

Package ‘causal.decomp’

August 27, 2025

Title Causal Decomposition Analysis

Version 0.2.0

Depends R (>= 3.5)

Imports stats, parallel, MASS, nnet, SuppDists, PSweight, utils,
rlang, DynTxRegime, distr, rpart, dplyr, modelObj, magrittr,
knitr

Suggests rmarkdown, rpart.plot, spelling, CBPS, pbmcapply

VignetteBuilder knitr

Description We implement causal decomposition analysis using methods proposed by Park, Lee, and Qin (2022) and Park, Kang, and Lee (2023), which provide researchers with multiple-mediator imputation, single-mediator imputation, and product-of-coefficients regression approaches to estimate the initial disparity, disparity reduction, and disparity remaining ([doi:10.1177/00491241211067516](https://doi.org/10.1177/00491241211067516); [doi:10.1177/00811750231183711](https://doi.org/10.1177/00811750231183711)). We also implement sensitivity analysis for causal decomposition using R-squared values as sensitivity parameters (Park, Kang, Lee, and Ma, 2023 [doi:10.1515/jci-2022-0031](https://doi.org/10.1515/jci-2022-0031)). Finally, we include individualized causal decomposition and sensitivity analyses proposed by Park, Kang, and Lee (2025+) [doi:10.48550/arXiv.2506.19010](https://doi.org/10.48550/arXiv.2506.19010).

License GPL-2

Encoding UTF-8

Language en-US

RoxygenNote 7.3.2

LazyData true

NeedsCompilation no

Author Suyeon Kang [aut, cre],
Soojin Park [aut],
Karen Xu [ctb]

Maintainer Suyeon Kang <suyeon.kang@ucf.edu>

Repository CRAN

Date/Publication 2025-08-27 16:20:22 UTC

Contents

causal.decomp-package	2
idata	2
ind.decomp	3
ind.sens	5
mmi	7
plot	11
pocr	12
sdata	14
sensitivity	15
smi	18
sMIDUS	23
Index	24

causal.decomp-package	<i>causal.decomp: Causal Decomposition Analysis.</i>
-----------------------	--

Description

The causal.decomp package provides six important functions: mmi, smi, pocr, sensitivity, ind.decomp, and ind.sens.

Author(s)

Maintainer: Suyeon Kang <suyeon.kang@ucf.edu>

Authors:

- Soojin Park <soojinp@ucr.edu>

Other contributors:

- Karen Xu <karenxu@ucr.edu> [contributor]

idata	<i>Synthetic Data for illustrating optimal treatment regimes and individualized effects</i>
-------	---

Description

A randomly generated dataset containing 2000 cases 7 columns with no missing values. The intermediate confounders are assumed to be independent of each other.

Usage

idata

Format

A data frame containing the following variables. The data are provided only for explanatory purposes.

Y: A continuous outcome variable.

R: A binary group indicator with a value of 0 (reference) and 1 (comparison).

M: A binary risk factor with a value of 0 (not treated/received) and 1(treated/received).

X1: First continuous intermediate confounder.

X2: Second continuous intermediate confounder.

X3: Third continuous intermediate confounder.

C: A continuous baseline covariate.

Details

Note that all the variables are randomly generated using the simulation setting in Park, S., Kang, S., & Lee, C. (2025).

References

Park, S., Kang, S., & Lee, C. (2025). Simulation-Based Sensitivity Analysis in Optimal Treatment Regimes and Causal Decomposition with Individualized Interventions. arXiv preprint arXiv:2506.19010.

ind.decomp

Causal Decomposition with Individualized Interventions

Description

‘ind.decomp’ estimates how much initial disparities would persist under two scenarios: (i) Everyone follows the optimal treatment regime (OTR), yielding the Individualized Controlled Direct Effect (ICDE); (ii) The proportion of individuals following the OTR is equalized across groups, yielding the Individualized Interventional Effect (IIE). This function implements the method proposed by Park, Kang, and Lee (2025+).

Usage

```
ind.decomp(outcome, group, group.ref, risk.factor, intermediates, moderators,
  covariates, data, B, cluster = NULL)
```

Arguments

outcome	a character string indicating the name of the outcome.
group	a character string indicating the name of the social group indicator such as race or gender.
group.ref	reference group of the social group indicator.
risk.factor	a character string indicating the name of the risk factor.

<code>intermediates</code>	vector containing the name of intermediate confounders.
<code>moderators</code>	a character string indicating the name of variables that have heterogeneous effects on the outcome based on the risk factor.
<code>covariates</code>	a vector containing the name of baseline covariate variable(s).
<code>data</code>	The data set for analysis (data.frame).
<code>B</code>	Number of bootstrapped samples in the causal decomposition analysis.
<code>cluster</code>	a vector of cluster indicators for the bootstrap. If provided, the cluster bootstrap is used. Default is 'NULL'.

Details

This function first estimates the initial disparity, defined as the average difference in an outcome between groups within the same levels of outcome-allowable covariates. Formally,

$$\tau_c \equiv E[Y \mid R = 1, c] - E[Y \mid R = 0, c], \quad \text{for } c \in \mathcal{C}$$

where $R = 1$ is the comparison group and $R = 0$ is the reference group. The term $c \in \mathcal{C}$ represents baseline covariates.

For individualized conditional effects, disparity remaining is defined as

$$\zeta_c^{\text{ICDE}}(d^{opt}) \equiv E[Y(d^{opt}) \mid R = 1, c] - E[Y(d^{opt}) \mid R = 0, c], \quad \text{for } c \in \mathcal{C}$$

where d^{opt} is an optimal value for risk factor M . This definition of disparity remaining states the difference in an outcome between the comparison and reference groups after setting their risk factor M to the optimal value obtained from the OTRs.

