Package ‘desla’

November 11, 2021

Type Package

Title Desparsified Lasso

Version 0.1.0

Description Calculates the desparsified lasso as originally introduced in van de Geer et al. (2014) <doi:10.1214/14-AOS1221>, and provides inference suitable for high-dimensional time series, based on the long run covariance estimator in Adamek et al. (2020) <arXiv:2007.10952>.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.1.2

LinkingTo Rcpp, RcppArmadillo, RcppProgress

Imports Rcpp, Rdpack

RdMacros Rdpack

NeedsCompilation yes

Author Robert Adamek [cre, aut],
Stephan Smeekes [aut],
Ines Wilms [aut]

Maintainer Robert Adamek <robertadamek94@gmail.com>

Repository CRAN

Date/Publication 2021-11-11 09:00:08 UTC

R topics documented:

desla *Desparsified lasso*

Description

Calculates the desparsified lasso as originally introduced in van de Geer et al. (2014), and provides inference suitable for high-dimensional time series, based on the long run covariance estimator in Adamek et al. (2021).

Usage

```
desla(
  X,
  y,
  H,
  init_partial = NA,
  nw_partials = NA,
  demean = TRUE,
  scale = TRUE,
  gridsize = 100,
  init_grid = NA,
  nw_grids = NA,
  init_selection_type = NA,
  nw_selection_types = NA,
  init_nonzero_limit = NA,
  nw_nonzero_limits = NA,
  init_opt_threshold = NA,
  nw_opt_thresholds = NA,
  init_opt_type = NA,
  nw_opt_types = NA,
  LRVtrunc = 0,
  T_multiplier = 0,
  alphas = c(0.01, 0.05, 0.1),
  R = NA,
  q = NA,
  PIconstant = 0.8,
  PIprobability = 0.05,
  manual_Thetahat_ = NULL,
  manual_Upsilonhat_inv_ = NULL,
  manual_nw_residuals_ = NULL
)
```

Arguments

X	T_xN regressor matrix
y	T_x1 dependent variable vector

H
: indexes of relevant regressors

init_partial
: (optional) boolean, true if you want the initial lasso to be partially penalized (false by default)

nw_partials
: (optional) boolean vector with the dimension of H, trues if you want the nodewise regressions to be partially penalized (all false by default)

demean
: (optional) boolean, true if X and y should be demeaned before the desparsified lasso is calculated. This is recommended, due to the assumptions for the method (true by default)

scale
: (optional) boolean, true if X and y should be scaled by the column-wise standard deviations. Recommended for lasso based methods in general, since the penalty is scale-sensitive (true by default)

gridsize
: (optional) integer, how many different lambdas there should be in both initial and nodewise grids (100 by default)

init_grid
: (optional) vector, containing user specified initial grid

nw_grids
: (optional) matrix with number of rows the size of H, rows containing user specified grids for the nodewise regressions

init_selection_type
: (optional) integer, how should lambda be selected in the initial regression, 1=BIC, 2=AIC, 3=EBIC, 4=PI (4 by default)

nw_selection_types
: (optional) integer vector with the dimension of H, how should lambda be selected in the nodewise regressions, 1=BIC, 2=AIC, 3=EBIC, 4=PI (all 4s by default)

init_nonzero_limit
: (optional) number controlling the maximum number of nonzeros that can be selected in the initial regression (0.5 by default, meaning no more than 0.5*T_ regressors can have nonzero estimates)

nw_nonzero_limits
: (optional) vector with the dimension of H, controlling the maximum number of nonzeros that can be selected in the nodewise regressions (0.5s by default)

init_opt_threshold
: (optional) optimization threshold for the coordinate descent algorithm in the initial regression (10^(-4) by default)

nw_opt_thresholds
: (optional) vector with the dimension of H, optimization thresholds for the coordinate descent algorithm in the nodewise lasso regression (10^(-4)s by default)

init_opt_type
: (optional) integer, which type of coordinate descent algorithm should be used in the initial regression, 1=naive, 2=covariance, 3=adaptive (3 by default)

nw_opt_types
: (optional)integer vector with the dimension of H, which type of coordinate descent algorithm should be used in the nodewise regressions, 1=naive, 2=covariance, 3=adaptive (3s by default)

