

Package ‘pros’

December 16, 2018

Title Penalized Regression on Steroids

Version 0.1

Author Austin David Brown <brow5079@umn.edu>

Maintainer Austin David Brown <brow5079@umn.edu>

Description This is a project for STAT8053 at the University of Minnesota. Please see the README on github for compiler flags to improve performance.

Depends R (>= 3.5.0)

SystemRequirements C++11

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

R topics documented:

cv.pros	1
predict.cv_pros	2
predict.pros	3
pros	3

Index	5
--------------	----------

cv.pros	<i>Cross-validation</i>
---------	-------------------------

Description

The cv.pros function is used for K-fold cross-validation.

Usage

```
cv.pros(X, y, K_fold = 10, alpha = c(1, 0, 0, 0, 0, 0),  
        lambdas = c(), step_size, algorithm = "proximal_gradient_cd",  
        max_iter = 10000, tolerance = 10^(-8), random_seed = 0)
```

Arguments

<code>X</code>	is an $n \times m$ -dimensional matrix of the data.
<code>y</code>	is an $n \times m$ -dimensional matrix of the data.
<code>K_fold</code>	is the number of folds in cross-validation.
<code>alpha</code>	is a 6-dimensional vector of the convex combination corresponding to the penalization: <ul style="list-style-type: none"> • α_1 is the l^1 penalty. • α_2 is the l^2 penalty. • α_3 is the l^4 penalty. • α_4 is the l^6 penalty. • α_5 is the l^8 penalty. • α_6 is the l^{10} penalty.
<code>lambdas</code>	is a vector of dual penalization values to be evaluated.
<code>step_size</code>	is a tuning parameter defining the step size. Larger values are more aggressive and smaller values are less aggressive.
<code>algorithm</code>	is the optimization algorithm <ul style="list-style-type: none"> • <code>proximal_gradient_cd</code> uses proximal gradient coordinate descent. • <code>subgradient_cd</code> uses subgradient coordinate descent.
<code>max_iter</code>	is the maximum iterations the algorithm will run regardless of convergence.
<code>tolerance</code>	is the accuracy of the stopping criterion.
<code>random_seed</code>	is the random seed used in the algorithms.

Value

A class `cv_pros`

<code>predict.cv_pros</code>	<i>Cross-validation Prediction</i>
------------------------------	------------------------------------

Description

The prediction function for `cv.pros`.

Usage

```
## S3 method for class 'cv_pros'
predict(object, X_new, ...)
```

Arguments

<code>object</code>	an object of class <code>cv_pros</code>
<code>X_new</code>	is an $n \times m$ -dimensional matrix of the data.
<code>...</code>	Other parameters (this is required by R)

Value

A vector of prediction values.

predict.pros	<i>Pros Prediction</i>
--------------	------------------------

Description

The prediction function for pros.

Usage

```
## S3 method for class 'pros'
predict(object, X, ...)
```

Arguments

object	an object of class pros
X	is an $n \times m$ -dimensional matrix of the data.
...	Other parameters (this is required by R)

Value

A vector of prediction values.

pros	<i>Pros</i>
------	-------------

Description

The pros function is used to fit a single regression model with a specified penalization.

Usage

```
pros(X, y, alpha = c(1, 0, 0, 0, 0, 0), lambda, step_size,
      algorithm = "proximal_gradient_cd", max_iter = 10000,
      tolerance = 10^(-8), random_seed = 0)
```

Arguments

X	is an $n \times m$ -dimensional matrix of the data.
y	is an $n \times m$ -dimensional matrix of the data.
alpha	is a 6-dimensional vector of the convex combination corresponding to the penalization: <ul style="list-style-type: none"> • α_1 is the l^1 penalty. • α_2 is the l^2 penalty. • α_3 is the l^4 penalty. • α_4 is the l^6 penalty. • α_5 is the l^8 penalty. • α_6 is the l^{10} penalty.

lambda	is the Lagrangian dual penalization parameter.
step_size	is a tuning parameter defining the step size. Larger values are more aggressive and smaller values are less aggressive.
algorithm	is the optimization algorithm <ul style="list-style-type: none">• proximal_gradient_cd uses proximal gradient coordinate descent.• subgradient_cd uses subgradient coordinate descent.
max_iter	is the maximum iterations the algorithm will run regardless of convergence.
tolerance	is the accuracy of the stopping criterion.
random_seed	is the random seed used in the algorithms.

Value

A class pros

Index

`cv.pros`, [1](#)

`predict.cv_pros`, [2](#)

`predict.pros`, [3](#)

`pros`, [3](#)