For individualized interventional effects, disparity reduction and disparity remaining due to equalizing compliance rates across groups can be expressed as follows:

$$\begin{aligned} \delta_c^{\text{IE}}(1) &\equiv E[Y \mid R = 1, c] - E[Y(K) \mid R = 1, c] \\ \zeta_c^{\text{IE}}(0) &\equiv E[Y(K) \mid R = 1, c] - E[Y \mid R = 0, c] \end{aligned}$$

where $Y(K)$ represents a potential outcome under the value of M that is determined by a random draw from the compliance distribution of the reference group among individuals with the same levels of target-factor-allowable covariates. Disparity reduction ($\delta_c^{\text{IE}}(1)$) represents the disparity in outcomes among a comparison group after intervening to setting the compliance rate equal to that of a reference group within the same target-factor-allowable covariate levels. Disparity remaining ($\zeta_c^{\text{IE}}(0)$) quantifies the outcome difference that persists between a comparison and reference group even after the intervention.

Value

a matrix containing the point estimates for:

1. the initial disparity and the proportion recommended for treatment,
2. the disparity remaining and percent reduction based on individualized conditional effects,
3. the disparity remaining, disparity reduction, and percent reduction based on individualized intervention effects.

It also returns the nonparametric bootstrap confidence intervals for each estimate.

Author(s)

Soojin Park, University of California, Riverside, <soojinp@ucr.edu>; Suyeon Kang, University of Central Florida, <suyeon.kang@ucf.edu>; Karen Xu, University of California, Riverside, <karenxu@ucr.edu>.

References

Park, S., Kang, S., & Lee, C. (2025). Simulation-Based Sensitivity Analysis in Optimal Treatment Regimes and Causal Decomposition with Individualized Interventions. arXiv preprint arXiv:2506.19010.

See Also

[ind.sens](#)

Examples

```
data(idata)

# NOTE: We recommend using at least B = 500 bootstrap replications.
results_unadj <- ind.decomp(outcome = "Y", group = "R", group.ref = "0", risk.factor = "M",
                           intermediates = c("X1", "X2", "X3"), moderators = c("X1", "X2"),
                           covariates = "C", data = idata, B = 50)

results_unadj
```

ind.sens	<i>Sensitivity Analysis for Causal Decomposition with Individualized Interventions</i>
----------	--

Description

‘ind.sens’ is used to assess the sensitivity of estimates from the individualized causal decomposition to potential omitted confounders between the risk factor M and the outcome Y, using a benchmarking approach based on the relative strength of an unmeasured confounder compared to a specified covariate. The function employs a stochastic EM algorithm to simulate the distribution of the unobserved confounder and generate bias-adjusted estimates.

Usage

```
ind.sens(k.y, k.m, Iternum, outcome, group, group.ref, intermediates, moderators,
        benchmark, risk.factor, covariates, data, B, cluster = NULL, nsim, mc.cores)
```

Arguments

k.y scales the association of the unmeasured confounder with the outcome (numeric). A value of 1 means a one-unit change in the unmeasured confounder shifts the mean of the outcome by the same amount as a one-unit change in the benchmark covariate—conditional on the social-group indicator, intermediate confounder(s), and baseline covariate(s). Default is 1.

<code>k.m</code>	scales the association of the unmeasured confounder with the log-odds of risk.factor = 1 (numeric). A value of 1 means a one-unit change in the unmeasured confounder changes the log-odds of the risk factor by the same amount as the benchmark covariate, conditional on the social-group indicator, intermediate confounder(s), and baseline covariate(s). Default is 1.
<code>Iternum</code>	Number of iterations in the stochastic EM algorithm for generating the unmeasured confounder. default is 10.
<code>outcome</code>	a character string indicating the name of the outcome.
<code>group</code>	a character string indicating the name of the social group indicator such as race or gender.
<code>group.ref</code>	reference group of the social group indicator.
<code>intermediates</code>	vector containing the name of intermediate confounders.
<code>moderators</code>	a character string indicating the name of variables that have heterogeneous effects on the outcome based on the risk factor.
<code>benchmark</code>	a vector containing the name of the benchmark covariate used to scale k.y and k.m.
<code>risk.factor</code>	a character string indicating the name of the risk factor.
<code>covariates</code>	a vector containing the name of baseline covariate variable(s).
<code>data</code>	The data set for analysis (data.frame).
<code>B</code>	Number of bootstrapped samples in the causal decomposition analysis.
<code>cluster</code>	a vector of cluster indicators for the bootstrap. If provided, the cluster bootstrap is used. Default is 'NULL'.
<code>nsim</code>	Number of random draws of the unmeasured confounder per observation. default is 15. Increase the number for more smooth curves.
<code>mc.cores</code>	Number of cores to use. Must be exactly 1 on Windows. Default is 1.

Details

This function returns the adjusted point estimates and confidence intervals after accounting for a simulated unmeasured confounder U . We employ an approach that compares the coefficient of the unmeasured confounder U with that of an observed covariate X_j , after controlling for the remaining observed covariates (R , X_{-j} , and C).

Specifically, k_y indicates the extent to which the outcome Y is associated with a one unit increase in U relative to how much it is associated with a one unit change in X_j , after controlling for R , X_{-j} , and C . Likewise, k_m indicates the extent to which the odds of $M = 1$ are explained by unmeasured confounder U relative to how much they are explained by X_j , after controlling for R , X_{-j} , and C . These parameters (k_m and k_y) should be specified by researchers based on the assumed strength of U relative to that of the given observed covariate X_j .

Value

A matrix containing the adjusted point estimates for:

1. the initial disparity, the proportion recommended for treatment,

2. the disparity remaining and percent reduction based on individualized conditional effects,
3. the disparity remaining, disparity reduction, and percent reduction based on individualized intervention effects.

It also returns the adjusted nonparametric bootstrap confidence intervals for each estimate.

Author(s)

Soojin Park, University of California, Riverside, <soojinp@ucr.edu>; Suyeon Kang, University of Central Florida, <suyeon.kang@ucf.edu>; Karen Xu, University of California, Riverside, <karenxu@ucr.edu>.