LRVtrunc
: (optional) parameter controlling the bandwidth Q_T used in the long run covariance matrix, Q_T=ceil(T_multiplier*T_^LRVtrunc). When LRVtrunc=T_multiplier=0, the bandwidth is selected according to Andrews (1991) (LRVtrunc=0 by default)

T_multiplier
: (optional) parameter controlling the bandwidth Q_T used in the long run covariance matrix, Q_T=ceil(T_multiplier*T_^LRVtrunc). When LRVtrunc=T_multiplier=0, the bandwidth is selected according to Andrews (1991) (T_multiplier=0 by default)

alphas
: (optional) vector of significance levels (c(0.01,0.05,0.1) by default)

R
: (optional) matrix with number of columns the dimension of H, used to test the null hypothesis R*beta=q (identity matrix as default)

q
: (optional) vector of size same as the rows of H, used to test the null hypothesis R*beta=q (zeroes by default)

PIconstant
: (optional) constant, used in the plug-in selection method (0.8 by default). For details see Adamek et al. (2021)

PIprobability
: (optional) probability, used in the plug-in selection method (0.05 by default). For details see Adamek et al. (2021)

manual_Thetahat_
: (optional) matrix with rows the size of H and columns the number of regressors. Can be obtained from earlier executions of the function to avoid unnecessary calculations of the nodewise regressions (NULL as default)

manual_Upsilonhat_inv_
: (optional) matrix with rows and columns the size of H. Can be obtained from earlier executions of the function to avoid unnecessary calculations of the nodewise regressions (NULL as default)

manual_nw_residuals_
: (optional) matrix with rows equal to the sample size and columns the size of H, containing the residuals from the nodewise regressions. Can be obtained from earlier executions of the function to avoid unnecessary calculations of the nodewise regressions (NULL as default)

Value

Returns a list with the following elements:

bhat_scaled
: desparsified lasso estimates for the parameters indexed by H. These estimates are based on data that is potentially standardized, for estimates that are brought back into the original scale of X, see bhat

bhat
: desparsified lasso estimates for the parameters indexed by H, unscaled to be in the original scale of y and X

intervals_scaled
: matrix containing the confidence intervals for parameters indexed in H, for significance levels given in alphas. These are based on data that is potentially standardized, for estimates that are brought back into the original scale of X, see intervals

intervals
: matrix containing the confidence intervals for parameters indexed in H,unscaled to be in the original scale of y and X

joint_chi2_stat
: test statistic for hull hypothesis R*beta=q, asymptotically chi squared distributed

chi2_critical_values
: critical values of the chi squared distribution with degrees of freedom corresponding to the joint test R*beta=q, for significance levels given in alphas

betahat
: lasso estimates from the initial regression of y on X

Gammahat
: matrix used for calculating the desparsified lasso, for details see Adamek et al. (2021)

Upsilonhat_inv
: matrix used for calculating the desparsified lasso, for details see Adamek et al. (2021)

Thetahat
: approximate inverse of (X'X)/T_, used for calculating the desparsified lasso, for details see Adamek et al. (2021)

Omegahat
: long run covariance matrix for the variables indexed by H, for details see Adamek et al. (2021)

init_residual
: vector of residuals from the initial lasso regression

nw_residuals
: matrix of residuals from the nodewise regressions

init_grid
: redundant output, returning the function input init_grid

nw_grids
: redundant output, returning the function input nw_grids

init_lambda
: value of lambda that was selected in the initial lasso regression

nw_lambdas
: values of lambdas that were selected in the nodewise lasso regressions

init_nonzero
: number on nonzero parameters in the initial lasso regression

nw_nonzeros
: vector of nonzero parameters in the nodewise lasso regressions

init_nonzero_pos
: vector of indexes of the nonzero parameters in the initial lasso

nw_nonzero_poss
: list of vectors for each nodewise regression, giving the indexes of nonzero parameters in the nodewise regressions

References

Adamek R, Smeekes S, Wilms I (2021). "LASSO inference for high-dimensional time series." *arXiv preprint arXiv:2007.10952*.

