Package 'CDLasso'

February 19, 2015

Type Package

Title Coordinate Descent Algorithms for Lasso Penalized L1, L2, and Logistic Regression

Version 1.1

Date 2013-19-03

Author Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer Edward Grant <edward.m.grant@gmail.com>

Description Coordinate Descent Algorithms for Lasso Penalized L1, L2, and Logistic Regression

License GPL-2

LazyLoad yes

LazyData yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-05-17 22:34:35

R topics documented:

`CDLasso-package` *Coordinate descent algorithms for L1 and L2 regression*

Description

Greedy coordinate descent for L1 regression and cyclic coordinate descent for L2 regression with p predictors and n cases

Details

Package:	CDLasso
Title:	Coordinate Descent for L1 and L2 Regression
Version:	1.1
Date:	2013-13-03
Author:	Edward Grant, Kenneth Lange, Tong Tong Wu
Maintainer:	Edward Grant <edward.m.grant@gmail.com>
Description:	Coordinate Descent for L1, L2, and Logistic Regression
License:	GPL-2

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant `<edward.m.grant@gmail.com>`

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

`l1.reg`

`l2.reg`

`logit.reg`

Examples

```
set.seed(1001)
n=500
p=2000
nz = c(1:5)
true.beta<-rep(0,p)
true.beta[nz] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

logity=exp(y)/(1+exp(y))
ylog=rbinom(n=length(logity),prob=logity,size=1)

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

#L1
outL1<-l1.reg(x,y,lambda=50)
outL1est<-l1.reg(x[outL1$selected,],y,lambda=0)

#L2
outL2<-l2.reg(x,y,2)
outL2est<-l2.reg(x[outL2$selected,],y,lambda=0)

#Logistic
outLOGIT<-logit.reg(x,ylog,lambda=50)
outLOGITest<-logit.reg(x[outLOGIT$selected,],ylog,lambda=0)
```

`cv.l1.reg` *k-fold Cross Validation*

Description

k-fold Cross Validation for L1 Greedy Coordinate Descent

Usage

```
cv.l1.reg(x, y, k, lam.vec)
```

Arguments

`x`	p x n design matrix - Note that the rows of X correspond to predictors and the columns to cases.
`y`	Outcome of length `n`
`k`	Number of folds for k-fold cross validation
`lam.vec`	Vector of penalization parameters

Details

K-fold cross validation to select optimal lambda for use in greedy coordinate descent for L1 regression l1.reg. The optimal value is considered the lambda value that retuns the lowest prediction error over the cross validation. If more than one lambda value give the minumum testing error, the smallest lambda is selected. Plot of the cross validation can be viewed through plot.cv.l1.reg

Value

k	The value of K used for the K-fold cross validation.
lam.vec	The values of lambda tested.
mean.error	The mean error corresponding to each lambda across k-folds
lam.opt	The determined lambda value among lam.vec that returns the smallest prediction error. This value is the optimal lambda value for use in l1.reg.
error.cv	The prediction error matrix returned by cross validation method.
num.pred	The number of predictors selected for the corresponding lambda during the cross validation.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

l1.reg

plot.cv.l1.reg

Examples

```
set.seed(6)
n=50
p=200
nzfixed = c(1:5)
true.beta<-rep(0,p)
true.beta[nzfixed] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

crossval<-cv.l1.reg(x,y,10,(0:20)*2)
```

```
plot(crossval)
out<-l1.reg(x,y,lambda=crossval$lam.opt)
out
```

`cv.l2.reg` *k-fold Cross Validation*

Description

k-fold Cross Validation for L2 Cyclic Coordinate Descent

Usage

```
cv.l2.reg(x, y, k, lam.vec)
```

Arguments

`x`	`p` x `n` design matrix - Note that the rows of X correspond to predictors and the columns to cases.
`y`	Outcome of length `n`
`k`	Number of folds for k-fold cross validation
`lam.vec`	Vector of penalization parameters

Details

K-fold cross validation to select optimal lambda for use in cyclic coordinate descent for L2 regression `l2.reg`. The optimal value is considered the `lambda` value that retuns the lowest prediction error over the cross validation. If more than one `lambda` value give the minumum testing error, the smallest `lambda` is selected. Plot of the cross validation can be viewed through `plot.cv.l2.reg`