References

Park, S., Kang, S., & Lee, C. (2025). Simulation-Based Sensitivity Analysis in Optimal Treatment Regimes and Causal Decomposition with Individualized Interventions. arXiv preprint arXiv:2506.19010.

See Also

[ind.decomp](#)

Examples

```
data(idata)

results_adj <- ind.sens(k.y = 1, k.m = 1, Iternum = 5, outcome = "Y", group = "R", group.ref = "0",
  intermediates = c("X1", "X2", "X3"), moderators = c("X1", "X2"),
  benchmark = "X1", risk.factor = "M", covariates = "C", data = idata,
  B = 50, cluster = NULL, nsim = 10, mc.cores = 1)

results_adj
```

mmi

Multiple-Mediator-Imputation Estimation Method

Description

'mmi' is used to estimate the initial disparity, disparity reduction, and disparity remaining for causal decomposition analysis, using the multiple-mediator-imputation estimation method proposed by Park et al. (2020). This estimator was originally developed to handle multiple mediators simultaneously; however, it can also be applied to a single mediator.

Usage

```
mmi(fit.r = NULL, fit.x, fit.y, group, covariates, sims = 100, conf.level = .95,
  conditional = TRUE, cluster = NULL, long = TRUE, mc.cores = 1L, seed = NULL)
```

Arguments

<code>fit.r</code>	a fitted model object for social group indicator (treatment). Can be of class 'CBPS' or 'SumStat'. Default is 'NULL'. Only necessary if 'conditional' is 'FALSE'.
<code>fit.x</code>	a fitted model object for intermediate confounder(s). Each intermediate model can be of class 'lm', 'glm', 'multinom', or 'polr'. When multiple confounders are considered, can be of class 'list' containing multiple models.
<code>fit.y</code>	a fitted model object for outcome. Can be of class 'lm' or 'glm'.
<code>group</code>	a character string indicating the name of the social group indicator such as race or gender (treatment) used in the models. The social group indicator can be categorical with two or more categories (two- or multi-valued factor).
<code>covariates</code>	a vector containing the name of the covariate variable(s) used in the models. Each covariate can be categorical with two or more categories (two- or multi-valued factor) or continuous (numeric).
<code>sims</code>	number of Monte Carlo draws for nonparametric bootstrap.
<code>conf.level</code>	level of the returned two-sided confidence intervals, which are estimated by the nonparametric percentile bootstrap method. Default is .95, which returns the 2.5 and 97.5 percentiles of the simulated quantities.
<code>conditional</code>	a logical value. If 'TRUE', the function will return the estimates conditional on those covariate values, and all covariates in mediator and outcome models need to be centered prior to fitting. Default is 'TRUE'. If 'FALSE', 'fit.r' needs to be specified.
<code>cluster</code>	a vector of cluster indicators for the bootstrap. If provided, the cluster bootstrap is used. Default is 'NULL'.
<code>long</code>	a logical value. If 'TRUE', the output will contain the entire sets of estimates for all bootstrap samples. Default is 'TRUE'.
<code>mc.cores</code>	The number of cores to use. Must be exactly 1 on Windows.
<code>seed</code>	seed number for the reproducibility of results. Default is 'NULL'.

Details

This function returns the point estimates of the initial disparity, disparity reduction, and disparity remaining for a categorical social group indicator and a variety of types of outcome and mediator(s) in causal decomposition analysis. It also returns nonparametric percentile bootstrap confidence intervals for each estimate.

The initial disparity between two groups is defined as $\tau(r, 0) \equiv E[Y|R = r, c] - E[Y|R = 0, c]$, for $c \in \mathcal{C}$ and $r \in \mathcal{R}$, where $R = 0$ denotes the reference group and $R = r$ is the comparison group.

The disparity reduction, conditional on baseline covariates, measures how much the initial disparity would shrink if we simultaneously equalized the distribution of the mediators W and M across groups within each level of C . Formally, $\delta(1) \equiv E[Y|R = 1, c] - E[Y(G_{w|c}(0)G_{m|c}(0))|R = 1, c]$, where $G_{m|c}(0)$ and $G_{w|c}(0)$ are a random draw from the reference group's mediator M and W distribution given C , respectively.

The remaining disparity, also conditional on C , is the difference between the reference group's observed outcome and the counterfactual outcome for the comparison group after equalizing M . Formally, $\zeta(0) \equiv E[Y(Y(G_{w|c}(0)G_{m|c}(0)))|R = 1, c] - E[Y|R = 0, c]$.

The disparity reduction and remaining can be estimated using the multiple-mediator-imputation method suggested by Park et al. (2020). See the references for more details.

If one wants to make the inference conditional on baseline covariates, set `'conditional = TRUE'` and center the data before fitting the models.

As of version 0.1.0, the intermediate confounder model (`'fit.x'`) can be of class `'lm'`, `'glm'`, `'multinom'`, or `'polr'`, corresponding respectively to the linear regression models and generalized linear models, multinomial log-linear models, and ordered response models. The outcome model (`'fit.y'`) can be of class `'lm'` or `'glm'`. Also, the social group model (`'fit.r'`) can be of class `'CBPS'` or `'SumStat'`, both of which use the propensity score weighting. It is only necessary when `'conditional = FALSE'`.

Value

<code>result</code>	a matrix containing the point estimates of the initial disparity, disparity remaining, and disparity reduction, and the percentile bootstrap confidence intervals for each estimate.
<code>all.result</code>	a matrix containing the point estimates of the initial disparity, disparity remaining, and disparity reduction for all bootstrap samples. Returned if <code>'long'</code> is <code>'TRUE'</code> .

Author(s)

Suyeon Kang, University of Central Florida, <suyeon.kang@ucf.edu>; Soojin Park, University of California, Riverside, <soojinp@ucr.edu>.

References

- Park, S., Qin, X., & Lee, C. (2022). "Estimation and sensitivity analysis for causal decomposition in health disparity research". *Sociological Methods & Research*, 53(2), 571-602.
- Park, S., Kang, S., & Lee, C. (2023). "Choosing an Optimal Method for Causal Decomposition Analysis with Continuous Outcomes: A Review and Simulation Study". *Sociological Methodology*, 54(1), 92-117.