Andrews DW (1991). "Heteroskedasticity and autocorrelation consistent covariance matrix estimation." *Econometrica*, **59**(3), 817–858.

van de Geer S, Buhlmann P, Ritov Y, Dezeure R (2014). "On asymptotically optimal confidence regions and tests for high-dimensional models." *Annals of Statistics*, **42**(3), 1166–1202.

Examples

```
X<-matrix(rnorm(100*100), nrow=100)
y<-X[,1:4] %*% c(1, 2, 3, 4) + rnorm(100)
H<-c(1, 2, 3, 4)
d<-desla(X, y, H)
```

HDLP *High-Dimensional Local Projection*

Description

Calculates impulse responses with local projections, using the desla function to estimate the high-dimensional linear models, and provide asymptotic inference. The naming conventions in this function follow the notation in Plagborg-Moller and Wolf (2021), in particular Equation 1 therein.

Usage

```
HDLP(
  r = NULL,
  x,
  y,
  q = NULL,
  y_predetermined = FALSE,
  cumulate_y = FALSE,
  hmax = 24,
  lags = 12,
  alphas = 0.05,
  init_partial = TRUE,
  selection = 4,
  PIconstant = 0.8,
  progress_bar = TRUE
)
```

Arguments

`r`	(optional) vector or matrix with `T_` rows, containing the "slow" variables, ones which do not react within the same period to a shock, see Plagborg-Moller and Wolf (2021) for details(NULL by default)
`x`	`T_`x1 vector containing the shock variable, see Plagborg-Moller and Wolf (2021) for details
`y`	`T_`x1 vector containing the response variable, see Plagborg-Moller and Wolf (2021) for details
`q`	(optional) vector or matrix with `T_` rows, containing the "fast" variables, ones which may react within the same period to a shock, see Plagborg-Moller and Wolf (2021) for details (NULL by default)
`y_predetermined`	(optional) boolean, true if the response variable `y` is predetermined with respect to `x`, i.e. cannot react within the same period to the shock. If true, the impulse response at horizon 0 is 0 (false by default)
`cumulate_y`	(optional) boolean, true if the impulse response of `y` should be cumulated, i.e. using the cumulative sum of `y` as the dependent variable (false by default)

hmax	(optional) integer, the maximum horizon up to which the impulse responses are computed. Should not exceed the `T_-lags` (24 by default)
lags	(optional) integer, the number of lags to be included in the local projection model. Should not exceed `T_-hmax`(12 by default)
alphas	(optional) vector of significance levels (0.05 by default)
init_partial	(optional) bool, true if the parameter of interest should NOT be penalized (true by default)
selection	(optional) integer, how should lambda be selected in BOTH the initial and nodewise regressions, 1=BIC, 2=AIC, 3=EBIC, 4=PI (4 by default)
PIconstant	(optional) constant, used in the plug-in selection method (0.8 by default). For details see Adamek et al. (2021)
progress_bar	(optional) boolean, true if a progress bar should be displayed during execution (true by default)

Value

Returns a list with the following elements:

intervals	matrix containing the point estimates and confidence intervals for the impulse response function, for significance levels given in `alphas`
Thetahat	matrix (row vector) calculated from the nodewise regression at horizon 0, which is re-used at later horizons

References

Adamek R, Smeekes S, Wilms I (2021). "LASSO inference for high-dimensional time series." *arXiv preprint arXiv:2007.10952*.

Plagborg-Moller M, Wolf CK (2021). "Local projections and VARs estimate the same impulse responses." *Econometrica*, **89**(2), 955–980.

Examples

```
X<-matrix(rnorm(100*100), nrow=100)
y<-X[,1:4] %*% c(1, 2, 3, 4) + rnorm(100)
h<-HDLP(x=X[,4], y=y, q=X[,-4], hmax=5, lags=1)
```

Index