitr

NeedsCompilation no

Author Kentaro Kawato [cre, cph],
Yi Zhang [aut],
Soichiro Yamauchi [aut],
Eli Ben-Michael [aut],
Kosuke Imai [aut]

Maintainer Kentaro Kawato <kentaro1358nohe@gmail.com>

Repository CRAN

Date/Publication 2025-01-29 18:30:02 UTC

Contents

`acces` *ACCES Program*

Description

A dataset comprising 8245 applicants to the ACCES Program across 23 different departments in Colombia, including eligibility for the ACCES Program, position score of the SABER 11, cutoff of each department, and the name of each department.

Usage

```
acces
```

Format

A data frame with 8245 rows and 4 columns:

elig eligibility for the ACCES Program. 1: eligible; 0: not eligible

saber11 position scores of the SABER 11. We multiply the position score by -1 so that the values of the running variable above a cutoff lead to the program eligibility.

cutoff cutoffs of each department.

department the names of each department.

References

Melguizo, T., F. Sanchez, and T. Velasco (2016). Credit for low-income students and access to and academic performance in higher education in colombia: A regression discontinuity approach. World development 80, 61-77.

plot *Plot Cutoff Changes for rdlearn Objects*

Description

This function plots the changes in cutoff values relative to the baseline cutoffs for each group, under different combinations of the smoothness multiplier (M) and the cost of treatment (C).

Usage

```
plot(x, opt, ...)
```

Arguments

`x`	An object of class `rdlearn` returned by the `rdlearn` function.
`opt`	When set to "safe", it displays the derived safe cutoffs and the original cutoffs. When set to "dif", it displays the change in cutoffs.
`...`	additional arguments.

Value

A `ggplot2` plot which also contains the distance measure between original cutoffs and safe cutoffs.

Examples

```
# Simulation Data B from Appendix D of Zhang et al. (2022)
set.seed(1)
n <- 300
X <- runif(n, -1000, -1)
G <- 2 * as.numeric(
I(0.01 * X + 5 + rnorm(n, sd = 10) > 0)
) +
as.numeric(
I(0.01 * X + 5 + rnorm(n, sd = 10) <= 0)
)
c1 <- -850
c0 <- -571
C <- ifelse(G == 1, c1, c0)
D <- as.numeric(X >= C)
coef0 <- c(-1.992230e+00, -1.004582e-02, -1.203897e-05, -4.587072e-09)
coef1 <- c(9.584361e-01, 5.308251e-04, 1.103375e-06, 1.146033e-09)
Px <- poly(X, degree = 3, raw = TRUE)
# Px = poly(X-735.4334-c1,degree=3,raw=TRUE) for Simulation A
Px <- cbind(rep(1, nrow(Px)), Px)
EY0 <- Px %*% coef0
EY1 <- Px %*% coef1
d <- 0.2 + exp(0.01 * X) * (1 - G) + 0.3 * (1 - D)
Y <- EY0 * (1 - D) + EY1 * D - d * as.numeric(I(G == 1)) + rnorm(n, sd = 0.3)
```

```
simdata_B_demo <- data.frame(Y,X,C)

# Learn new treatment assignment cutoffs
rdlearn_result <- rdlearn(
  y = "Y", x = "X", c = "C", data = simdata_B_demo,
  fold = 2, M = 0, cost = 0
)

# Summarise the learned policies
summary(rdlearn_result)

# Visualize the learned policies
plot(rdlearn_result, opt = "dif")
# The learned cutoff for Group 1 is the same as the baseline cutoff, because
# the baseline cutoff is set to equal to oracle cutoff in this simulation.

# Implement sensitivity analysis
sens_result <- sens(rdlearn_result, M = 1, cost = 0)
plot(sens_result, opt = "dif")
```

`rdestimate` *RD Estimate Function*

Description

This function estimates local causal effect of treatment under standard regression discontinuity (RD) setting.

Usage

```
rdestimate(y, x, c, group_name = NULL, data)
```

Arguments

`y`	A character string specifying the name of column containing the outcome variable.
`x`	A character string specifying the name of column containing the running variable.
`c`	A character string specifying the name of column containing the cutoff variable.
`group_name`	A character ctring specifying the name of the column containing group names (e.g., department names) for each cutoff. If not provided, the groups are assigned names "Group 1", "Group 2", ... in ascending order of cutoff values.
`data`	A data frame containing all required variables.

Value

A data frame with the RD estimates for each group, including the sample size of each group, baseline cutoff, RD estimate, standard error, and p-value.

Examples

```
rdestimate_result <- rdestimate(
  y = "elig", x = "saber11", c = "cutoff",
  group_name = "department", data = acces
)
print(rdestimate_result)
```

---

rdlearn *Safe Policy Learning for Regression Discontinuity Design with Multiple Cutoffs*

Description

The rdlearn function implements safe policy learning under a regression discontinuity design with multiple cutoffs. The resulting new treatment assignment rules (cutoffs) are guaranteed to yield no worse overall outcomes than the existing cutoffs.