See Also

[smi](#)

Examples

```
data(sdata)

#-----#
# Example 1-a: Continuous Outcome
#-----#
fit.m1 <- lm(M.num ~ R + C.num + C.bin, data = sdata)
fit.m2 <- glm(M.bin ~ R + C.num + C.bin, data = sdata,
              family = binomial(link = "logit"))
require(MASS)
fit.m3 <- polr(M.cat ~ R + C.num + C.bin, data = sdata)
```

```

fit.x1 <- lm(X ~ R + C.num + C.bin, data = sdata)
require(nnet)
fit.m4 <- multinom(M.cat ~ R + C.num + C.bin, data = sdata)
fit.y1 <- lm(Y.num ~ R + M.num + M.bin + M.cat + X + C.num + C.bin,
            data = sdata)

require(PStweight)
fit.r1 <- SumStat(R ~ C.num + C.bin, data = sdata, weight = "IPW")
require(CBPS)
fit.r2 <- CBPS(R ~ C.num + C.bin, data = sdata, method = "exact",
              standardize = "TRUE")

res.1a <- mmi(fit.r = fit.r1, fit.x = fit.x1,
              fit.y = fit.y1, sims = 40, conditional = FALSE,
              covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.1a

#-----#
# Example 1-b: Binary Outcome
#-----#
fit.y2 <- glm(Y.bin ~ R + M.num + M.bin + M.cat + X + C.num + C.bin,
             data = sdata, family = binomial(link = "logit"))

res.1b <- mmi(fit.r = fit.r1, fit.x = fit.x1,
              fit.y = fit.y2, sims = 40, conditional = FALSE,
              covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.1b

#-----#
# Example 2-a: Continuous Outcome, Conditional on Covariates
#-----#
# For conditional = TRUE, need to create data with centered covariates
# copy data
sdata.c <- sdata
# center quantitative covariate(s)
sdata.c$C.num <- scale(sdata.c$C.num, center = TRUE, scale = FALSE)
# center binary (or categorical) covariates(s)
# only necessary if the desired baseline level is NOT the default baseline level.
sdata.c$C.bin <- relevel(sdata.c$C.bin, ref = "1")

# fit mediator and outcome models
fit.m1 <- lm(M.num ~ R + C.num + C.bin, data = sdata.c)
fit.m2 <- glm(M.bin ~ R + C.num + C.bin, data = sdata.c,
             family = binomial(link = "logit"))
fit.m3 <- polr(M.cat ~ R + C.num + C.bin, data = sdata.c)
fit.x2 <- lm(X ~ R + C.num + C.bin, data = sdata.c)
fit.y1 <- lm(Y.num ~ R + M.num + M.bin + M.cat + X + C.num + C.bin,
            data = sdata.c)

res.2a <- mmi(fit.x = fit.x2,
              fit.y = fit.y1, sims = 40, conditional = TRUE,
              covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.2a

```

```

#-----#
# Example 2-b: Binary Outcome, Conditional on Covariates
#-----#
fit.y2 <- glm(Y.bin ~ R + M.num + M.bin + M.cat + X + C.num + C.bin,
             data = sdata.c, family = binomial(link = "logit"))

res.2b <- mmi(fit.x = fit.x2,
             fit.y = fit.y2, sims = 40, conditional = TRUE,
             covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.2b

#-----#
# Example 3: Case with Multiple Intermediate Confounders
#-----#
fit.r <- SumStat(R ~ C, data = idata, weight = "IPW")
fit.x1 <- lm(X1 ~ R + C, data = idata)
fit.x2 <- lm(X2 ~ R + C, data = idata)
fit.x3 <- lm(X3 ~ R + C, data = idata)
fit.y <- lm(Y ~ R + M + X1 + X2 + X3 + C, data = idata)

res.3 <- mmi(fit.r = fit.r, fit.x = list(fit.x1, fit.x2, fit.x3),
             fit.y = fit.y, sims = 40, conditional = FALSE,
             covariates = "C", group = "R", seed = 111)
res.3

```

plot

Visualize sensitivity Objects

Description

S3 methods visualizing results for some objects generated by sensitivity.

Usage

```
## S3 method for class 'sensitivity'
plot(x, xlim = c(0, 0.3), ylim = c(0, 0.3), ...)
```

Arguments

x	an object of class sensitivity.
xlim	limits of the x-axis of contour plots.
ylim	limits of the y-axis of contour plots.
...	Other arguments for future usage.

pocr

*Product-of-Coefficients-Regression Estimation Method***Description**

'pocr' is used to estimate the initial disparity, disparity reduction, and disparity remaining for causal decomposition analysis, using the product-of-coefficients-regression estimation method proposed by Park et al. (2021+).

Usage

```
pocr(fit.x = NULL, fit.m, fit.y, group, covariates, sims = 100, conf.level = .95,
     cluster = NULL, long = TRUE, mc.cores = 1L, seed = NULL)
```

Arguments

fit.x	a fitted model object for intermediate confounder. Can be of class 'lm'. Only necessary if the mediator is categorical. Default is 'NULL'.
fit.m	a fitted model object for mediator. Can be of class 'lm', 'glm', 'multinom', or 'polr'.
fit.y	a fitted model object for outcome. Can be of class 'lm'.
group	a character string indicating the name of the social group indicator such as race or gender (treatment) used in the models. The social group indicator can be categorical with two or more categories (two- or multi-valued factor).
covariates	a vector containing the name of the covariate variable(s) used in the models. Each covariate can be categorical with two or more categories (two- or multi-valued factor) or continuous (numeric).
sims	number of Monte Carlo draws for nonparametric bootstrap.
conf.level	level of the returned two-sided confidence intervals, which are estimated by the nonparametric percentile bootstrap method. Default is to return the 2.5 and 97.5 percentiles of the simulated quantities.
cluster	a vector of cluster indicators for the bootstrap. If provided, the cluster bootstrap is used. Default is 'NULL'.
long	a logical value. If 'TRUE', the output will contain the entire sets of estimates for all bootstrap samples. Default is 'TRUE'.
mc.cores	The number of cores to use. Must be exactly 1 on Windows.
seed	seed number for the reproducibility of results. Default is 'NULL'.

