Package 'natural'

January 16, 2018

Type Package

Title Estimating the Error Variance in a High-Dimensional Linear Model

Version 0.9.0

Maintainer Guo Yu <gy63@cornell.edu>

Description Implementation of the two error variance estimation methods in high-dimensional linear models of Yu, Bien (2017) <arXiv:1712.02412>.

URL https://arxiv.org/abs/1712.02412

BugReports https://github.com/hugogogo/natural/issues

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Imports Matrix, glmnet

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Author Guo Yu [aut, cre]

Repository CRAN

Date/Publication 2018-01-16 10:35:43 UTC

R topics documented:

`getLam_olasso` *Get the two (theoretical) values of lambdas used in the organic lasso*

Description

Get the two (theoretical) values of lambdas used in the organic lasso

Usage

```
getLam_olasso(x)
```

Arguments

x	design matrix

`getLam_slasso` *Get the two (theoretical) values of lambdas used in scaled lasso*

Description

Get the two (theoretical) values of lambdas used in scaled lasso

Usage

```
getLam_slasso(n, p)
```

Arguments

n	number of observations
p	number of features

`make_sparse_model` *Generate sparse linear model and random samples*

Description

Generate design matrix and response following linear models $y = X\beta + \epsilon$, where ϵ $N(0, \sigma^2)$, and X $N(0, \Sigma)$.

Usage

```
make_sparse_model(n, p, alpha, rho, snr, nsim)
```

Arguments

`n`	the sample size
`p`	the number of features
`alpha`	sparsity, i.e., n^α nonzeros in the true regression coefficient.
`rho`	pairwise correlation among features
`snr`	signal to noise ratio, defined as $\beta^T\Sigma\beta/\sigma^2$
`nsim`	the number of simulations

Value

A list object containing:

`x`: The `n` by `p` design matrix

`y`: The `n` by `nsim` matrix of response vector, each column representing one replication of the simulation

`beta`: The true regression coefficient vector

`sigma`: The true error standard deviation

`natural` *natural: Natural and Organic lasso estimates of error variance in high-dimensional linear models*

Description

The package contains implementation of the two methods introduced in Yu, Bien (2017) `https://arxiv.org/abs/1712.02412`.

Details

The main functions are `nlasso_cv`, `olasso_cv`, and `olasso`.

nlasso_cv *Cross-validation for natural lasso*

Description

Provide natural lasso estimate (of the error standard deviation) using cross-validation to select the tuning parameter value The output also includes the cross-validation result of the naive estimate and the degree of freedom adjusted estimate of the error standard deviation.

Usage

```
nlasso_cv(x, y, lambda = NULL, intercept = TRUE, nlam = 100,
  flmin = 0.01, nfold = 5, foldid = NULL, thresh = 1e-08,
  glmnet_output = NULL)
```

Arguments

`x` An `n` by `p` design matrix. Each row is an observation of `p` features.

`y` A response vector of size `n`.

`lambda` A user specified list of tuning parameter. Default to be NULL, and the program will compute its own `lambda` path based on `nlam` and `flmin`.

`intercept` Indicator of whether intercept should be fitted. Default to be `TRUE`.

`nlam` The number of `lambda` values. Default value is `100`.

`flmin` The ratio of the smallest and the largest values in `lambda`. The largest value in `lambda` is usually the smallest value for which all coefficients are set to zero. Default to be `1e-2`.

`nfold` Number of folds in cross-validation. Default value is 5. If each fold gets too view observation, a warning is thrown and the minimal `nfold = 3` is used.

`foldid` A vector of length `n` representing which fold each observation belongs to. Default to be `NULL`, and the program will generate its own randomly.

`thresh` Threshold value for underlying optimization algorithm to claim convergence. Default to be `1e-8`.

`glmnet_output` Should the estimate be computed using a user-specified output from `cv.glmnet`. If not `NULL`, it should be the output from `cv.glmnet` call with `standardize = TRUE` and `keep = TRUE`, and then the arguments `lambda`, `intercept`, `nlam`, `flmin`, `nfold`, `foldid`, and `thresh` will be ignored. Default to be `NULL`, in which case the function will call `cv.glmnet` internally.

Value

A list object containing:

`n` **and** `p`**:** The dimension of the problem.

`lambda`**:** The path of tuning parameter used.

