Package 'iilasso'

June 21, 2018

Type Package

Title Independently Interpretable Lasso

Version 0.0.2

Date 2018-06-21

Author Masaaki Takada

Maintainer Masaaki Takada <tkdmah@gmail.com>

Description Efficient algorithms for fitting linear / logistic regression model with Independently Interpretable Lasso.
Takada, M., Suzuki, T., & Fujisawa, H. (2018). Independently Interpretable Lasso: A New Regularizer for Sparse Regression with Uncorrelated Variables. AISTATS.
<http://proceedings.mlr.press/v84/takada18a/takada18a.pdf>.

License MIT + file LICENSE

Imports Rcpp, Matrix

LinkingTo Rcpp, BH

RoxygenNote 6.0.1

Suggests testthat, knitr, rmarkdown, MASS, parallel

VignetteBuilder knitr

URL http://proceedings.mlr.press/v84/takada18a/takada18a.pdf

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-06-21 16:52:35 UTC

R topics documented:

`cov_lasso` *Fit a linear regression model using a covariance matrix*

Description

Fit a linear regression model using a covariance matrix

Usage

```
cov_lasso(Gamma, gamma, lambda.min.ratio = 1e-04, nlambda = 100,
  lambda = NULL, delta = 0, alpha = NULL, R = NULL,
  funcR = function(G) {     abs(G)^2 }, maxit = 10000, eps = 1e-04,
  warm = "lambda", init.beta = NULL, strong = TRUE, sparse = FALSE,
  impl = "cpp", abs = TRUE)
```

Arguments

`Gamma`	covariance matrix of explanatory variables
`gamma`	covariance vector of explanatory and objective variables
`lambda.min.ratio`	ratio of max lambda and min lambda
`nlambda`	the number of lambda (ignored if lambda is specified)
`lambda`	lambda sequence
`delta`	ratio of regularization (exclusive penalty / l1 penalty) (default: 0)
`alpha`	mixing parameter of regularization of l1 and exclusive penalty terms (delta = (1 - alpha) / alpha)
`R`	matrix using exclusive penalty term
`funcR`	function of R (input: X, output: R)
`maxit`	max iteration (default: 1e+4)
`eps`	convergence threshold for optimization (default: 1e-4)
`warm`	warm start direction: "lambda" (default) or "delta"
`init.beta`	initial values of beta
`strong`	whether use strong screening (default) or not
`sparse`	whether use sparse matrix or not (default)
`impl`	implementation language of optimization: "cpp" (default) or "r"
`abs`	(experimental) whether use absolute value of beta (default) or not

Value

lasso model

beta_standard	standardized coefficients
lambda	regularization parameters
alpha	alpha defined above
delta	delta defined above

Examples

```
X <- matrix(c(1,2,3,5,4,7,6,8,9,10), nrow=5, ncol=2)
b <- matrix(c(-1,1), nrow=2, ncol=1)
e <- matrix(c(0,-0.1,0.1,-0.1,0.1), nrow=5, ncol=1)
y <- as.numeric(X %*% b + e)
fit <- lasso(X, y)
pr <- predict_lasso(fit, X)
plot_lasso(fit)
```

---

cv_lasso *Fit a model using a design matrix with cross validation*

Description

Fit a model using a design matrix with cross validation

Usage

```
cv_lasso(X, y, nfolds = 10, lambda.min.ratio = 1e-04, nlambda = 100,
  lambda = NULL, foldid = NULL, unit = "sample", seed, cl, ...)
```

Arguments

X	matrix of explanatory variables
y	vector of objective variable
nfolds	the number of folds (ignored if foldid is specified)
lambda.min.ratio	ratio of max lambda and min lambda (ignored if lambda is specified)
nlambda	the number of lambda (ignored if lambda is specified)
lambda	lambda sequence
foldid	vector indicating id of fold for each sample
unit	unit for cross validation error: "sample" (default) or "fold"
seed	random seed of cross validation
cl	(not yet implemented)
...	parameters of lasso function

