Package: TransHDM (via r-universe)

May 23, 2026

Title High-Dimensional Mediation Analysis via Transfer Learning

Version 1.0.1

Description Provides a framework for high-dimensional mediation analysis using transfer learning. The main function TransHDM() integrates large-scale source data to improve the detection power of potential mediators in small-sample target studies. It addresses data heterogeneity via transfer regularization and debiased estimation while controlling the false discovery rate. The package also includes utilities for data generation (gen_simData_homo(), gen_simData_hetero()), baseline methods such as lasso() and dblasso(), sure independence screening via SIS(), and model diagnostics through source_detection(). The methodology is described in Pan et al. (2025) <doi:10.1093/bib/bbaf460>.

License GPL (>= 3)

URL https://github.com/Gaohuer/TransHDM

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports glmnet (>= 4.1-10), caret (>= 7.0-1), MASS (>= 7.3-61), doParallel (>= 1.0.17), foreach (>= 1.5.2), HDMT (>= 1.0.5)

Suggests knitr (>= 1.50), rmarkdown (>= 2.30), spelling (>= 2.3.2)

Language en-US

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev

Repository https://gaohuer.r-universe.dev

Date/Publication 2026-03-24 10:13:59 UTC

RemoteUrl https://github.com/gaohuer/transhdm

RemoteRef HEAD

RemoteSha 06b1d7b919217342e9e8137a3a196d7bc03e7211

Contents

`dblasso` *Fit Debiased LASSO with Transfer Learning*

Description

Fits a debiased LASSO regression model under transfer learning framework, supporting feature selection and coefficient estimation by combining target and source data.

Usage

```
dblasso(
  target,
  source = NULL,
  transfer = FALSE,
  level = 0.95,
  lambda = "lambda.1se"
)
```

Arguments

`target`	A list containing two elements: • x: Feature matrix of target data (numeric matrix) • y: Response vector of target data (numeric vector) Required.
`source`	A list (optional, default: NULL) containing two elements: • x: Feature matrix of source data (numeric matrix) • y: Response vector of source data (numeric vector) Used when transfer = TRUE.
`transfer`	A logical value (default: FALSE) indicating whether to enable transfer learning (combining source data with target data for estimation).

`level` A numeric value (default: 0.95) specifying confidence level for confidence intervals.

`lambda` A string specifying criterion for selecting regularization parameter:

- 'lambda.min': Lambda value giving minimum cross-validation error
- 'lambda.1se': Largest lambda within 1 standard error of minimum error

Value

A list containing:

- dbcoef.hat: Debiased LASSO coefficient vector (including intercept)
- coef.hat: Original LASSO coefficient vector
- CI: Data frame with confidence intervals (lb = lower bound, ub = upper bound)
- var.est: Variance estimates for debiased coefficients
- se.est: Standard errors for debiased coefficients
- P.value: Vector of p-values for coefficients

Examples

```
# Prepare target and source data
target <- list(x = matrix(rnorm(100 * 20), 100, 20), y = rnorm(100))
source <- list(x = matrix(rnorm(200 * 20), 200, 20), y = rnorm(200))

# Non-transfer mode
result_no_transfer <- dblasso(target = target, transfer = FALSE,
                              level = 0.95, lambda = 'lambda.min')
summary(result_no_transfer)

# Transfer learning mode
result_transfer <- dblasso(target = target, source = source, transfer = TRUE,
                           level = 0.95, lambda = 'lambda.min')
summary(result_transfer)
```

---

gen_simData_hetero *Simulated Dataset Generation for High-Dimensional Mediation Analysis*

Description

Generates synthetic datasets mimicking high-dimensional mediation structures, optionally incorporating transferable source data under varying covariate correlation and heterogeneity levels. This function supports heterogeneous settings for data generation.

