Package 'FAS'

May 7, 2026

Type Package

Title Factor-Augmented Sparse Regression Tuning-Free Testing

Version 1.0.0

Maintainer Jonas Striaukas <jonas.striaukas@gmail.com>

Description The 'FAS' package implements the bootstrap method for the tuning parameter selection and tuning-free inference on sparse regression coefficient vectors. Currently, the test could be applied to linear and factor-augmented sparse regressions, see Lederer & Vogt (2021, JMLR) <https://www.jmlr.org/papers/volume22/20-539/20-539.pdf> and Beyhum & Striaukas (2023) <doi:10.48550/arXiv.2307.13364>.

License GPL (>= 2)

Depends pracma, Matrix, R (>= 3.5.0)

Imports stats, graphics, methods

RoxygenNote 7.2.3

NeedsCompilation yes

Author Jonas Striaukas [cre, aut],
Jad Beyhum [aut]

Repository CRAN

Date/Publication 2024-01-10 12:53:03 UTC

Contents

FAS-package 2
factorsparsetest 2
lassofit 3

Index **6**

FAS-package *FAS*

Description

Bootstrap methods for selecting the tuning parameter for LASSO-type regression models and testing sparse regression coefficients

Author(s)

Jonas Striaukas (maintainer) <jonas.striaukas@gmail.com>

factorsparsetest *Test of the factor model against factor augmented sparse alternative*

Description

Test of the factor model against factor augmented sparse alternative

Usage

```
factorsparsetest(x, y, w = NULL, q.levels = c(0.90, 0.95, 0.99),
                 p.value = FALSE, rmax = 10, ...)
```

Arguments

x	T by p data matrix, where T and p respectively denote the sample size and the number of regressors.
y	T by 1 response variable.
w	T BY k additional regressors added in to the factor model under H0.
q.levels	quantile levels of effective noise.
p.value	whether pvalue should be computed. Default is FALSE.
rmax	maximum number of factors. Use in eigenvalue ratio estimator. Default is 10.
...	other arguments that can be passed to lassofit.

Details

Computes the test statistic and the p-value for testing the factor model against factor augmented sparse alternative. The number of factors are estimated by eigenvalue ratio estimator.

Value

factorsparsetest object.

Author(s)

Jonas Striaukas

Examples

```
set.seed(1)
x = matrix(rnorm(100 * 20), 100, 20)
beta = c(5,4,3,2,1,rep(0, times = 15))
y = x%*%beta + rnorm(100)
factorsparsetest(x = x, y = y)
```

`lassofit` *Fits effective noise of LASSO regressions*

Description

Fits effective noise of LASSO regressions.

Usage

```
lassofit(x, y, q.levels = c(0.90, 0.95, 0.99), p.value = FALSE,
         numboot = 1000L, nlambda = 100L,
         lambda.factor = ifelse(nobs < nvars, 1e-02, 1e-04),
         lambda = NULL, pf = rep(1, nvars),
         dfmax = nvars + 1,
         pmax = min(dfmax * 1.2, nvars), standardize = FALSE,
         intercept = FALSE, eps = 1e-08, maxit = 1000000L)
```

Arguments

`x`	T by p data matrix, where T and p respectively denote the sample size and the number of regressors.
`y`	T by 1 response variable.
`q.levels`	quantile levels of effective noise.
`p.value`	whether pvalue should be computed. Default is `FALSE`.
`numboot`	bootstrap replications.
`nlambda`	number of λ's to use in the regularization path; used if `lambda = NULL`.
`lambda.factor`	The factor for getting the minimal λ in the λ sequence, where `min(lambda) = lambda.factor * max(lambda)`. max(lambda) is the smallest value of lambda for which all coefficients are zero. λ_{max} is determined for each γ tuning parameter separately. The default depends on the relationship between `T` (the sample size) and `p` (the number of predictors). If `T < p`, the default is `0.01`. If `T > p`, the default is `0.0001`, closer to zero. The smaller the value of `lambda.factor` is, the denser is the fit for λ_{min}. Used only if `lambda = NULL`.

`lambda` a user-supplied lambda sequence. By leaving this option unspecified (recommended), users can have the program compute its own `lambda` sequence based on `nlambda` and `lambda.factor`. It is better to supply, if necessary, a decreasing sequence of lambda values than a single (small) value, as warm-starts are used in the optimization algorithm. The program will ensure that the user-supplied λ sequence is sorted in decreasing order before fitting the model.

`pf` the ℓ_1 penalty factor of length `p` used for the adaptive sg-LASSO. Separate ℓ_1 penalty weights can be applied to each coefficient to allow different ℓ_1 + $\ell_{2,1}$ shrinkage. Can be 0 for some variables, which imposes no shrinkage, and results in that variable always be included in the model. Default is 1 for all variables.

`dfmax` the maximum number of variables allowed in the model. Useful for very large `p` when a partial path is desired. Default is `p+1`. In case `method='fe'`, `dfmax` is ignored.

`pmax` the maximum number of coefficients allowed ever to be nonzero. For example, once $\beta_i \neq 0$ for some $i \in [p]$, no matter how many times it exits or re-enters the model through the path, it will be counted only once. Default is `min(dfmax*1.2, p)`.

`standardize` logical flag for variable standardization, prior to fitting the model sequence. The coefficients are always returned to the original scale. It is recommended to keep `standardize=TRUE`. Default is `FALSE`.

`intercept` whether intercept be fitted (`TRUE`) or set to zero (`FALSE`). Default is `FALSE`. In case `method='pooled'`, `intercept=TRUE` is forced. In case `method='fe'`, `intercept=FALSE` is forced and `entity` specific intercepts are fitted in a separate output variable `a0`.

`eps` convergence threshold for block coordinate descent. Each inner block coordinate-descent loop continues until the maximum change in the objective after any coefficient update is less than thresh times the null deviance. Defaults value is `1e-8`.

`maxit` maximum number of outer-loop iterations allowed at fixed lambda values. Default is `1e6`. If the algorithm does not converge, consider increasing `maxit`.

Details

Fits effective noise of LASSO regressions.

Value

lassofit object.

Author(s)

Jonas Striaukas

Examples

```
set.seed(1)
x = matrix(rnorm(100 * 20), 100, 20)
```

```
beta = c(5,4,3,2,1,rep(0, times = 15))
y = x%*%beta + rnorm(100)
lassofit(x = x, y = y)
```

Index

∗ **package**
FAS-package, 2

factorsparsetest, 2
FAS *(*FAS-package*)*, 2
FAS-package, 2

lassofit, *2*, 3