Details

This function returns the point estimates of the initial disparity, disparity reduction, and disparity remaining for a categorical social group indicator (treatment) and a variety of types of outcome and mediator(s) in causal decomposition analysis. It also returns nonparametric percentile bootstrap confidence intervals for each estimate.

The definition of the initial disparity, disparity reduction, and disparity remaining can be found in `help('mmi')`. As opposed to the `'mmi'` and `'smi'` function, this function uses the product-of-coefficients-regression method suggested by Park et al. (2021+). It always make the inference conditional on baseline covariates. Therefore, users need to center the data before fitting the models. See the reference for more details.

As of version 0.1.0, the mediator model (`'fit.m'`) can be of class `'lm'`, `'glm'`, `'multinom'`, or `'polr'`, corresponding respectively to the linear regression models and generalized linear models, multinomial log-linear models, and ordered response models. The outcome model (`'fit.y'`) can be of class `'lm'`. The intermediate confounder model (`'fit.x'`) can also be of class `'lm'` and only necessary when the mediator is categorical.

Value

<code>result</code>	a matrix containing the point estimates of the initial disparity, disparity remaining, and disparity reduction, and the percentile bootstrap confidence intervals for each estimate.
<code>all.result</code>	a matrix containing the point estimates of the initial disparity, disparity remaining, and disparity reduction for all bootstrap samples. Returned if <code>'long'</code> is <code>'TRUE'</code> .

Author(s)

Suyeon Kang, University of Central Florida, <suyeon.kang@ucf.edu>; Soojin Park, University of California, Riverside, <soojinp@ucr.edu>.

References

Park, S., Kang, S., and Lee, C. (2024). "Choosing an optimal method for causal decomposition analysis with continuous outcomes: A review and simulation study", *Sociological methodology*, 54(1), 92-117.

See Also

[mmi](#), [smi](#)

Examples

```
data(sdata)

# To be conditional on covariates, first create data with centered covariates
# copy data
sdata.c <- sdata
# center quantitative covariate(s)
sdata.c$C.num <- scale(sdata.c$C.num, center = TRUE, scale = FALSE)
# center binary (or categorical) covariates(s)
# only neccessary if the desired baseline level is NOT the default baseline level.
sdata.c$C.bin <- relevel(sdata.c$C.bin, ref = "1")

#-----#
# Example 1: Continuous Mediator
```

```

#-----#
fit.m1 <- lm(M.num ~ R + C.num + C.bin, data = sdata.c)
fit.y1 <- lm(Y.num ~ R + M.num + M.num:R + X +
            C.num + C.bin, data = sdata.c)
res1 <- pocr(fit.m = fit.m1, fit.y = fit.y1, sims = 40,
            covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res1

#-----#
# Example 2: Binary Mediator
#-----#
fit.x1 <- lm(X ~ R + C.num + C.bin, data = sdata.c)
fit.m2 <- glm(M.bin ~ R + C.num + C.bin, data = sdata.c,
            family = binomial(link = "logit"))
fit.y2 <- lm(Y.num ~ R + M.bin + M.bin:R + X +
            C.num + C.bin, data = sdata.c)
res2 <- pocr(fit.x = fit.x1, fit.m = fit.m2, fit.y = fit.y2,
            sims = 40, covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res2

#-----#
# Example 3: Ordinal Mediator
#-----#
require(MASS)
fit.m3 <- polr(M.cat ~ R + C.num + C.bin, data = sdata.c)
fit.y3 <- lm(Y.num ~ R + M.cat + M.cat:R + X +
            C.num + C.bin, data = sdata.c)
res3 <- pocr(fit.x = fit.x1, fit.m = fit.m3, fit.y = fit.y3,
            sims = 40, covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res3

#-----#
# Example 4: Nominal Mediator
#-----#
require(nnet)
fit.m4 <- multinom(M.cat ~ R + C.num + C.bin, data = sdata.c)
res4 <- pocr(fit.x = fit.x1, fit.m = fit.m4, fit.y = fit.y3,
            sims = 40, covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res4

```

sdata

Synthetic Data for Illustration

Description

A randomly generated dataset containing 1000 rows and 9 columns with no missing values.

Usage

```
sdata
```

Format

A data frame containing the following variables. The data are provided only for explanatory purposes. The mediators are assumed to be independent of each other.

- C.num:** A quantitative covariate.
- C.bin:** A binary covariates with a value of 0 or 1.
- R:** A group indicator with four levels.
- X:** A quantitative intermediate confounder between a mediator and the outcome.
- M.num:** A quantitative mediator.
- M.bin:** A binary mediator with a value of 0 or 1.
- M.cat:** A categorical mediator with three levels.
- Y.num:** A quantitative outcome.
- Y.bin:** A binary outcome with a value of 0 or 1.

Details

Note that all the variables are randomly generated using the dataset used in Park et al. (2024).

References

Park, S., Kang, S., and Lee, C. (2024). "Choosing an optimal method for causal decomposition analysis with continuous outcomes: A review and simulation study", *Sociological methodology*, 54(1), 92-117.

sensitivity	<i>Sensitivity Analysis Using R-Squared Values for Causal Decomposition Analysis</i>
-------------	--

Description

The function 'sensitivity' is used to implement sensitivity analysis for the causal decomposition analysis, using R-squared values as sensitivity parameters.

