Package: hierSDR (via r-universe)

September 9, 2024

Type Package

Title Hierarchical Sufficient Dimension Reduction

Version 0.1

Description Provides semiparametric sufficient dimension reduction for central mean subspaces for heterogeneous data defined by combinations of binary factors (such as chronic conditions). Subspaces are estimated to be hierarchically nested to respect the structure of subpopulations with overlapping characteristics. This package is an implementation of the proposed methodology of Huling and Yu (2021) <doi:10.1111/biom.13546>.

BugReports `https://github.com/jaredhuling/hierSDR/issues`

License GPL-2

Encoding UTF-8

Depends R (>= 3.2.0), MASS, Matrix, locfit, lbfgs

Imports numDeriv, optimx

LazyData TRUE

RoxygenNote 7.1.1

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

RemoteUrl https://github.com/jaredhuling/hiersdr

RemoteRef HEAD

RemoteSha f5fe8e27201983c795fc6983c33c2f61e653c6cd

Contents

angle *Angle between two subspaces*

Description

Measures angle between two subspaces. Smallest value is 0, largest is 90 from http://www4.stat.ncsu.edu/~li/software/GroupI
http://lexinli.biostat.berkeley.edu/softwares/dr/GroupDR.R

Usage

```
angle(B1, B2)
```

Arguments

`B1`	first matrix
`B2`	second matrix

Value

scalar value of the angle between `B1` and `B2`

Examples

```
## case where any relation between b1 and b2 is random
b1 <- matrix(rnorm(10 * 2), ncol = 2)
b2 <- matrix(rnorm(10 * 2), ncol = 2)
angle(b1, b2)

## angle here should be small
b1 <- matrix(rnorm(10 * 2), ncol = 2)
b2 <- b1 + matrix(rnorm(10 * 2, sd = 0.2), ncol = 2)
angle(b1, b2)
```

`hier.phd.nt` *Main hierarchical SDR fitting function*

Description

fits hierarchical SDR models

Usage

```
hier.phd.nt(
  x,
  y,
  z,
  z.combinations,
  d,
  weights = rep(1L, NROW(y)),
  constrain.none.subpop = TRUE,
  pooled = FALSE,
  ...
)
```

Arguments

`x`	an n x p matrix of covariates, where each row is an observation and each column is a predictor
`y`	vector of responses of length n
`z`	an n x C matrix of binary indicators, where each column is a binary variable indicating the presence of a binary variable which acts as a stratifying variable. Each combination of all columns of `z` pertains to a different subpopulation. WARNING: do not use too many binary variables in `z` or else it will quickly result in subpopulations with no observations
`z.combinations`	a matrix of dimensions 2^C x C with each row indicating a different combination of the possible values in `z`. Each combination represents a subpopulation. This is necessary because we need to specify a different structural dimension for each subpopulation, so we need to know the ordering of the subpopulations so we can assign each one a structural dimension
`d`	an integer vector of length 2^C of structural dimensions. Specified in the same order as the rows in `z.combinations`
`weights`	vector of observation weights
`constrain.none.subpop`	should the "none" subpopulation be constrained to be contained in every other subpopulation's dimension reduction subspace? Recommended to set to `TRUE`
`pooled`	should the estimator be a pooled estimator?
`...`	not used

Value

A list with the following elements

- beta a list of estimated sufficient dimension reduction matrices, one for each subpopulation
- directions a list of estimated sufficient dimension reduction directions (i.e. the reduced dimension predictors/variables), one for each subpopulation. These have number of rows equal to the sample size for the subpopulation and number of columns equal to the specified dimensions of the reduced dimension spaces.
- y.list a list of vectors of responses for each subpopulation
- z.combinations the `z.combinations` specified as an input
- cov list of variance covariance matrices for the covariates for each subpopulation
- sqrt.inv.cov list of inverse square roots of the variance covariance matrices for the covariates for each subpopulation. These are used for scaling