- `beta`: Estimate of the regression coefficients, in the original scale, corresponding to the tuning parameter selected by cross-validation.
- `a0`: Estimate of intercept
- `mat_mse`: The estimated prediction error on the test sets in cross-validation. A matrix of size `nlam` by `nfold`. If `glmnet_output` is not `NULL`, then `mat_mse` will be NULL.
- `cvm`: The averaged estimated prediction error on the test sets over K folds.
- `cvse`: The standard error of the estimated prediction error on the test sets over K folds.
- `ibest`: The index in `lambda` that attains the minimal mean cross-validated error.
- `foldid`: Fold assignment. A vector of length `n`.
- `nfold`: The number of folds used in cross-validation.
- `sig_obj`: Natural lasso estimate of standard deviation of the error, with the optimal tuning parameter selected by cross-validation.
- `sig_obj_path`: Natural lasso estimates of standard deviation of the error. A vector of length `nlam`.
- `sig_naive`: Naive estimates of the error standard deviation based on lasso regression, i.e., $||y - X\hat{\beta}||_2/\sqrt{n}$, selected by cross-validation.
- `sig_naive_path`: Naive estimate of standard deviation of the error based on lasso regression. A vector of length `nlam`.
- `sig_df`: Degree-of-freedom adjusted estimate of standard deviation of the error, selected by cross-validation. See Reid, et, al (2016).
- `sig_df_path`: Degree-of-freedom adjusted estimate of standard deviation of the error. A vector of length `nlam`.
- `type`: whether the output is of a natural or an organic lasso.

See Also

`nlasso_path`

Examples

```
set.seed(123)
sim <- make_sparse_model(n = 50, p = 200, alpha = 0.6, rho = 0.6, snr = 2, nsim = 1)
nl_cv <- nlasso_cv(x = sim$x, y = sim$y[, 1])
```

`nlasso_path` *Fit a linear model with natural lasso*

Description

Calculate a solution path of the natural lasso estimate (of error standard deviation) with a list of tuning parameter values. In particular, this function solves the lasso problems and returns the lasso objective function values as estimates of the error variance: $\hat{\sigma}^2_\lambda = \min_\beta ||y - X\beta||_2^2/n + 2\lambda||\beta||_1$. The output also includes a path of naive estimates and a path of degree of freedom adjusted estimates of the error standard deviation.

Usage

```
nlasso_path(x, y, lambda = NULL, nlam = 100, flmin = 0.01,
  thresh = 1e-08, intercept = TRUE, glmnet_output = NULL)
```

Arguments

x — An n by p design matrix. Each row is an observation of p features.

y — A response vector of size n.

lambda — A user specified list of tuning parameter. Default to be NULL, and the program will compute its own lambda path based on nlam and flmin.

nlam — The number of lambda values. Default value is 100.

flmin — The ratio of the smallest and the largest values in lambda. The largest value in lambda is usually the smallest value for which all coefficients are set to zero. Default to be 1e-2.

thresh — Threshold value for the underlying optimization algorithm to claim convergence. Default to be 1e-8.

intercept — Indicator of whether intercept should be fitted. Default to be TRUE.

glmnet_output — Should the estimate be computed using a user-specified output from glmnet. If not NULL, it should be the output from glmnet call with standardize = TRUE, and then the arguments lambda, nlam, flmin, thresh, and intercept will be ignored. Default to be NULL, in which case the function will call glmnet internally.

Value

A list object containing:

n and p: The dimension of the problem.

lambda: The path of tuning parameters used.

beta: Matrix of estimates of the regression coefficients, in the original scale. The matrix is of size p by nlam. The j-th column represents the estimate of coefficient corresponding to the j-th tuning parameter in lambda.

a0: Estimate of intercept. A vector of length nlam.

sig_obj_path: Natural lasso estimates of the error standard deviation. A vector of length nlam.

sig_naive_path: Naive estimates of the error standard deviation based on lasso regression, i.e., $||y - X\hat{\beta}||_2/\sqrt{n}$. A vector of length nlam.

sig_df_path: Degree-of-freedom adjusted estimate of standard deviation of the error. A vector of length nlam. See Reid, et, al (2016).

type: whether the output is of a natural or an organic lasso.