Usage

```
sensitivity(boot.res, fit.y, fit.m = NULL, mediator = NULL, covariates, group, sel.lev.group, conf.level = 0.95, max.rsq = 0.3)
```

Arguments

<code>boot.res</code>	bootstrap results from an object fitted by the 'smi' function.
<code>fit.y</code>	outcome model used in fitting the 'smi' function. Can be of class 'lm' or 'glm'.
<code>fit.m</code>	mediator model used in fitting the 'smi' function. Can be of class 'lm', 'glm', or 'polr'.
<code>mediator</code>	a vector containing the name of mediator used in the models.
<code>covariates</code>	a vector containing the name of the covariate variable(s) used in the models. Each covariate can be categorical with two or more categories (two- or multi-valued factor) or continuous (numeric).
<code>group</code>	a character string indicating the name of the social group indicator such as race or gender (treatment) used in the models. The social group indicator can be categorical with two or more categories (two- or multi-valued factor).
<code>sel.lev.group</code>	a level of categorical social group indicator (treatment) which is to be compared with the reference level.
<code>conf.level</code>	level of the returned two-sided confidence intervals, which are estimated by the nonparametric percentile bootstrap method. Default is .95, which returns the 2.5 and 97.5 percentiles of the simulated quantities.
<code>max.rsq</code>	upper limit of the two sensitivity parameters (R-squared values). Once it is set, the R-squared values between 0 and this upper limit are explored to draw the sensitivity contour plots. Default is 0.3.

Details

This function is used to implement sensitivity analysis for the causal decomposition analysis, using two sensitivity parameters: (i) the R-squared value between the outcome and unobserved confounder given the social group indicator (treatment), intermediate confounder, mediator, and covariates; and (ii) the R-squared value between the mediator and unobserved confounder given the social group indicator (treatment), intermediate confounder, and covariates (Park et al., 2023).

As of version 0.1.0, 'boot.res' must be fitted by the 'smi' function with a single mediator, which can be of class 'lm', 'glm', or 'polr'.

Value

<code>call</code>	original function call.
<code>disparity.reduction</code>	a matrix containing the estimated disparity reduction value along with lower and upper limits of the confidence interval, for each combination of the two sensitivity parameters, assuming those two are equal.
<code>disparity.remaining</code>	a matrix containing the estimated disparity remaining value along with lower and upper limits of the confidence interval, for each combination of the two sensitivity parameters, assuming those two are equal.
<code>rm</code>	R-squared values between the mediator and unobserved confounder (first sensitivity parameter), which are explored for the sensitivity analysis.

ry	R-squared values between the outcome and unobserved confounder, (second sensitivity parameter), which are explored for the sensitivity analysis.
red	a matrix containing the estimated disparity reduction values given each combination of the two sensitivity parameters.
lower_red	a matrix containing the lower limit of disparity reduction given each combination of the two sensitivity parameters.
rem	a matrix containing the estimated disparity remaining values given each combination of the two sensitivity parameters.
lower_rem	a matrix containing the lower limit of disparity remaining given each combination of the two sensitivity parameters.
RV_red	robustness value for disparity reduction, which represents the strength of association that will explain away the estimated disparity reduction.
RV_red_alpha	robustness value for disparity reduction, which represents the strength of association that will change the significance of the estimated disparity reduction at the given significance level, assuming an equal association to the mediator and the outcome.
RV_rem	robustness value for disparity remaining, which represents the strength of association that will explain away the estimated disparity remaining.
RV_rem_alpha	robustness value for disparity remaining, which represents the strength of association that will change the significance of the estimated disparity remaining at the given significance level, assuming an equal association to the mediator and the outcome.

Author(s)

Suyeon Kang, University of Central Florida, <suyeon.kang@ucf.edu>; Soojin Park, University of California, Riverside, <soojinp@ucr.edu>.

References

Park, S., Kang, S., Lee, C., & Ma, S. (2023). Sensitivity analysis for causal decomposition analysis: Assessing robustness toward omitted variable bias, *Journal of Causal Inference*, 11(1), 20220031.

See Also

[smi](#)

Examples

```
data(sMIDUS)

# Center covariates
sMIDUS$age <- scale(sMIDUS$age, center = TRUE, scale = FALSE)
sMIDUS$stroke <- scale(sMIDUS$stroke, center = TRUE, scale = FALSE)
sMIDUS$T2DM <- scale(sMIDUS$T2DM, center = TRUE, scale = FALSE)
sMIDUS$heart <- scale(sMIDUS$heart, center = TRUE, scale = FALSE)

fit.m <- lm(edu ~ racesex + age + stroke + T2DM + heart, data = sMIDUS)
```

```
fit.y <- lm(health ~ racesex + lowchildSES + abuse + edu + racesex:edu +
           age + stroke + T2DM + heart, data = sMIDUS)
smiRes <- smi(fit.m = fit.m, fit.y = fit.y, sims = 40, conf.level = .95,
             covariates = c("age", "stroke", "T2DM", "heart"), group = "racesex", seed = 227)
sensRes <- sensitivity(boot.res = smiRes, fit.m = fit.m, fit.y = fit.y, mediator = "edu",
                     covariates = c("age", "stroke", "T2DM", "heart"), group = "racesex",
                     sel.lev.group = "4", max.rsq = 0.3)

sensRes
names(sensRes)
# Create sensitivity contour plots
plot(sensRes)
```

smi	<i>Single-Mediator-Imputation Estimation Method</i>
-----	---

Description

'smi' is used to estimate the initial disparity, disparity reduction, and disparity remaining for causal decomposition analysis, using the single-mediator-imputation estimation method described by Park, Kang, and Lee (2024).