Examples

```
library(hierSDR)
```

`hier.sphd` *Main hierarchical sufficient dimension reduction fitting function*

Description

fits hierarchically nested sufficient dimension reduction models

Usage

```
hier.sphd(
  x,
  y,
  z,
  z.combinations,
  d,
  weights = rep(1L, NROW(y)),
  maxit = 250L,
  tol = 1e-09,
  h = NULL,
 opt.method = c("lbfgs2", "lbfgs.x", "bfgs.x", "bfgs", "lbfgs", "spg", "ucminf", "CG",
    "nlm", "nlminb", "newuoa"),
  init.method = c("random", "phd"),
  vic = TRUE,
  grassmann = TRUE,
  nn = NULL,
  nn.try = c(0.15, 0.25, 0.5, 0.75, 0.9, 0.95),
  n.random = 100L,
  optimize.nn = FALSE,
  separate.nn = FALSE,
  constrain.none.subpop = TRUE,
  verbose = TRUE,
  degree = 2,
  pooled = FALSE,
  maxk = 5000,
  ...
)
```

Arguments

`x`	an n x p matrix of covariates, where each row is an observation and each column is a predictor
`y`	vector of responses of length n
`z`	an n x C matrix of binary indicators, where each column is a binary variable indicating the presence of a binary variable which acts as a stratifying variable. Each combination of all columns of `z` pertains to a different subpopulation. WARNING: do not use too many binary variables in `z` or else it will quickly result in subpopulations with no observations
`z.combinations`	a matrix of dimensions 2^C x C with each row indicating a different combination of the possible values in `z`. Each combination represents a subpopulation. This is necessary because we need to specify a different structural dimension for each subpopulation, so we need to know the ordering of the subpopulations so we can assign each one a structural dimension
`d`	an integer vector of length 2^C of structural dimensions. Specified in the same order as the rows in `z.combinations`
`weights`	vector of observation weights
`maxit`	maximum number of iterations for optimization routines
`tol`	convergence tolerance for optimization routines. Defaults to `1e-6`
`h`	bandwidth parameter. By default, a reasonable choice is selected automatically
`opt.method`	optimization method to use. Available choices are `c("lbfgs2", "lbfgs.x", "bfgs.x", "bfgs", "lbfgs", "spg", "ucminf", "CG", "nlm", "nlminb", "newuoa")`
`init.method`	method for parameter initialization. Either `"random"` for random initialization or `"phd"` for a principle Hessian directions initialization approach
`vic`	logical value of whether or not to compute the VIC criterion for dimension determination
`grassmann`	logical value of whether or not to enforce parameters to be on the Grassmann manifold
`nn`	nearest neighbor parameter for `locfit.raw`
`nn.try`	vector of nearest neighbor parameters for `locfit.raw` to try in random initialization
`n.random`	integer number of random initializations for parameters to try
`optimize.nn`	should `nn` be optimized? Not recommended

`separate.nn`	should each subpopulation have its own `nn`? If `TRUE`, optimization takes much longer. It is rarely better, so recommended to set to `FALSE`
`constrain.none.subpop`	should the "none" subpopulation be constrained to be contained in every other subpopulation's dimension reduction subspace? Recommended to set to `TRUE`
`verbose`	should results be printed along the way?
`degree`	degree of kernel to use
`pooled`	should the estimator be a pooled estimator?
`maxk`	maxk parameter for `locfit.raw`. Set to a large number if an out of vertex space error occurs.
`...`	extra arguments passed to `locfit.raw`