See Also

nlasso_cv

Examples

```
set.seed(123)
sim <- make_sparse_model(n = 50, p = 200, alpha = 0.6, rho = 0.6, snr = 2, nsim = 1)
nl_path <- nlasso_path(x = sim$x, y = sim$y[, 1])
```

`olasso` *Error standard deviation estimation using organic lasso*

Description

Solve the organic lasso problem $\tilde{\sigma}^2_\lambda = \min_\beta ||y - X\beta||_2^2/n + 2\lambda||\beta||_1^2$ with two pre-specified values of tuning parameter: $\lambda_1 = logp/n$, and λ_2, which is a Monte-Carlo estimate of $||X^Te||_\infty^2/n^2$, where e is n-dimensional standard normal.

Usage

```
olasso(x, y, intercept = TRUE, thresh = 1e-08)
```

Arguments

`x` An `n` by `p` design matrix. Each row is an observation of `p` features.

`y` A response vector of size `n`.

`intercept` Indicator of whether intercept should be fitted. Default to be `TRUE`.

`thresh` Threshold value for underlying optimization algorithm to claim convergence. Default to be `1e-8`.

Value

A list object containing:

`n` and `p`: The dimension of the problem.

`lam_1`, `lam_2`: $log(p)/n$, and an Monte-Carlo estimate of $||X^Te||_\infty^2/n^2$, where e is n-dimensional standard normal.

`a0_1`, `a0_2`: Estimate of intercept, corresponding to `lam_1` and `lam_2`.

`beta_1`, `beta_2`: Organic lasso estimate of regression coefficients, corresponding to `lam_1` and `lam_2`.

`sig_obj_1`, `sig_obj_2`: Organic lasso estimate of the error standard deviation, corresponding to `lam_1` and `lam_2`.

See Also

`olasso_path`, `olasso_cv`

Examples

```
set.seed(123)
sim <- make_sparse_model(n = 50, p = 200, alpha = 0.6, rho = 0.6, snr = 2, nsim = 1)
ol <- olasso(x = sim$x, y = sim$y[, 1])
```

olasso_cv *Cross-validation for organic lasso*

Description

Provide organic lasso estimate (of the error standard deviation) using cross-validation to select the tuning parameter value

Usage

```
olasso_cv(x, y, lambda = NULL, intercept = TRUE, nlam = 100,
  flmin = 0.01, nfold = 5, foldid = NULL, thresh = 1e-08)
```

Arguments

x	An n by p design matrix. Each row is an observation of p features.
y	A response vector of size n.
lambda	A user specified list of tuning parameter. Default to be NULL, and the program will compute its own lambda path based on nlam and flmin.
intercept	Indicator of whether intercept should be fitted. Default to be TRUE.
nlam	The number of lambda values. Default value is 100.
flmin	The ratio of the smallest and the largest values in lambda. The largest value in lambda is usually the smallest value for which all coefficients are set to zero. Default to be 1e-2.
nfold	Number of folds in cross-validation. Default value is 5. If each fold gets too view observation, a warning is thrown and the minimal nfold = 3 is used.
foldid	A vector of length n representing which fold each observation belongs to. Default to be NULL, and the program will generate its own randomly.
thresh	Threshold value for underlying optimization algorithm to claim convergence. Default to be 1e-8.

Value

A list object containing:

n and p: The dimension of the problem.

lambda: The path of tuning parameter used.

beta: Estimate of the regression coefficients, in the original scale, corresponding to the tuning parameter selected by cross-validation.

a0: Estimate of intercept

mat_mse: The estimated prediction error on the test sets in cross-validation. A matrix of size nlam by nfold

cvm: The averaged estimated prediction error on the test sets over K folds.

`cvse`**:** The standard error of the estimated prediction error on the test sets over K folds.

`ibest`**:** The index in `lambda` that attains the minimal mean cross-validated error.

`foldid`**:** Fold assignment. A vector of length `n`.

`nfold`**:** The number of folds used in cross-validation.

`sig_obj`**:** Organic lasso estimate of the error standard deviation, selected by cross-validation.

`sig_obj_path`**:** Organic lasso estimates of the error standard deviation. A vector of length `nlam`.

`type`**:** whether the output is of a natural or an organic lasso.