Usage

```
smi(fit.r = NULL, fit.m, fit.y, group, covariates, sims = 100, conf.level = .95,
    conditional = TRUE, cluster = NULL, long = TRUE, mc.cores = 1L, seed = NULL)
```

Arguments

fit.r	a fitted model object for social group indicator (treatment). Can be of class 'CBPS' or 'SumStat'. Default is 'NULL'. Only necessary if 'conditional' is 'FALSE'.
fit.m	a fitted model object for mediator. Can be of class 'lm', 'glm', 'multinom', or 'polr'.
fit.y	a fitted model object for outcome. Can be of class 'lm' or 'glm'.
group	a character string indicating the name of the social group indicator such as race or gender (treatment) used in the models. The social group indicator can be categorical with two or more categories (two- or multi-valued factor).
covariates	a vector containing the name of the covariate variable(s) used in the models. Each covariate can be categorical with two or more categories (two- or multi-valued factor) or continuous (numeric).
sims	number of Monte Carlo draws for nonparametric bootstrap.
conf.level	level of the returned two-sided confidence intervals, which are estimated by the nonparametric percentile bootstrap method. Default is .95, which returns the 2.5 and 97.5 percentiles of the simulated quantities.

<code>conditional</code>	a logical value. If 'TRUE', the function will return the estimates conditional on those covariate values; and all covariates in mediator and outcome models need to be centered prior to fitting. Default is 'TRUE'. If 'FALSE', 'fit.r' needs to be specified.
<code>cluster</code>	a vector of cluster indicators for the bootstrap. If provided, the cluster bootstrap is used. Default is 'NULL'.
<code>long</code>	a logical value. If 'TRUE', the output will contain the entire sets of estimates for all bootstrap samples. Default is 'TRUE'.
<code>mc.cores</code>	The number of cores to use. Must be exactly 1 on Windows.
<code>seed</code>	seed number for the reproducibility of results. Default is 'NULL'.

Details

This function returns the point estimates of the initial disparity, disparity reduction, and disparity remaining for a categorical social group indicator (treatment) and a variety of types of outcome and mediator(s) in causal decomposition analysis. It also returns nonparametric percentile bootstrap confidence intervals for each estimate.

The definition of the initial disparity, disparity reduction, and disparity remaining can be found in `help('mmi')`. As opposed to the 'mmi' function, this function uses the single-mediator-imputation method suggested by Park et al. (2021+). See the reference for more details.

If one wants to make the inference conditional on baseline covariates, set 'conditional = TRUE' and center the data before fitting the models.

As of version 0.1.0, the mediator model ('fit.m') can be of class 'lm', 'glm', 'multinom', or 'polr', corresponding respectively to the linear regression models and generalized linear models, multinomial log-linear models, and ordered response models. The outcome model ('fit.y') can be of class 'lm' or 'glm'. Also, the social group (treatment) model ('fit.r') can be of class 'CBPS' or 'SumStat', both of which use the propensity score weighting. It is only necessary when 'conditional = FALSE'.

Value

<code>result</code>	a matrix containing the point estimates of the initial disparity, disparity remaining, and disparity reduction, and the percentile bootstrap confidence intervals for each estimate.
<code>all.result</code>	a matrix containing the point estimates of the initial disparity, disparity remaining, and disparity reduction for all bootstrap samples. Returned if 'long' is 'TRUE'.
<code>alpha.r</code>	a vector containing the estimates of the regression coefficient of the social group indicator (treatment) in the mediator. Not needed unless sensitivity analysis is conducted afterwards.
<code>se.gamma</code>	a vector containing the estimates of standard error of the regression coefficient of the mediator in the outcome model. Not needed unless sensitivity analysis is conducted afterwards.

Author(s)

Suyeon Kang, University of Central Florida, <suyeon.kang@ucf.edu>; Soojin Park, University of California, Riverside, <soojinp@ucr.edu>.

References

Park, S., Kang, S., and Lee, C. (2024). "Choosing an optimal method for causal decomposition analysis with continuous outcomes: A review and simulation study", *Sociological methodology*, 54(1), 92-117.

See Also

[mmi](#), [sensitivity](#)

Examples

```
data(sdata)

#-----#
# Example 1-a: Continuous Outcome
#-----#
require(PSweight)
fit.r1 <- SumStat(R ~ C.num + C.bin, data = sdata, weight = "IPW")
require(CBPS)
fit.r2 <- CBPS(R ~ C.num + C.bin, data = sdata, method = "exact",
              standardize = "TRUE")

# Continuous mediator
fit.m1 <- lm(M.num ~ R + C.num + C.bin, data = sdata)
fit.y1 <- lm(Y.num ~ R + M.num + X + C.num + C.bin, data = sdata)
res.1a1 <- smi(fit.r = fit.r1, fit.m = fit.m1,
              fit.y = fit.y1, sims = 40, conditional = FALSE,
              covariates = c("C.num", "C.bin"), group = "R", seed = 32)
res.1a1

# Binary mediator
fit.m2 <- glm(M.bin ~ R + C.num + C.bin, data = sdata,
              family = binomial(link = "logit"))
fit.y2 <- lm(Y.num ~ R + M.bin + X + C.num + C.bin, data = sdata)
res.1a2 <- smi(fit.r = fit.r1, fit.m = fit.m2,
              fit.y = fit.y2, sims = 40, conditional = FALSE,
              covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.1a2

# Categorical mediator
require(MASS)
fit.m3 <- polr(M.cat ~ R + C.num + C.bin, data = sdata)
fit.y3 <- lm(Y.num ~ R + M.cat + X + C.num + C.bin, data = sdata)
res.1a3 <- smi(fit.r = fit.r1, fit.m = fit.m3,
              fit.y = fit.y3, sims = 40, conditional = FALSE,
              covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.1a3
```

```

require(nnet)
fit.m4 <- multinom(M.cat ~ R + C.num + C.bin, data = sdata)
res.1a4 <- smi(fit.r = fit.r1, fit.m = fit.m4,
              fit.y = fit.y3, sims = 40, conditional = FALSE,
              covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.1a4

#-----#
# Example 1-b: Binary Outcome
#-----#
# Continuous mediator
fit.y1 <- glm(Y.bin ~ R + M.num + X + C.num + C.bin,
              data = sdata, family = binomial(link = "logit"))
res.1b1 <- smi(fit.r = fit.r1, fit.m = fit.m1,
              fit.y = fit.y1, sims = 40, conditional = FALSE,
              covariates = c("C.num", "C.bin"), group = "R", seed = 32)
res.1b1

# Binary mediator
fit.y2 <- glm(Y.bin ~ R + M.bin + X + C.num + C.bin,
              data = sdata, family = binomial(link = "logit"))
res.1b2 <- smi(fit.r = fit.r1, fit.m = fit.m2,
              fit.y = fit.y2, sims = 40, conditional = FALSE,
              covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.1b2

# Categorical mediator
fit.y3 <- glm(Y.bin ~ R + M.cat + X + C.num + C.bin,
              data = sdata, family = binomial(link = "logit"))
res.1b3 <- smi(fit.r = fit.r1, fit.m = fit.m3,
              fit.y = fit.y3, sims = 40, conditional = FALSE,
              covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.1b3

res.1b4 <- smi(fit.r = fit.r1, fit.m = fit.m4,
              fit.y = fit.y3, sims = 40, conditional = FALSE,
              covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.1b4

#-----#
# Example 2-a: Continuous Outcome, Conditional on Covariates
#-----#
# For conditional = T, need to create data with centered covariates
# copy data
sdata.c <- sdata
# center quantitative covariate(s)
sdata.c$C.num <- scale(sdata.c$C.num, center = TRUE, scale = FALSE)
# center binary (or categorical) covariates(s)
# only necessary if the desired baseline level is NOT the default baseline level.
sdata.c$C.bin <- relevel(sdata.c$C.bin, ref = "1")

# Continuous mediator