Value

A list with the following elements

- beta a list of estimated sufficient dimension reduction matrices, one for each subpopulation
- beta.init a list of the initial sufficient dimension reduction matrices, one for each subpopulation – do not use, just for the sake of comparisons
- directions a list of estimated sufficient dimension reduction directions (i.e. the reduced dimension predictors/variables), one for each subpopulation. These have number of rows equal to the sample size for the subpopulation and number of columns equal to the specified dimensions of the reduced dimension spaces.
- y.list a list of vectors of responses for each subpopulation
- z.combinations the `z.combinations` specified as an input
- cov list of variance covariance matrices for the covariates for each subpopulation
- sqrt.inv.cov list of inverse square roots of the variance covariance matrices for the covariates for each subpopulation. These are used for scaling
- solver.obj object returned by the solver/optimization function
- value value of the objective function at the solution
- value.init value of the objective function at the initial beta (`beta.init`) used
- vic.est.eqn the average (unpenalized) VIC value across the r different input values. This assesses model fit
- vic.eqns the individual (unpenalized) VIC values across the r input values. Not used.
- vic the penalized VIC value. This is used for dimension selection, with dimensions chosen by the set of dimensions that minimize this penalized vic value that trades off model complexity and model fit

Examples

```
library(hierSDR)

set.seed(123)
dat <- simulate_data(nobs = 200, nvars = 6,
                     x.type = "some_categorical",
                     sd.y = 1, model = 2)

x <- dat$x ## covariates
z <- dat$z ## factor indicators
y <- dat$y ## response

dat$beta ## true coefficients that generate the subspaces

dat$z.combinations ## what combinations of z represent different subpops

## correct structural dimensions:
dat$d.correct

## fit hier SPHD model:


hiermod <- hier.sphd(x, y, z, dat$z.combinations, d = dat$d.correct,
                     verbose = FALSE, maxit = 250, maxk = 8200)

## validated inf criterion for choosing dimensions (the smaller the better)
hiermod$vic


cbind(hiermod$beta[[4]], NA, dat$beta[[4]])

## angles between estimated and true subspaces for each population:
mapply(function(x,y) angle(x,y), hiermod$beta, dat$beta)

## projection difference norm between estimated and true subspaces for each population:
mapply(function(x,y) projnorm(x,y), hiermod$beta, dat$beta)
```

phd — *PHD SDR fitting function*

Description

fits SDR models (PHD approach)

Usage

```
phd(x, y, d = 5L)
```

Arguments

x	an n x p matrix of covariates, where each row is an observation and each column is a predictor

`y`	vector of responses of length n
`d`	an integer representing the structural dimension

Value

A list with the following elements

- beta.hat estimated sufficient dimension reduction matrix
- eta.hat coefficients on the scale of the scaled covariates
- cov variance covariance matric for the covariates
- sqrt.inv.cov inverse square root of the variance covariance matrix for the covariates. Used for scaling
- M matrix from principal Hessian directions
- eigenvalues eigenvalues of the M matrix

`plot.hier_sdr_fit` *Plotting hierarchical SDR models*

Description

Plots hier.sdr objects

Usage

```
## S3 method for class 'hier_sdr_fit'
plot(x, ...)
```

Arguments

`x`	fitted object returned by `hier.sphd`
`...`	not used

Value

No return value, called for side effects

See Also

`hier.sphd` for function which fits hierarchical SDR model

Examples

```
library(hierSDR)
```

projnorm *Norm of difference of projections*

Description

Measures distance between two subspaces

Usage

```
projnorm(B1, B2)
```

Arguments

B1	first matrix
B2	second matrix

Value

scalar value of the projection difference norm between B1 and B2

Examples

```
b1 <- matrix(rnorm(10 * 2), ncol = 2)
b2 <- matrix(rnorm(10 * 2), ncol = 2)
projnorm(b1, b2)

## angle here should be smalls
b1 <- matrix(rnorm(10 * 2), ncol = 2)
b2 <- b1 + matrix(rnorm(10 * 2, sd = 0.2), ncol = 2)
projnorm(b1, b2)
```

semi.phd *Semiparametric PHD SDR fitting function*

Description

fits semiparametric SDR models (PHD approach)