See Also

`olasso_path`, `olasso`

Examples

```
set.seed(123)
sim <- make_sparse_model(n = 50, p = 200, alpha = 0.6, rho = 0.6, snr = 2, nsim = 1)
ol_cv <- olasso_cv(x = sim$x, y = sim$y[, 1])
```

`olasso_path` *Fit a linear model with organic lasso*

Description

Calculate a solution path of the organic lasso estimate (of error standard deviation) with a list of tuning parameter values. In particular, this function solves the squared-lasso problems and returns the objective function values as estimates of the error variance: $\tilde{\sigma}^2_\lambda = \min_\beta ||y - X\beta||_2^2/n + 2\lambda||\beta||_1^2$.

Usage

```
olasso_path(x, y, lambda = NULL, nlam = 100, flmin = 0.01,
  thresh = 1e-08, intercept = TRUE)
```

Arguments

`x` An `n` by `p` design matrix. Each row is an observation of `p` features.

`y` A response vector of size `n`.

`lambda` A user specified list of tuning parameter. Default to be NULL, and the program will compute its own `lambda` path based on `nlam` and `flmin`.

`nlam` The number of `lambda` values. Default value is `100`.

`flmin` The ratio of the smallest and the largest values in `lambda`. The largest value in `lambda` is usually the smallest value for which all coefficients are set to zero. Default to be `1e-2`.

`thresh` Threshold value for underlying optimization algorithm to claim convergence. Default to be `1e-8`.

`intercept` Indicator of whether intercept should be fitted. Default to be `FALSE`.

Details

This package also includes the outputs of the naive and the degree-of-freedom adjusted estimates, in analogy to nlasso_path.

Value

A list object containing:

- `n` **and** `p`**:** The dimension of the problem.
- `lambda`**:** The path of tuning parameter used.
- `a0`**:** Estimate of intercept. A vector of length `nlam`.
- `beta`**:** Matrix of estimates of the regression coefficients, in the original scale. The matrix is of size `p` by `nlam`. The `j`-th column represents the estimate of coefficient corresponding to the `j`-th tuning parameter in `lambda`.
- `sig_obj_path`**:** Organic lasso estimates of the error standard deviation. A vector of length `nlam`.
- `sig_naive`**:** Naive estimate of the error standard deviation based on the squared-lasso regression. A vector of length `nlam`.
- `sig_df`**:** Degree-of-freedom adjusted estimate of the error standard deviation, based on the squared-lasso regression. A vector of length `nlam`.
- `type`**:** whether the output is of a natural or an organic lasso.

See Also

olasso, olasso_cv

Examples

```
set.seed(123)
sim <- make_sparse_model(n = 50, p = 200, alpha = 0.6, rho = 0.6, snr = 2, nsim = 1)
ol_path <- olasso_path(x = sim$x, y = sim$y[, 1])
```

---

olasso_slow *Solve organic lasso problem with a single value of lambda The lambda values are for slow rates, which could give less satisfying results*

Description

Solve organic lasso problem with a single value of lambda The lambda values are for slow rates, which could give less satisfying results

Usage

```
olasso_slow(x, y, thresh = 1e-08)
```

Arguments

`x` An `n` by `p` design matrix. Each row is an observation of `p` features.

`y` A response vector of size `n`.

`thresh` Threshold value for underlying optimization algorithm to claim convergence. Default to be `1e-8`.

`plot.natural.cv` *plot a natural.cv object*

Description

This function is adapted from the **ggb** R package.

Usage

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

Arguments

`x` an object of class `natural.cv`, as returned by `nlasso_cv` and `olasso_cv`

`...` additional argument(not used here, only for S3 generic/method consistency)

`plot.natural.path` *plot a natural.path object*

Description

This function is adapted from the **ggb** R package.

Usage

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

Arguments

`x` an object of class `natural.path`, as returned by `nlasso_path` and `olasso_path`

`...` additional argument(not used here, only for S3 generic/method consistency)

print.natural.path *print a natural.path object*

Description

This function is adapted from the **ggb** R package.

Usage

```
## S3 method for class 'natural.path'
print(x, ...)
```

Arguments

x	an object of class natural.path, as returned by nlasso_path and olasso_path
...	additional argument(not used here, only for S3 generic/method consistency)

standardize *Standardize the n -by- p design matrix X to have column means zero and ||X_j||_2^2 = n for all j*

Description

Standardize the n -by- p design matrix X to have column means zero and ||X_j||_2^2 = n for all j

Usage

```
standardize(x, center = TRUE)
```

Arguments

x	design matrix
center	should we set column means equal to zero

Index