Usage

```
rdlearn(
  y,
  x,
  c,
  group_name = NULL,
  data,
  fold = 10,
  M = 1,
  cost = 0,
  trace = TRUE
)
```

Arguments

y	A character string specifying the name of column containing the outcome variable.
x	A character string specifying the name of column containing the running variable.
c	A character string specifying the name of column containing the cutoff variable.
group_name	A character string specifying the name of the column containing group names (e.g., department names) for each cutoff. If not provided, the groups are assigned names "Group 1", "Group 2", ... in ascending order of cutoff values.
data	A data frame containing all required variables.
fold	The number of folds for cross-fitting. Default is 10.
M	A numeric value or vector specifying the multiplicative smoothness factor(s) for sensitivity analysis. Default is 1.

`cost` A numeric value or vector specifying the cost of treatment for calculating regret. This cost should be scaled by the range of the outcome variable Y. Default is 0.

`trace` A logical value that controls whether to display the progress. If set to TRUE, the progress will be printed. The default value is TRUE.

Details

Regarding the detail of the algorithm, please refer to Zhang et al. (2022) "4 Empirical policy learning" and "A.2 A double robust estimator for heterogeneous cross-group differences".

Value

An object of class `rdlearn`, which is a list containing the following components:

call The original function call.

var_names A list of variable names for the outcome, running variable, and cutoff.

org_cut A vector of original cutoff values.

safe_cut A data frame containing the obtained new treatment assignment cutoffs.

sample The total sample size.

num_group The number of groups.

group_name A vector of group names.

cross_fit_output The intermediate output of the cross-fitting procedure.

dif_lip_output The intermediate output of the cross-group differences and the smoothness parameters

distance A numeric vector containing the measures of difference between safe cutoffs and original cutoffs

rdestimates A data frame containing the result of `rdesimate` such as causal effect estimates.

temp_reg_df A data frame containing the regrets of every alternative cutoff.

Examples

```
# Simulation Data B from Appendix D of Zhang et al. (2022)
set.seed(1)
n <- 300
X <- runif(n, -1000, -1)
G <- 2 * as.numeric(
I(0.01 * X + 5 + rnorm(n, sd = 10) > 0)
) +
as.numeric(
I(0.01 * X + 5 + rnorm(n, sd = 10) <= 0)
)
c1 <- -850
c0 <- -571
C <- ifelse(G == 1, c1, c0)
D <- as.numeric(X >= C)
coef0 <- c(-1.992230e+00, -1.004582e-02, -1.203897e-05, -4.587072e-09)
coef1 <- c(9.584361e-01, 5.308251e-04, 1.103375e-06, 1.146033e-09)
```

```
Px <- poly(X, degree = 3, raw = TRUE)
# Px = poly(X-735.4334-c1,degree=3,raw=TRUE) for Simulation A
Px <- cbind(rep(1, nrow(Px)), Px)
EY0 <- Px %*% coef0
EY1 <- Px %*% coef1
d <- 0.2 + exp(0.01 * X) * (1 - G) + 0.3 * (1 - D)
Y <- EY0 * (1 - D) + EY1 * D - d * as.numeric(I(G == 1)) + rnorm(n, sd = 0.3)

simdata_B_demo <- data.frame(Y,X,C)

# Learn new treatment assignment cutoffs
rdlearn_result <- rdlearn(
  y = "Y", x = "X", c = "C", data = simdata_B_demo,
  fold = 2, M = 0, cost = 0
)

# Summarise the learned policies
summary(rdlearn_result)

# Visualize the learned policies
plot(rdlearn_result, opt = "dif")
# The learned cutoff for Group 1 is the same as the baseline cutoff, because
# the baseline cutoff is set to equal to oracle cutoff in this simulation.

# Implement sensitivity analysis
sens_result <- sens(rdlearn_result, M = 1, cost = 0)
plot(sens_result, opt = "dif")
```

---

sens *Sensitivity Analysis for rdlearn Objects*

Description

This function performs sensitivity analysis for the `rdlearn` object under different smoothness multiplier (M) and the cost of treatment (cost).

Usage

```
sens(object, M = NULL, cost = NULL, trace = TRUE)
```

Arguments

`object`	An object of class `rdlearn` returned by the `rdlearn` function.
`M`	A numeric value or vector specifying the multiplicative smoothness factor(s) for sensitivity analysis.
`cost`	A numeric value or vector specifying the cost of treatment for calculating regret.
`trace`	A logical value that controls whether to display the progress of cross-fitting and regret calculation. If set to TRUE, the progress will be printed. The default value is TRUE.

Value

An updated `rdlearn` object with the new cutoffs based on the provided values of M and cost.