Value

lasso model

`fit`	lasso model with hole data
`lambda.min`	lambda with minimum cross validation error
`lambda.min.index`	index of lambda.min
`lambda.1se`	largest lambda such that error is within 1 standard error of the minimum
`lambda.1se.index`	index of lambda.1se
`delta`	delta defined above
`foldid`	fold id
`cve`	cross validation error
`cvse`	cross validation standard error
`cvup`	cross validation error + standard error
`cvlo`	cross validation error - standard error
`pe`	prediction error (for family="binomial")

Examples

```
X <- matrix(c(1,2,3,5,4,7,6,8,9,10), nrow=5, ncol=2)
b <- matrix(c(-1,1), nrow=2, ncol=1)
e <- matrix(c(0,-0.1,0.1,-0.1,0.1), nrow=5, ncol=1)
y <- as.numeric(X %*% b + e)
cv_fit <- cv_lasso(X, y, nfolds=5)
fit <- cv_fit$fit
pr <- predict_lasso(fit, X, cv_fit$lambda.min)
plot_cv_lasso(cv_fit)
```

`lasso` *Fit a model using a design matrix*

Description

Fit a model using a design matrix

Usage

```
lasso(X, y, family = "gaussian", impl = "cpp", lambda.min.ratio = 1e-04,
  nlambda = 100, lambda = NULL, warm = "lambda", ...)
```

Arguments

X	matrix of explanatory variables
y	vector of objective variable
family	family of regression: "gaussian" (default) or "binomial"
impl	implementation language of optimization: "cpp" (default) or "r"
lambda.min.ratio	ratio of max lambda and min lambda (ignored if lambda is specified)
nlambda	the number of lambda (ignored if lambda is specified)
lambda	lambda sequence
warm	warm start direction: "lambda" (default) or "delta"
...	parameters for optimization

Value

lasso model

beta	coefficients
beta_standard	standardized coefficients
a0	intercepts
lambda	regularization parameters
alpha	alpha defined above
delta	delta defined above
family	family

Examples

```
X <- matrix(c(1,2,3,5,4,7,6,8,9,10), nrow=5, ncol=2)
b <- matrix(c(-1,1), nrow=2, ncol=1)
e <- matrix(c(0,-0.1,0.1,-0.1,0.1), nrow=5, ncol=1)
y <- as.numeric(X %*% b + e)
fit <- lasso(X, y)
pr <- predict_lasso(fit, X)
plot_lasso(fit)
```

---

logit_lasso *Fit a logistic regression model using a design matrix*

Description

Fit a logistic regression model using a design matrix

Usage

```
logit_lasso(X_tilde, y, lambda.min.ratio = 1e-04, nlambda = 100,
  lambda = NULL, delta = 0, alpha = NULL, R = NULL,
  funcR = function(G) {     abs(G)^2 }, maxit = 10000, eps = 1e-04,
  warm = "lambda", init.beta = NULL, strong = FALSE, sparse = FALSE,
  impl = "cpp", abs = TRUE)
```

Arguments

X_tilde	standardized matrix of explanatory variables
y	vector of objective variable
lambda.min.ratio	ratio of max lambda and min lambda
nlambda	the number of lambda (ignored if lambda is specified)
lambda	lambda sequence
delta	ratio of regularization (exclusive penalty / l1 penalty) (default: 0)
alpha	mixing parameter of regularization of l1 and exclusive penalty terms (delta = (1 - alpha) / alpha)
R	matrix using exclusive penalty term
funcR	function of R (input: X, output: R)
maxit	max iteration (default: 1e+4)
eps	convergence threshold for optimization (default: 1e-4)
warm	warm start direction: "lambda" (default) or "delta"
init.beta	initial values of beta
strong	whether use strong screening (default) or not
sparse	whether use sparse matrix or not (default)
impl	implementation language of optimization: "cpp" (default) or "r"
abs	(experimental) whether use absolute value of beta (default) or not