Usage

```
gen_simData_hetero(
  n = 100,
  p_x = 5,
  rho = 0,
  p_m = 100,
  h = 0,
  source = FALSE,
  transferable = TRUE,
  seed = NULL
)
```

Arguments

n	Integer. Number of observations (sample size). Default is 100.
p_x	Integer. Number of covariates (confounders). Default is 5.
rho	Numeric. Correlation coefficient (0–1) controlling correlation between mediators. Default is 0 (no correlation).
p_m	Integer. Number of mediators. Default is 100.
h	Integer. Degree of heterogeneity (for source data). Default is 0.
source	Logical. If TRUE, generate source (external) dataset. Default is FALSE.
transferable	Logical. If TRUE, generates a transferable source dataset sharing mediator-outcome structure with the target. Default is TRUE.
seed	Integer. Optional random seed for reproducibility. Default is NULL.

Details

This function generates data according to a structural equation model (SEM):

- Treatment D depends linearly on covariates X.
- Mediators M depend on D and X, with residual correlation controlled by rho.
- Outcome Y depends on D, M, and X.
- If source = TRUE, then a source dataset is simulated with potentially transferable mechanisms.

Value

A list with the following components:

- data: A data.frame of dimension n × (2 + p_m + p_x) containing outcome (Y), treatment (D), mediators (M1–Mp_m), and covariates (X1–Xp_x).
- coef: A named list of true model coefficients (alpha1, alpha2, beta1, beta2, beta4, etc.).

Examples

```
source_data <- gen_simData_hetero(
  n = 100, p_x = 5, rho = 0, p_m = 100, h = 0,
  source = TRUE, transferable = TRUE, seed = 123
)
source_data <- gen_simData_hetero(
  n = 100, p_x = 5, rho = 0, p_m = 100, h = 0,
  source = FALSE, transferable = TRUE, seed = 123
)
```

---

gen_simData_homo *Simulated Dataset Generation for High-Dimensional Mediation Analysis*

Description

Generates synthetic datasets mimicking high-dimensional mediation structures, optionally incorporating transferable source data under varying covariate correlation and heterogeneity levels. This function supports homogeneous settings for data generation.

Usage

```
gen_simData_homo(
  n = 100,
  p_x = 5,
  rho = 0,
  p_m = 100,
  h = 0,
  source = FALSE,
  transferable = TRUE,
  seed = NULL
)
```

Arguments

n	Integer. Number of observations (sample size). Default is 100.
p_x	Integer. Number of covariates (confounders). Default is 5.
rho	Numeric. Correlation coefficient (0–1) controlling correlation between mediators. Default is 0 (no correlation).
p_m	Integer. Number of mediators. Default is 100.
h	Integer. Degree of heterogeneity (for source data). Default is 0.
source	Logical. If TRUE, generate source (external) dataset. Default is FALSE.
transferable	Logical. If TRUE, generates a transferable source dataset sharing mediator-outcome structure with the target. Default is TRUE.
seed	Integer. Optional random seed for reproducibility. Default is NULL.

Details

This function generates data according to a structural equation model (SEM):

- Treatment D depends linearly on covariates X.
- Mediators M depend on D and X, with residual correlation controlled by `rho`.
- Outcome Y depends on D, M, and X.
- If `source = TRUE`, then a source dataset is simulated with potentially transferable mechanisms.

Value

A list with the following components:

- `data`: A data.frame of dimension `n * (2 + p_m + p_x)` containing outcome (Y), treatment (D), mediators (M1–Mp_m), and covariates (X1–Xp_x).
- `coef`: A named list of true model coefficients (`alpha1`, `alpha2`, `beta1`, `beta2`, `beta3`, `beta4`, etc.).

Examples

```
source_data <- gen_simData_homo(
  n = 100, p_x = 5, rho = 0, p_m = 100, h = 0,
  source = TRUE, transferable = TRUE, seed = 123
)
target_data <- gen_simData_homo(
  n = 100, p_x = 5, rho = 0, p_m = 100, h = 0,
  source = FALSE, transferable = TRUE, seed = 123
)
```

---

lasso *Fit LASSO Regression with Transfer Learning*

Description

Fits a LASSO (Least Absolute Shrinkage and Selection Operator) regression model under a transfer learning framework. Supports feature selection and coefficient estimation by combining target data and source data.