Value

`k`	The value of `K` used for the K-fold cross validation.
`lam.vec`	The values of lambda tested.
`mean.error`	The mean error corresponding to each lambda across k-folds
`lam.opt`	The determined lambda value among `lam.vec` that returns the smallest prediction error. This value is the optimal `lambda` value for use in `l2.reg`.
`error.cv`	The prediction error matrix returned by cross validation method.
`num.pred`	The number of predictors selected for the corresponding `lambda` during the cross validation.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

l2.reg

plot.cv.l2.reg

Examples

```
set.seed(100)
n=50
p=200
nzfixed = c(1:5)
true.beta<-rep(0,p)
true.beta[nzfixed] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

crossval2<-cv.l2.reg(x,y,10,exp((1:15))/1000)
plot(crossval2)
out2<-l2.reg(x,y,crossval2$lam.opt)
```

---

cv.logit.reg *k-fold Cross Validation*

Description

k-fold Cross Validation to find optimal lambda for Cyclic Coordinate Descent for logistic regression

Usage

```
cv.logit.reg(x, y, k, lam.vec)
```

Arguments

x	p x n design matrix - Note that the rows of X correspond to predictors and the columns to cases.
y	Outcome of length n. Outcome must be 0 and 1.
k	Number of folds for k-fold cross validation
lam.vec	Vector of penalization parameters

Details

K-fold cross validation to select optimal lambda for use in cyclic coordinate descent for logistic regression logit.reg. The optimal value is considered the lambda value that retuns the lowest testing error over the cross validation. If more than one lambda value give the minumum testing error, the largest lambda is selected. Plot of the cross validation can be viewed through plot.cv.logit.reg

Value

k	The value of K used for the K-fold cross validation.
lam.vec	The values of lambda tested.
lam.opt	The determined lambda value among lam.vec that returns the smallest prediction error. This value is the optimal lambda value for use in logit.reg.
error.cv	The prediction error matrix returned by cross validation method.
num.pred	The number of selected predictors when using the corresponding lambda value.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T., Chen, Y.F., Hastie, T., Sobel E. and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics, Volume 25, No 6, 714-721.

See Also

logit.reg

plot.cv.logit.reg

Examples

```
set.seed(1001)
n=250;p=50
beta=c(1,1,1,1,1,rep(0,p-5))
x=matrix(rnorm(n*p),p,n)
xb = t(x) %*% beta
logity=exp(xb)/(1+exp(xb))
y=rbinom(n=length(logity),prob=logity,size=1)

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)
lam.vec = (0:15)*2

#K-fold cross validation
cv <- cv.logit.reg(x,y,5,lam.vec)
plot(cv)
cv
```

```
#Lasso penalized logistic regression using optimal lambda
out<-logit.reg(x,y,cv$lam.opt)

#Re-estimate parameters without penalization
out2<-logit.reg(x[out$selected,],y,0)
out2
```

`l1.reg` *Greedy Coordinate Descent for L1 regression*

Description

Greedy Coordinate Descent for L1 regression with *p* predictors and *n* cases

Usage

```
l1.reg(X, Y, lambda = 1)
```

Arguments

`X`	p x n design matrix - Note that the rows of X correspond to predictors and the columns to cases.
`Y`	Outcome of length `n`
`lambda`	Penalization Parameter. To find optimal `lambda`, use cv.l1.reg.

Details

`l1.reg` performs a new algorithm for estimating regression coefficients with a lasso penalty. The algorithm is based on greedy coordinate descent and Edgeworth's algorithm for ordinary L1 regression. This L1 algorithm is faster than the cyclic coordinate descent in L2 regression (l2.reg).

