

Package ‘Slab’

August 19, 2024

Type Package

Title Slab and Shrinkage Generalised LASSO Estimators

Version 0.1.0

Maintainer Hoi Chun Chen <jessechen0228@outlook.com>

Description This package aim to provide better estimations that have lower mean square errors (MSE) than ordinary least square (OLS) when handling high dimension data.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Imports maotai, MASS, mnormt, glmnet, ggplot2

Suggests knitr,
rmarkdown,
testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

Author Hoi Chun Chen [aut, cre],
Vali Asimit [aut],
Feng Zhou [aut]

VignetteBuilder knitr

R topics documented:

plot.cv_model	2
plot.model	2
slab_shrink	4

plot.cv_model

Plot Cross-Validation Coefficients

Description

This function plots the cross-validation error against the log of lambda values for a given glmnet model. It visualizes the performance of the model across different levels of regularization.

Usage

```
## S3 method for class 'cv_model'
plot(result_model)
```

Arguments

result_model A model object returned by cv.glmnet containing cross-validation results. This object should include cvm (cross-validation error mean), cvsd (cross-validation error standard deviation), lambda_range (range of lambda values), and lambda (optimal lambda value).

Value

This function does not return a value. It generates a plot.

Examples

```
# Same data generate process as slab_shrink
set.seed(123)
n <- 100 # Number of observations
p <- 50 # Number of variables
test_x <- matrix(rnorm(n * p), n, p)
beta <- matrix(rnorm(p + 1), p + 1, 1)
test_y <- cbind(1, test_x) %*% beta + rnorm(n, sd = 0.5)

# Fit a model using a fictional slab_shrink function
result <- slab_shrink(test_x, test_y) # Estimation works with just x and y

# Plot the model using default settings
plot.cv_model(result)
```

plot.model

Plot Model Coefficients

Description

This function visualizes the coefficients from the provided model results. It supports selection of specific models to display, grouping of coefficients for cleaner visualization, and focusing on specific coefficients through the 'show_only' parameter. Coefficients can be grouped into multiple charts to avoid cluttered plots.

Usage

```
## S3 method for class 'model'
plot(model_results, models = NULL, group = 20, show_only = NULL)
```

Arguments

model_results	A list containing the model results, which must include a named list of coefficients. Each element of the list should correspond to a different model.
models	Optional vector of model names to display. If NULL, all models in 'model_results' are used. If specified, only coefficients from these models are plotted.
group	The maximum number of coefficients to display per plot. If the number of coefficients or specified 'show_only' indices exceeds this number, the coefficients will be split across multiple plots. Defaults to 20. If 'group' is greater than 20, it will be set to 20 to maintain plot readability.
show_only	Optional numeric vector specifying the indices of the coefficients to display. This allows for focusing on specific coefficients. If not NULL, only the specified indices are shown. If 'show_only' contains more indices than 'group', coefficients will be distributed across multiple plots according to 'group'.

Value

A ggplot object displaying the specified or all coefficients. The plot includes options for interactivity, such as zooming and panning, if displayed in an interactive R environment.

Examples

```
# Same data generate process as slab_shrink
set.seed(123)
n <- 100 # Number of observations
p <- 50 # Number of variables
test_x <- matrix(rnorm(n * p), n, p)
beta <- matrix(rnorm(p + 1), p + 1, 1)
test_y <- cbind(1, test_x) %*% beta + rnorm(n, sd = 0.5)

# Fit a model using a fictional slab_shrink function
result <- slab_shrink(test_x, test_y) # Estimation works with just x and y

# Plot the model using default settings
plot.model(result)

# Plot the model for specific models "SR" and "ST"
plot.model(result, models = c("SR", "ST"))

# Plot the model with groups of 15 coefficients each
plot.model(result, group = 15)

# Plot the model showing only specified coefficient indices
plot.model(result, show_only = c(1:12, 30:40, 45:50))

# Expect an error when out-of-range indices are provided in show_only
# This line is meant to show error handling and should cause an error when run
tryCatch({
  plot.model(result, show_only = c(51:60)),
  error = function(e) cat("Expected error for out-of-range show_only: ", e$message, "\n")
})
```

```
})
```

slab_shrink

Slab and Shrinkage Generalised LASSO Estimator

Description

This function implements the Slab and Shrinkage Generalised LASSO Estimation method for regression models.