Usage

```
lasso(target, source = NULL, transfer = FALSE, lambda = "lambda.1se")
```

Arguments

target	A list containing two elements: • x: Feature matrix of target data (numeric matrix) • y: Response vector of target data (numeric vector) Required.
source	A list (optional, default: NULL) containing two elements: • x: Feature matrix of source data (numeric matrix) • y: Response vector of source data (numeric vector) Used when transfer = TRUE.
transfer	A logical value (default: FALSE) indicating whether to enable transfer learning mode (combine source data with target data).
lambda	A string (default: 'lambda.1se') specifying the criterion for selecting regularization parameter: • 'lambda.min': Lambda value that gives minimum cross-validation error • 'lambda.1se': Largest lambda value within 1 standard error of the minimum error

Value

A numeric vector `coef` containing LASSO coefficient estimates (including intercept).

Examples

```
# Prepare target and source data
target <- list(x = matrix(rnorm(100 * 20), 100, 20), y = rnorm(100))
source <- list(x = matrix(rnorm(200 * 20), 200, 20), y = rnorm(200))

# Non-transfer mode
coef_no_transfer <- lasso(target = target, transfer = FALSE, lambda = 'lambda.min')
print(coef_no_transfer)
summary(coef_no_transfer)

# Transfer learning mode
coef_transfer <- lasso(target = target, source = source, transfer = TRUE, lambda = 'lambda.min')
print(coef_transfer)
summary(coef_transfer)
```

`SIS` *Sure Independence Screening for High-Dimensional Mediation Analysis*

Description

This function performs dimension reduction for high-dimensional mediation analysis using Sure Independence Screening (SIS). Mediators are ranked based on the product of their marginal associations with the exposure and the outcome, and the top-ranked mediators are retained for downstream analysis.

Usage

```
SIS(
  target_data,
  source_data = NULL,
  Y,
  D,
  M,
  X,
  topN = NULL,
  transfer = FALSE,
  verbose = TRUE,
  ncore = 1,
  dblasso_method = FALSE
)
```

Arguments

target_data	A data frame containing the target dataset. All variables must be numeric.
source_data	A list of data frames containing source datasets (optional, default: NULL). All variables must be numeric and have the same column names as target_data.
Y	Character string specifying the outcome variable name.
D	Character string specifying the exposure (treatment) variable name.
M	Character vector specifying mediator variable names.
X	Character vector specifying covariate variable names.
topN	An integer specifying the number of mediators to retain after screening. If NULL, the number is automatically determined as $\lceil 2n/\log(n) \rceil$, where n is the target sample size.
transfer	A logical value (default: FALSE) indicating whether to apply transfer learning by incorporating the source dataset in the screening procedure.
verbose	A logical value (default: TRUE) controlling whether progress messages are printed to the console.
ncore	An integer (default: 1) specifying the number of CPU cores for parallel computation.
dblasso_method	A logical value (default: FALSE). If TRUE, the debiased lasso (dblasso) is used to estimate marginal effects. If FALSE, standard linear or generalized linear models are used.

Details

The function supports transfer learning, allowing information from a source dataset to be leveraged to improve screening stability and robustness in the target dataset.

Value

A list with the following components:

- `target_SIS`: A data frame containing the outcome, exposure, selected mediators, and covariates from the target dataset.
- `source_SIS`: A data frame containing the same variables from the source dataset if `transfer = TRUE`; otherwise `NULL`.
- `M_ID_name_SIS`: A character vector of selected mediator names.