```

```

fit.y1 <- lm(Y.num ~ R + M.num + X + C.num + C.bin, data = sdata.c)
fit.m1 <- lm(M.num ~ R + C.num + C.bin, data = sdata.c)
res.2a1 <- smi(fit.m = fit.m1,
               fit.y = fit.y1, sims = 40, conditional = TRUE,
               covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.2a1

# Binary mediator
fit.y2 <- lm(Y.num ~ R + M.bin + X + C.num + C.bin, data = sdata.c)
fit.m2 <- glm(M.bin ~ R + C.num + C.bin, data = sdata.c,
              family = binomial(link = "logit"))
res.2a2 <- smi(fit.m = fit.m2,
               fit.y = fit.y2, sims = 40, conditional = TRUE,
               covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.2a2

# Categorical mediator
fit.y3 <- lm(Y.num ~ R + M.cat + X + C.num + C.bin, data = sdata.c)
fit.m3 <- polr(M.cat ~ R + C.num + C.bin, data = sdata.c)
res.2a3 <- smi(fit.m = fit.m3,
               fit.y = fit.y3, sims = 40, conditional = TRUE,
               covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.2a3

fit.m4 <- multinom(M.cat ~ R + C.num + C.bin, data = sdata.c)
res.2a4 <- smi(fit.m = fit.m4,
               fit.y = fit.y3, sims = 40, conditional = TRUE,
               covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.2a4

#-----#
# Example 2-b: Binary Outcome, Conditional on Covariates
#-----#

# Continuous mediator
fit.y1 <- glm(Y.bin ~ R + M.num + X + C.num + C.bin,
              data = sdata.c, family = binomial(link = "logit"))
res.2b1 <- smi(fit.m = fit.m1,
               fit.y = fit.y1, sims = 40, conditional = TRUE,
               covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.2b1

# Binary mediator
fit.y2 <- glm(Y.bin ~ R + M.bin + X + C.num + C.bin,
              data = sdata.c, family = binomial(link = "logit"))
res.2b2 <- smi(fit.m = fit.m2,
               fit.y = fit.y2, sims = 40, conditional = TRUE,
               covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.2b2

# Categorical mediator
fit.y3 <- glm(Y.bin ~ R + M.cat + X + C.num + C.bin,
              data = sdata.c, family = binomial(link = "logit"))
res.2b3 <- smi(fit.m = fit.m3,

```

```
fit.y = fit.y3, sims = 40, conditional = TRUE,
covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.2b3

res.2b4 <- smi(fit.m = fit.m4,
fit.y = fit.y3, sims = 40, conditional = TRUE,
covariates = c("C.num", "C.bin"), group = "R", seed = 111)
res.2b4
```

sMIDUS	<i>Synthetic Data Generated Based on the Midlife Development in the U.S. (MIDUS) Study</i>
--------	--

Description

This is a synthetic dataset that includes variables from the Midlife Development in the U.S. (MIDUS) study. It has been artificially generated based on the actual MIDUS data, which is not publicly available due to confidentiality concerns. The synthetic data set consists of 1948 rows and 9 columns, with no missing values.

Usage

sMIDUS

Format

- A data frame containing the following variables.
- health:** cardiovascular health score.
 - racesex:** race-gender groups with four levels (1: White men, 2: White women, 3: Black men, 4: Black women).
 - lowchildSES:** socioeconomic status (SES) in the childhood.
 - abuse:** adverse experience in the childhood.
 - edu:** education level.
 - age:** age.
 - stroke:** genetic vulnerability with a value of 0 or 1.
 - T2DM:** genetic vulnerability with a value of 0 or 1.
 - heart:** genetic vulnerability with a value of 0 or 1.

Details

Note that all the variables are fabricated using the actual MIDUS data used in Park et al. (2023).

References

Park, S., Kang, S., Lee, C., & Ma, S. (2023). Sensitivity analysis for causal decomposition analysis: Assessing robustness toward omitted variable bias, *Journal of Causal Inference*, 11(1), 20220031.

Index

* **datasets**

idata, [2](#)

sdata, [14](#)

sMIDUS, [23](#)

causal.decomp (causal.decomp-package), [2](#)

causal.decomp-package, [2](#)

idata, [2](#)

ind.decomp, [3](#), [7](#)

ind.sens, [5](#), [5](#)

mmi, [7](#), [13](#), [20](#)

plot, [11](#)

pocr, [12](#)

sdata, [14](#)

sensitivity, [15](#), [20](#)

smi, [9](#), [13](#), [17](#), [18](#)

sMIDUS, [23](#)