# Package: desla (via r-universe)

September 28, 2024

**Type** Package

**Title** Desparsified Lasso Inference for Time Series

**Version** 0.3.0

**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>. Also estimates high-dimensional local projections by the desparsified lasso, as described in Adamek et al. (2022) <arXiv:2209.03218>.

**License** GPL (>= 2)

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**LinkingTo** Rcpp, RcppArmadillo, RcppProgress, sitmo

**Imports** Rcpp, Rdpack, stats, parallelly

**URL** https://github.com/RobertAdamek/desla

**BugReports** https://github.com/RobertAdamek/desla/issues

**Suggests** ggplot2

**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** 2023-06-29 11:50:06 UTC

# Contents



---

## `create_state_dummies` *Create State Dummies*

---

### Description

Creates state dummies for use in `HDLP`.

### Usage

```
create_state_dummies(x)
```

### Arguments

| | |
|---|---|
| `x` | Contains the variables that define the states. Each column should either represent a categorical variable indicating the state of each observation, or each column should be a binary indicator for one particular state. |

### Details

The function first checks if `x` is already in the correct output format by evaluating if each row sums up to one. If this is not the case, each column is treated as a categorical variable for which its unique entries define the states it can take. If `x` contains more than one column, interactions between the variables are created. Example, inputting two variables that can take two states each, results in a total of four possible states, and hence the output matrix contains four columns.

### Value

A matrix where each column is a binary indicator for one state.

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,
  alphas = 0.05,
  penalize_H = TRUE,
  R = NULL,
  q = NULL,
  demean = TRUE,
  scale = TRUE,
  progress_bar = TRUE,
  parallel = TRUE,
  threads = NULL,
  PI_constant = NULL,
  LRV_bandwidth = NULL
)
```

## Arguments

| | |
|---|---|
| `X` | `T_` x `N` regressor matrix |
| `y` | `T_` x 1 dependent variable vector |
| `H` | indexes of relevant regressors |
| `alphas` | (optional) vector of significance levels (0.05 by default) |
| `penalize_H` | (optional) boolean, true if you want the variables in H to be penalized (`TRUE` 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) |
| `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)

progress_bar (optional) boolean, displays a progress bar while running if true, tracking the progress of estimating the nodewise regressions (TRUE by default)

parallel boolean, whether parallel computing should be used (TRUE by default)

threads (optional) integer, how many threads should be used for parallel computing if parallel=TRUE (default is to use all but two)

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

LRV_bandwidth (optional) vector of parameters controlling the bandwidth Q_T used in the long run covariance matrix, Q_T=ceil(LRV_bandwidth[1]*T_^LRV_bandwidth[2]). When LRV_bandwidth=NULL, the bandwidth is selected according to Andrews (1991) (default)

## Value

Returns a list with the following elements:

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

standard_errors
standard errors of the estimates for variables indexed by H

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

betahat lasso estimates from the initial regression of y on X

DSL_matrices list containing the matrices Gammahat, Upsilonhat_inv and Thetahat used for calculating the desparsified lasso, as well as Omegahat, the long run covariance matrix for the variables indexed by H. For details see Adamek et al. (2021)

residuals list containing the vector of residuals from the initial lasso regression (init) and the matrix of residuals from the nodewise regressions (nw)

lambdas values of lambda selected in the initial lasso regression (init) and the nodewise lasso regressions (nw)

selected_vars vector of indexes of the nonzero parameters in the initial lasso (init) and each nodewise regression (nw)

wald_test list containing elements for inference on R beta=q. joint_test contains the test statistic for the overall null hypothesis R beta=q along with the p-value. At default values of R and q, this tests the joint significance of all variables indexed by H. row_tests contains the vector of z-statistics and confidence intervals associated with each row of R beta - q, unscaled to be in the original scale of y and X. This output is only given when either R or q are supplied

### 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(50*50), nrow=50)
y<-X[,1:4] %*% c(1, 2, 3, 4) + rnorm(50)
H<-c(1, 2, 3, 4)
d<-desla(X, y, H)
```

---

## HDLP *State Dependent 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. This function also allows for estimating state-dependent responses, as in Ramey and Zubairy (2018).

### Usage

```
HDLP(
  x,
  y,
  r = NULL,
  q = NULL,
  state_variables = NULL,
  y_predetermined = FALSE,
  cumulate_y = FALSE,
  hmax = 24,
  lags = 12,
  alphas = 0.05,
  penalize_x = FALSE,
  PI_constant = NULL,
  progress_bar = TRUE,
  OLS = FALSE,
  parallel = TRUE,
  threads = NULL
)
```

### Arguments

- `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
- `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)
- `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)
- `state_variables`: (optional) matrix or data frame with `T_` rows, containing the variables that define the states. Each column should either represent a categorical variable indicating the state of each observation, or each column should be a binary indicator for one particular state; see 'Details'.
- `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)
- `penalize_x`: (optional) boolean, true if the parameter of interest should be penalized (`FALSE` by default)
- `PI_constant`: (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)
- `OLS`: (optional) boolean, whether the local projections should be computed by OLS instead of the desparsified lasso. This should only be done for low-dimensional regressions (FALSE by default)
- `parallel`: boolean, whether parallel computing should be used. Default is TRUE.
- `threads`: (optional) integer, how many threads should be used for parallel computing if `parallel=TRUE`. Default is to use all but two.

### Details

The input to `state_variables` is transformed to a suitable matrix where each column represents one state using the function `create_state_dummies`. See that function for further details.

## Value

Returns a list with the following elements:

| | |
|---|---|
| `intervals` | list of matrices containing the point estimates and confidence intervals for the impulse response functions in each state, 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 |
| `betahats` | list of matrices (column vectors), giving the initial lasso estimate at each horizon |

## 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.

Ramey VA, Zubairy S (2018). "Government spending multipliers in good times and in bad: evidence from US historical data." *Journal of Political Economy*, **126**(2), 850–901.

## Examples

```
X<-matrix(rnorm(50*50), nrow=50)
y<-X[,1:4] %*% c(1, 2, 3, 4) + rnorm(50)
s<-matrix(c(rep(1,25),rep(0,50),rep(1,25)), ncol=2, dimnames = list(NULL, c("A","B")))
h<-HDLP(x=X[,4], y=y, q=X[,-4], state_variables=s, hmax=5, lags=1)
plot(h)
```

---

# plot.hdlp *Plot Impulse Responses obtained from HDLP.*

---

## Description

Plot Impulse Responses obtained from HDLP.

## Usage

```
## S3 method for class 'hdlp'
plot(
  x,
  y = NULL,
  response = NULL,
  impulse = NULL,
  states = NULL,
  units = NULL,
  title = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| x | Output of the HDLP() function. |
| y | Has no function, included for compatibility with plot.default(). |
| response | Name of the response variable (y in HDLP()). |
| impulse | Name of the shock variable (x in HDLP()). |
| states | Optional names of the states (when applicable). If not provided, names will be determined from x. |
| units | Units of the response variable (y-axis label). |
| title | String containing title of the plot; can be used to overwrite default generated based on the names of the response and impulse variables. |
| ... | Other arguments forwarded to plot function (currently inactive). |

## Value

A ggplot object.

# Index