Usage

```
slab_shrink(
  x,
  y,
  u = 1,
  lambda = NULL,
  nlambdas = 100,
  models = c("SR", "ST", "DSH", "SH"),
  exclude = NULL,
  ...
)
```

Arguments

x	Input matrix of dimension $n \times p$, where n is the number of observations and p is the number of variables. The matrix should not include an intercept term as this will be handled internally if needed.
y	Response variable vector with n observations.
u	A numeric or numeric vector that represents the tuning parameter(s) for regularization strength in the model. Defaults to 1. If a vector is provided, the first element should be the tuning value for the intercept, followed by tuning values for other coefficients.
lambda	A numeric value, vector or NULL. If NULL, the function will automatically determine the optimal lambda value using cross-validation with <code>cv.glmnet()</code> . The optimal lambda, selected based on the given criteria (typically the one minimizing prediction error), will be used for subsequent regression models, and this value will be returned as the best lambda. If a lambda value is provided by the user, this specified value will be used directly in the regression models without further modification, and the same value will be returned.
nlambdas	The number of lambda values to be used if lambda is not specified. This is typically used in conjunction with cross-validation to determine the optimal lambda. Note that specifying both 'nlambdas' and 'lambda' simultaneously is not allowed and will result in an error.
models	A character vector specifying which regression models to estimate. Available choices are: <ul style="list-style-type: none"> 'SR': Slab Regression 'ST': Stein Regression

- 'DSH': Diagonal Shrinkage Regression
- 'SH': Shrinkage Regression

The specified models will be applied sequentially to the input data. If no models are provided by the user, all four models will be processed by default, and the results for each will be included in the output.

exclude An optional vector of column indices to be excluded from the input matrix *x* before the model fitting. This allows for the omission of variables that are not to be considered in the analysis.

...

Value

A list of results with class 'custom_lm' containing the following components:

coefficients A list of coefficient vectors for each specified model (SR, ST, DSH, SH).

fitted.values A list of fitted values for each specified model.

residuals A list of residuals for each specified model.

est The estimated parameter vector θ .

cvm The cross-validation mean squared error.

cvstd The cross-validation standard deviation.

lambda_range The range of lambda values used in the cross-validation.

u The regularization parameter vector u .

lambda The best lambda value selected.

sigma The estimated error variance.

model The list of models specified.

Examples

```
Set a random seed for reproducibility
set.seed(123)

# Define the number of observations and variables
n <- 100 # Number of observations
p <- 5   # Number of variables

# Generate random data for predictors
test_x <- matrix(rnorm(n * p), n, p)

# Generate coefficients including an intercept
beta <- matrix(rnorm(p + 1), p + 1, 1)

# Calculate response variable with added noise
test_y <- cbind(1, test_x) %*% beta + rnorm(n, sd = 0.5)

# Fit a model using a fictional slab_shrink function
result <- slab_shrink(test_x, test_y) # Estimation works with just x and y
print(result)

# Fit models "ST" and "DSH" using a fictional slab_shrink function
result_ST_DSH <- slab_shrink(test_x, test_y, models = c("ST", "DSH"))
print(result_ST_DSH)
```

```
# Fit a model using a fictional slab_shrink function with a specific u value
result_u_2 <- slab_shrink(test_x, test_y, u = 2)
print(result_u_2)
# Fit a model with a vector of u values including tuning for intercept
result_u_vector <- slab_shrink(test_x, test_y, u = c(0, 1:5))
print(result_u_vector)

# Fit a model with a specific lambda value
result_lambda_10 <- slab_shrink(test_x, test_y, lambda = 10)
print(result_lambda_10)
# Fit a model with a vector of randomly generated lambda values
result_lambda_vector <- slab_shrink(test_x, test_y, lambda = runif(10, min = 0, max = 10))
print(result_lambda_vector)

# Fit a model excluding the first and third predictors from x
# This might be useful to test the impact of specific variables on the model
result_exclude_1_3 <- slab_shrink(test_x, test_y, exclude = c(1,3))
print(result_exclude_1_3)
```