Examples

```
set.seed(123)

# Target data
M_target <- matrix(rnorm(200 * 50), nrow = 200)
colnames(M_target) <- paste0("M", 1:50)

target_data <- data.frame(
  Y = rnorm(200),
  D = rnorm(200),
  M_target,
  X1 = rnorm(200)
)

# Source data
M_source <- matrix(rnorm(300 * 50), nrow = 300)
colnames(M_source) <- paste0("M", 1:50)

source_data <- data.frame(
  Y = rnorm(300),
  D = rnorm(300),
  M_source,
  X1 = rnorm(300)
)

# Run SIS
result <- SIS(
  target_data = target_data,
  source_data = source_data,
  Y = "Y",
  D = "D",
  M = paste0("M", 1:50),
  X = "X1",
  transfer = TRUE,
  topN = 10
)
```

```
result$M_ID_name_SIS
```

`source_detection` *Detect Transferable Source Data via Cross-Validation*

Description

Determines whether external source datasets can be effectively transferred to the target data by comparing predictive performance using LASSO regression under a transfer learning framework.

Usage

```
source_detection(
  target_data,
  source_data = NULL,
  Y,
  D,
  M,
  X,
  kfold = 5,
  C0 = 0.05,
  verbose = TRUE
)
```

Arguments

`target_data`	A data frame containing the target dataset. All variables must be numeric.
`source_data`	A list of data frames containing source datasets (optional, default: NULL). All variables must be numeric and have the same column names as target_data.
`Y`	Character string specifying the outcome variable name.
`D`	Character string specifying the exposure (treatment) variable name.
`M`	Character vector specifying mediator variable names.
`X`	Character vector specifying covariate variable names.
`kfold`	Integer (default: 5). Number of folds for cross-validation.
`C0`	Numeric (default: 0.05). Threshold constant for determining transferability. Larger values make the criterion more lenient.
`verbose`	Logical (default: TRUE). Whether to print progress messages.

Value

A list containing:

- transfer.source.id: Indices of source datasets deemed transferable
- source.loss: Mean validation loss for each source dataset
- target.valid.loss: Mean validation loss using target-only model
- T_index: Difference between source loss and target-only loss for each source
- threshold: Calculated transferability threshold
- loss.cv: Full k-fold cross-validation loss matrix

Examples

```
## Reproducible example
set.seed(123)

# Generate synthetic target data
target_data <- data.frame(
  Y  = rnorm(200),
  D  = rnorm(200),
  M1 = rnorm(200),
  M2 = rnorm(200),
  X1 = rnorm(200)
)

# Generate synthetic source data
source1 <- data.frame(
  Y  = rnorm(300),
  D  = rnorm(300),
  M1 = rnorm(300),
  M2 = rnorm(300),
  X1 = rnorm(300)
)

source2 <- data.frame(
  Y  = rnorm(250),
  D  = rnorm(250),
  M1 = rnorm(250),
  M2 = rnorm(250),
  X1 = rnorm(250)
)

# Run source detection
result <- source_detection(
  target_data = target_data,
  source_data = list(source1, source2),
  Y = "Y",
  D = "D",
  M = c("M1", "M2"),
  X = "X1",
  kfold = 5,
```

```
  C0 = 0.05,
  verbose = FALSE
)

# Get Summary
summary(result)

# Transferable source indices
result$transfer.source.id

# Compare validation losses
data.frame(
  Source = c(paste0("Source", seq_along(result$source.loss)), "Target"),
  Loss   = c(result$source.loss, result$target.valid.loss)
)
```

---

summary.dblasso *Summary of Debiased Lasso Inference*

Description

Summary of Debiased Lasso Inference

Usage

```
## S3 method for class 'dblasso'
summary(object, ...)
```

Arguments

object	An object of class "dblasso".
...	Further arguments (currently not used).

Value

An object of class "summary.dblasso".

summary.lasso *Summary of Lasso Regression*

Description

Summary of Lasso Regression

Usage

```
## S3 method for class 'lasso'
summary(object, ...)
```

Arguments

object	A numeric vector of lasso coefficients with names.
...	Further arguments (currently not used).

Value

An object of class "summary.lasso".

summary.source_detection *Summary of Source Detection Results*

Description

Summary of Source Detection Results

Usage

```
## S3 method for class 'source_detection'
summary(object, ...)
```

Arguments

object	An object of class "source_detection".
...	Further arguments (unused).