Value

lasso model

beta_standard	standardized coefficients
lambda	regularization parameters
alpha	alpha defined above
delta	delta defined above

Examples

```
X <- matrix(c(1,2,3,5,4,7,6,8,9,10), nrow=5, ncol=2)
b <- matrix(c(-1,1), nrow=2, ncol=1)
e <- matrix(c(0,-0.1,0.1,-0.1,0.1), nrow=5, ncol=1)
y <- as.numeric(X %*% b + e)
y <- ifelse(y>mean(y), 1, 0)
fit <- lasso(X, y, family="binomial")
pr <- predict_lasso(fit, X)
plot_lasso(fit)
```

---

plot_cv_lasso *Plot a cross validation error path*

Description

Plot a cross validation error path

Usage

```
plot_cv_lasso(cv_fit, ...)
```

Arguments

cv_fit	cross validated IILasso model
...	parameters of

Examples

```
X <- matrix(c(1,2,3,5,4,7,6,8,9,10), nrow=5, ncol=2)
b <- matrix(c(-1,1), nrow=2, ncol=1)
e <- matrix(c(0,-0.1,0.1,-0.1,0.1), nrow=5, ncol=1)
y <- as.numeric(X %*% b + e)
cv_fit <- cv_lasso(X, y, nfolds=5)
fit <- cv_fit$fit
pr <- predict_lasso(fit, X, cv_fit$lambda.min)
plot_cv_lasso(cv_fit)
```

plot_lasso *Plot a solution path*

Description

Plot a solution path

Usage

```
plot_lasso(fit, ...)
```

Arguments

fit	IILasso model
...	parameters of matlines function

Examples

```
X <- matrix(c(1,2,3,5,4,7,6,8,9,10), nrow=5, ncol=2)
b <- matrix(c(-1,1), nrow=2, ncol=1)
e <- matrix(c(0,-0.1,0.1,-0.1,0.1), nrow=5, ncol=1)
y <- as.numeric(X %*% b + e)
fit <- lasso(X, y)
pr <- predict_lasso(fit, X)
plot_lasso(fit)
```

predict_lasso *Predict responses*

Description

Predict responses

Usage

```
predict_lasso(fit, newx, s = NULL, type = "response")
```

Arguments

fit	IILasso model
newx	matrix of explanatory variables
s	selected lambda (default: all)
type	prediction type for logistic lasso: "response" (default) or "class"

Value

prediction matrix (if s is NULL) or vector (if s is specified)

Examples

```
X <- matrix(c(1,2,3,5,4,7,6,8,9,10), nrow=5, ncol=2)
b <- matrix(c(-1,1), nrow=2, ncol=1)
e <- matrix(c(0,-0.1,0.1,-0.1,0.1), nrow=5, ncol=1)
y <- as.numeric(X %*% b + e)
fit <- lasso(X, y)
pr <- predict_lasso(fit, X)
plot_lasso(fit)
```

---

setup_lambda *Set up a lambda sequence*

Description

Set up a lambda sequence

Usage

```
setup_lambda(X, y, family = "gaussian", lambda.min.ratio = 1e-04,
  nlambda = 100)
```

Arguments

X	matrix of explanatory variables
y	vector of objective variable
family	family of regression: "gaussian" (default) or "binomial"
lambda.min.ratio	ratio of max lambda and min lambda
nlambda	the number of lambda (ignored if lambda is specified)

Value

lambda

Examples

```
X <- matrix(c(1,2,3,5,4,7,6,8,9,10), nrow=5, ncol=2)
b <- matrix(c(-1,1), nrow=2, ncol=1)
e <- matrix(c(0,-0.1,0.1,-0.1,0.1), nrow=5, ncol=1)
y <- as.numeric(X %*% b + e)
setup_lambda(X, y)
```

Index