Examples

```
# Simulation Data B from Appendix D of Zhang et al. (2022)
set.seed(1)
n <- 300
X <- runif(n, -1000, -1)
G <- 2 * as.numeric(
I(0.01 * X + 5 + rnorm(n, sd = 10) > 0)
) +
as.numeric(
I(0.01 * X + 5 + rnorm(n, sd = 10) <= 0)
)
c1 <- -850
c0 <- -571
C <- ifelse(G == 1, c1, c0)
D <- as.numeric(X >= C)
coef0 <- c(-1.992230e+00, -1.004582e-02, -1.203897e-05, -4.587072e-09)
coef1 <- c(9.584361e-01, 5.308251e-04, 1.103375e-06, 1.146033e-09)
Px <- poly(X, degree = 3, raw = TRUE)
# Px = poly(X-735.4334-c1,degree=3,raw=TRUE) for Simulation A
Px <- cbind(rep(1, nrow(Px)), Px)
EY0 <- Px %*% coef0
EY1 <- Px %*% coef1
d <- 0.2 + exp(0.01 * X) * (1 - G) + 0.3 * (1 - D)
Y <- EY0 * (1 - D) + EY1 * D - d * as.numeric(I(G == 1)) + rnorm(n, sd = 0.3)

simdata_B_demo <- data.frame(Y,X,C)

# Learn new treatment assignment cutoffs
rdlearn_result <- rdlearn(
  y = "Y", x = "X", c = "C", data = simdata_B_demo,
  fold = 2, M = 0, cost = 0
)

# Summarise the learned policies
summary(rdlearn_result)

# Visualize the learned policies
plot(rdlearn_result, opt = "dif")
# The learned cutoff for Group 1 is the same as the baseline cutoff, because
# the baseline cutoff is set to equal to oracle cutoff in this simulation.

# Implement sensitivity analysis
sens_result <- sens(rdlearn_result, M = 1, cost = 0)
plot(sens_result, opt = "dif")
```

simdata_A *Simulation Data A*

Description

This dataset is based on the ACCES Program and generated according to scenario A described in Appendix D of Zhang et al. (2022). In this scenario, the baseline policy (cutoff) is set equal to the oracle policy.

Usage

```
simdata_A
```

Format

A data frame with 2000 rows and 3 columns:

Y outcome variable

X running variable

C cutoff value

References

Zhang, Y., Ben-Michael, E. and Imai, K. (2022) 'Safe Policy Learning under Regression Discontinuity Designs with Multiple Cutoffs', arXiv [stat.ME]. Available at: http://arxiv.org/abs/2208.13323.

simdata_B *Simulation Data B*

Description

This dataset is based on the ACCES Program and generated according to scenario B described in Appendix D of Zhang et al. (2022). In this scenario, the baseline policy (cutoff) differs from the oracle policy.

Usage

```
simdata_B
```

Format

A data frame with 2000 rows and 3 columns:

Y outcome variable

X running variable

C cutoff value

References

Zhang, Y., Ben-Michael, E. and Imai, K. (2022) 'Safe Policy Learning under Regression Discontinuity Designs with Multiple Cutoffs', arXiv [stat.ME]. Available at: http://arxiv.org/abs/2208.13323.

`summary` *Summary function*

Description

This function summarizes the key results returned by `rdlearn`.

Usage

```
summary(object, ...)
```

Arguments

`object`	An object of class `rdlearn` returned by the `rdlearn` function.
`...`	additional arguments.

Value

Displays key outputs from the `rdlearn` function. It provides basic information and RD causal effect estimates from `rdestimate`, as well as the safe cutoffs derived by `rdlearn` and the difference between them and the original cutoffs.

Examples

```
# Simulation Data B from Appendix D of Zhang et al. (2022)
set.seed(1)
n <- 300
X <- runif(n, -1000, -1)
G <- 2 * as.numeric(
I(0.01 * X + 5 + rnorm(n, sd = 10) > 0)
) +
as.numeric(
I(0.01 * X + 5 + rnorm(n, sd = 10) <= 0)
)
c1 <- -850
c0 <- -571
C <- ifelse(G == 1, c1, c0)
D <- as.numeric(X >= C)
coef0 <- c(-1.992230e+00, -1.004582e-02, -1.203897e-05, -4.587072e-09)
coef1 <- c(9.584361e-01, 5.308251e-04, 1.103375e-06, 1.146033e-09)
Px <- poly(X, degree = 3, raw = TRUE)
# Px = poly(X-735.4334-c1,degree=3,raw=TRUE) for Simulation A
Px <- cbind(rep(1, nrow(Px)), Px)
```

```
EY0 <- Px %*% coef0
EY1 <- Px %*% coef1
d <- 0.2 + exp(0.01 * X) * (1 - G) + 0.3 * (1 - D)
Y <- EY0 * (1 - D) + EY1 * D - d * as.numeric(I(G == 1)) + rnorm(n, sd = 0.3)

simdata_B_demo <- data.frame(Y,X,C)

# Learn new treatment assignment cutoffs
rdlearn_result <- rdlearn(
  y = "Y", x = "X", c = "C", data = simdata_B_demo,
  fold = 2, M = 0, cost = 0
)

# Summarise the learned policies
summary(rdlearn_result)

# Visualize the learned policies
plot(rdlearn_result, opt = "dif")
# The learned cutoff for Group 1 is the same as the baseline cutoff, because
# the baseline cutoff is set to equal to oracle cutoff in this simulation.

# Implement sensitivity analysis
sens_result <- sens(rdlearn_result, M = 1, cost = 0)
plot(sens_result, opt = "dif")
```

Index