Usage

```
semi.phd(
  x,
  y,
  d = 5L,
  maxit = 100L,
  h = NULL,
  opt.method = c("lbfgs.x", "bfgs", "lbfgs2", "bfgs.x", "lbfgs", "spg", "ucminf", "CG",
    "nlm", "nlminb", "newuoa"),
  nn = 0.95,
  init.method = c("random", "phd"),
  optimize.nn = FALSE,
  verbose = TRUE,
  n.samples = 100,
  degree = 2,
  vic = TRUE,
  ...
)
```

Arguments

`x`	an n x p matrix of covariates, where each row is an observation and each column is a predictor
`y`	vector of responses of length n
`d`	an integer representing the structural dimension
`maxit`	maximum number of iterations
`h`	bandwidth parameter. By default, a reasonable choice is selected automatically
`opt.method`	optimization method to use. Available choices are `c("lbfgs2", "lbfgs.x", "bfgs.x", "bfgs", "lbfgs", "spg", "ucminf", "CG", "nlm", "nlminb", "newuoa")`
`nn`	nearest neighbor parameter for `locfit.raw`
`init.method`	method for parameter initialization. Either `"random"` for random initialization or `"phd"` for a principle Hessian directions initialization approach
`optimize.nn`	should `nn` be optimized? Not recommended
`verbose`	should results be printed along the way?
`n.samples`	number of samples for the random initialization method
`degree`	degree of kernel to use
`vic`	logical value of whether or not to compute the VIC criterion for dimension determination
`...`	extra arguments passed to `locfit.raw`

Value

A list with the following elements

- beta estimated sufficient dimension reduction matrix
- beta.init initial sufficient dimension reduction matrix – do not use, just for the sake of comparisons
- cov variance covariance matric for the covariates
- sqrt.inv.cov inverse square root of the variance covariance matrix for the covariates. Used for scaling
- solver.obj object returned by the solver/optimization function
- vic the penalized VIC value. This is used for dimension selection, with dimension chosen to minimize this penalized vic value that trades off model complexity and model fit

`simulate_data` *Simulate data with hierarchical subspaces*

Description

Simulates data with hierarchical subspaces. Data are generated with two factors that induce heterogeneity

Usage

```
simulate_data(
  nobs,
  nvars,
  x.type = c("continuous", "some_categorical"),
  sd.y = 1,
  rho = 0.5,
  model = c("1", "2", "3")
)
```

Arguments

`nobs`	positive integer for the sample size per subpopulation
`nvars`	positive integer for the dimension
`x.type`	variable type for covariates, either `"continuous"` (where the covariates are multivariate normal with a variance-matrix with AR-1 form with parameter `rho`) or `"some_categorical"` (where half covariates are continuous and the other half are binary with dependencies on the continuous covariates)
`sd.y`	standard deviation of responsee
`rho`	correlation parameter for AR-1 covariance structure for continuous covariates
`model`	model number used, either "1", "2", or "3", each corresponds to a different outcome model setting

Value

A list with the following elements

- x a matrix of covariates with number of rows equal to the total sample size and columns equal to the number of variables
- z a matrix with number of rows equal to the total sample size and columns as dummy variables indicating presence of a stratifying factor
- y a vector of all responses
- beta a list of the true sufficient dimension reduction matrices, one for each subpopulation
- z.combinations all possible combinations of the stratifying factors `z`
- snr scalar the observed signal-to-noise ratio for the response
- d.correct the true dimensions of the dimension reduction spaces

Examples

```
library(hierSDR)

set.seed(123)
dat <- simulate_data(nobs = 100, nvars = 6,
                     x.type = "some_categorical",
                     sd.y = 1, model = 2)

x <- dat$x ## covariates
z <- dat$z ## factor indicators
y <- dat$y ## response

dat$beta ## true coefficients that generate the subspaces

dat$snr ## signal-to-noise ratio

str(x)
str(z)

dat$z.combinations ## what combinations of z represent different subpops

## correct structural dimensions:
dat$d.correct
```

Index