

Package ‘logitr’

June 1, 2018

Type Package

Title Penalized Logistic Regression

Version 0.1.0

Description This is an R package for linear and logistic regression with optional ridge and bridge regularization penalties.

URL <https://github.com/MGallow/logitr>

BugReports <https://github.com/MGallow/logitr/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends Rcpp (>= 0.12.10),
RcppArmadillo,
dplyr

LinkingTo Rcpp,
RcppArmadillo

RoxygenNote 6.0.1

Suggests testthat

R topics documented:

linearr	2
logisticr	3
predict.linearr	4
predict.logisticr	4
print.linearr	5
print.logisticr	5
Index	6

linearr

*Linear Regression***Description**

Computes the linear regression coefficient estimates (ridge-penalization and weights, optional)

Usage

```
linearr(X, y, lam = seq(0, 2, 0.1), alpha = 1.5, penalty = c("none",
  "ridge", "bridge"), weights = NULL, intercept = TRUE, kernel = FALSE,
  method = c("SVD", "MM"), tol = 1e-05, maxit = 1e+05, vec = NULL,
  init = 1, K = 5)
```

Arguments

X	matrix or data frame
y	matrix or data frame of response values
lam	optional tuning parameter for ridge regularization term. If passing a list of values, the function will choose the optimal value based on K-fold cross validation. Defaults to 'lam = seq(0, 2, 0.1)'
alpha	optional tuning parameter for bridge regularization term. If passing a list of values, the function will choose the optimal value based on K-fold cross validation. Defaults to 'alpha = 1.5'
penalty	choose from c('none', 'ridge', 'bridge'). Defaults to 'none'
weights	optional vector of weights for weighted least squares
intercept	add column of ones if not already present. Defaults to TRUE
kernel	use linear kernel to compute ridge regression coefficients. Defaults to TRUE when $p \gg n$ (for 'SVD')
method	optimization algorithm. Choose from 'SVD' or 'MM'. Defaults to 'SVD'
tol	tolerance - used to determine algorithm convergence for 'MM'. Defaults to 10^{-5}
maxit	maximum iterations for 'MM'. Defaults to 10^5
vec	optional vector to specify which coefficients will be penalized
init	optional initialization for MM algorithm
K	specify number of folds for cross validation, if necessary

Value

returns the selected tuning parameters, coefficient estimates, MSE, and gradients

Examples

```
library(dplyr)
X = dplyr::select(iris, -c(Species, Sepal.Length))
y = dplyr::select(iris, Sepal.Length)
linearr(X, y, lam = 0.1, penalty = 'ridge')
```

logisticr

*Logistic Regression***Description**

Computes the coefficient estimates for logistic regression. ridge regularization and bridge regularization optional.

Usage

```
logisticr(X, y, lam = seq(0, 2, 0.1), alpha = 1.5, penalty = c("none",
  "ridge", "bridge"), intercept = TRUE, method = c("IRLS", "MM"),
  tol = 1e-05, maxit = 1e+05, vec = NULL, init = 1,
  criteria = c("logloss", "mse", "misclass"), K = 5)
```

Arguments

X	matrix or data frame
y	matrix or vector of response values 0,1
lam	optional tuning parameter(s) for ridge regularization term. If passing a list of values, the function will choose optimal value based on K-fold cross validation. Defaults to 'lam = seq(0, 2, 0.1)'
alpha	optional tuning parameter for bridge regularization term. If passing a list of values, the function will choose the optimal value based on K-fold cross validation. Defaults to 'alpha = 1.5'
penalty	choose from c('none', 'ridge', 'bridge'). Defaults to 'none'
intercept	Defaults to TRUE
method	optimization algorithm. Choose from 'IRLS' or 'MM'. Defaults to 'IRLS'
tol	tolerance - used to determine algorithm convergence. Defaults to 10 ⁻⁵
maxit	maximum iterations. Defaults to 10 ⁵
vec	optional vector to specify which coefficients will be penalized
init	optional initialization for MM algorithm
criteria	specify the criteria for cross validation. Choose from c('mse', 'logloss', 'misclass'). Defaults to 'logloss'
K	specify number of folds for cross validation, if necessary

Value

returns selected tuning parameters, beta estimates (includes intercept), MSE, log loss, misclassification rate, total iterations, and gradients.

Examples

```
library(dplyr)
X = dplyr::select(iris, -Species)
y = dplyr::select(iris, Species)
y$Species = ifelse(y$Species == 'setosa', 1, 0)
logisticr(X, y)
```

```
# ridge Logistic Regression with IRLS
logisticr(X, y, lam = 0.1, penalty = 'ridge')

# ridge Logistic Regression with MM
logisticr(X, y, lam = 0.1, penalty = 'ridge', method = 'MM')
```

predict.linearrr	<i>Predict Linear Regression</i>
------------------	----------------------------------

Description

Generates prediction for linear regression. Note that one can either input a 'linearrr' object or a matrix of beta coefficients.

Usage

```
## S3 method for class 'linearrr'
predict(object, X, y = NULL)
```

Arguments

object	'linearrr' object or matrix of betas
X	matrix or data frame of (new) observations
y	optional, matrix or vector of response values

Value

predictions and loss metrics

Examples

```
library(dplyr)
X = dplyr::select(iris, -c(Species, Sepal.Length))
y = dplyr::select(iris, Sepal.Length)
fitted = linearrr(X, y, lam = 0.1)
predict(fitted, X)
```

predict.logisticr	<i>Predict Logistic Regression</i>
-------------------	------------------------------------

Description

Generates prediction for logistic regression. Note that one can either input a 'logisticr' object or a matrix of beta coefficients.

Usage

```
## S3 method for class 'logisticr'
predict(object, X, y = NULL)
```

Arguments

object	'logisticr' object or matrix of betas
X	matrix or data frame of (new) observations
y	optional, matrix or vector of response values 0,1

Value

predictions and loss metrics

Examples

```
library(dplyr)
X = dplyr::select(iris, -Species)
y = dplyr::select(iris, Species)
y$Species = ifelse(y$Species == 'setosa', 1, 0)
logisticr(X, y)

fitted = logisticr(X, y, lam = 0.1, penalty = 'ridge', method = 'MM')
predict(fitted, X)
```

print.linearrr	<i>Print linearrr object</i>
----------------	------------------------------

Usage

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

Arguments

x	linearrr class object
---	-----------------------

print.logisticr	<i>Print logisticr object</i>
-----------------	-------------------------------

Usage

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

Arguments

x	logisticr class object
---	------------------------

Index

linearr, [2](#)
logisticr, [3](#)

predict.linearr, [4](#)
predict.logisticr, [4](#)
print.linearr, [5](#)
print.logisticr, [5](#)