Value

An object of class "summary.source_detection".

summary.TransHDM *Summary of TransHDM Mediation Analysis*

Description

Summary of TransHDM Mediation Analysis

Usage

```
## S3 method for class 'TransHDM'
summary(object, top = 10, digits = 4, ...)
```

Arguments

object	An object of class "TransHDM".
top	Integer, maximum number of mediators to display.
digits	Number of digits for rounding estimates.
...	Further arguments (unused).

Value

An object of class "summary.TransHDM".

TransHDM *TransHDM: High-Dimensional Mediation Analysis with Transfer Learning*

Description

The TransHDM function performs high-dimensional mediation analysis under a transfer learning framework. It identifies and estimates indirect (mediation) effects of a high-dimensional set of mediators between an exposure and an outcome by integrating a target dataset and a source datasets.

Usage

```
TransHDM(
  target_data,
  source_data = NULL,
  Y,
  D,
  M,
  X,
  transfer = FALSE,
  verbose = TRUE,
  ncore = 1,
```

```
  topN = NULL,
  dblasso_SIS = FALSE
)
```

Arguments

target_data	A data frame containing the target dataset. All variables must be numeric.
source_data	A list of data frames containing source datasets (optional, default: NULL). All variables must be numeric and have the same column names as target_data.
Y	Character string specifying the outcome variable name.
D	Character string specifying the exposure (treatment) variable name.
M	Character vector specifying mediator variable names.
X	Character vector specifying covariate variable names..
transfer	A logical value (default: FALSE) indicating whether to enable transfer learning by incorporating information from source_data.
verbose	A logical value (default: TRUE) controlling whether progress messages are printed to the console.
ncore	An integer (default: 1) specifying the number of CPU cores to use for parallel computation when fitting mediator models.
topN	An integer (default: NULL) specifying the number of mediators to retain after Sure Independence Screening (SIS). If NULL, the number is determined automatically based on the data dimensions.
dblasso_SIS	A logical value (default: FALSE) indicating whether to apply a two-stage procedure combining SIS and debiased Lasso. When TRUE, mediators are first screened via SIS and then debiased Lasso is applied to the reduced set, which is recommended for ultra-high-dimensional settings.

Value

A list with the following components:

- contributions: A data frame of identified mediators containing:
  - mediator: Mediator name
  - alpha: Estimated exposure mediator effect
  - alpha_pv: P-value for the exposure mediator effect
  - beta: Estimated mediator outcome effect
  - beta_pv: P-value for the mediator outcome effect
  - alpha_beta: Estimated indirect (mediation) effect
  - ab_pv: Joint p-value for the indirect effect
  - pa: Proportion of the total effect mediated
- effects: A data frame summarizing the total indirect effect, direct effect, total effect, and proportion mediated.
- IDE_est: A numeric vector of indirect effect estimates for all specified mediators, with non-selected mediators set to zero.
- DE_est: The estimated direct effect of the exposure on the outcome.

References

Pan L, Liu Y, Huang C, Lin R, Yu Y, Qin G. Transfer learning reveals the mediating mechanisms of cross-ethnic lipid metabolic pathways in the association between APOE gene and Alzheimer's disease. Brief Bioinform. 2025;26(5):bbaf460. doi:10.1093/bib/bbaf460

Examples

```
set.seed(123)

# Target data
target_data <- gen_simData_homo(n = 50, p_x = 3, p_m = 20, rho = 0.1)$data

# Source data
source_data <- gen_simData_homo(n = 100, p_x = 3, p_m = 20, rho = 0.1, source = TRUE,
transferable = TRUE)$data

# Run TransHDM
result <- TransHDM(
  target_data = target_data,
  source_data = source_data,
  Y = "Y",
  D = "D",
  M = paste0("M", 1:20),
  X = paste0("X", 1:3),
  transfer = TRUE,
  ncore = 1,
  topN = 10
)
summary(result)
```

Index