Value

`X`	The design matrix.
`Y`	The outcome variable for cases.
`cases`	The number of cases
`predictors`	The number of predictors
`lambda`	The value of penalization parameter `lambda` used.
`objective`	The value of the objective function
`residual`	A vector of length `p` listing the residuals
`L1`	The sum of the residuals
`estimate`	The estimate of the coefficients
`nonzeros`	The name of "selected" variables included in the model.
`selected`	The name of the "selected" variables included in the model.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

print.l1.reg

summary.l1.reg

cv.l1.reg

plot.cv.l1.reg

l2.reg

Examples

```
set.seed(100)
n=500
p=2000
nz = c(1:5)
true.beta<-rep(0,p)
true.beta[nz] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

#Lasso penalized L1 regression
out<-l1.reg(x,y,lambda=50)

#Re-estimate parameters without penalization
out2<-l1.reg(x[out$selected,],y,lambda=0)
out2
```

---

l2.reg *Cyclic Coordinate Descent for L2 regression*

Description

Cyclic Coordinate Descent for L2 regression with p predictors and n cases

Usage

```
l2.reg(X, Y, lambda = 1)
```

Arguments

X	p x n design matrix - Note that the rows of X correspond to predictors and the columns to cases.
Y	Outcome of length n
lambda	Penalization Parameter. For optimal lambda, use cv.l2.reg.

Details

l2.reg performs an algorithm for estimating regression coefficients in a penalized L2 regression model. The algorithm is based on cyclic coordinate descent. For the new L1 algorithm that is faster, see (l1.reg).

Value

X	The design matrix.
cases	The number of cases
predictors	The number of predictors
lambda	The value of penalization parameter lambda used.
residual	A vector of length p listing the residuals
L2	The sum of the residuals
estimate	The estimate of the coefficients
nonzeros	The number "selected" variables included in the model.
selected	The name of the "selected" variables included in the model.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

print.l2.reg

summary.l2.reg

cv.l2.reg

plot.cv.l2.reg

l1.reg

Examples

```
set.seed(100)
n=500
p=2000
nzfixed = c(1:5)
true.beta<-rep(0,p)
true.beta[nzfixed] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

#Lasso penalized L2 regression
out<-l2.reg(x,y,lambda=2)

#Re-estimate parameters without penalization
out2<-l2.reg(x[out$selected,],y,lambda=0)
out2
```

`logit.reg` *Cyclic Coordinate Descent for Logistic regression*

Description

Cyclic Coordinate Descent for Logistic regression with p predictors and n cases

Usage

```
logit.reg(X, Y, lambda = 1)
```

Arguments

`X`	p x n design matrix - Note that the rows of X correspond to predictors and the columns to cases.
`Y`	Outcome of length n
`lambda`	Penalization Parameter. For optimal `lambda`, use cv.logit.reg.

Details

`logit.reg` performs an algorithm for estimating regression coefficients in a penalized logistic regression model. The algorithm is based on cyclic coordinate descent.

Value

X	The design matrix.
cases	The number of cases
predictors	The number of predictors
lambda	The value of penalization parameter lambda used.
residual	A vector of length p listing the residuals
estimate	The estimate of the coefficients
nonzeros	The number "selected" variables included in the model.
selected	The name of the "selected" variables included in the model.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T., Chen, Y.F., Hastie, T., Sobel E. and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics, Volume 25, No 6, 714-721.

See Also

print.logit.reg

summary.logit.reg

cv.logit.reg

plot.cv.logit.reg

l1.reg

Examples

```
set.seed(1001)
n=500;p=5000
beta=c(1,1,1,1,1,rep(0,p-5))
x=matrix(rnorm(n*p),p,n)
xb = t(x) %*% beta
logity=exp(xb)/(1+exp(xb))
y=rbinom(n=length(logity),prob=logity,size=1)

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

#Lasso penalized logistic regression using optimal lambda
out<-logit.reg(x,y,50)
print(out)

#Re-estimate parameters without penalization
```

```
out2<-logit.reg(x[out$selected,],y,0)
out2
```

`plot.cv.l1.reg` *Cross validation plot*

Description

Plot cross validation results across lambdas for greedy coordinate descent for L1 regression

Usage

```
## S3 method for class 'cv.l1.reg'
plot(x, ...)
```

Arguments

x	Output of cv.l1.reg. Must be of class "cv.l1.reg"
...	N/A

Details

`plot.cv.l1.reg` plots the prediction error of k fold cross validation across lambda values.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

`cv.l1.reg`

`l1.reg`

Examples

```
set.seed(100)
n=50
p=200
nzfixed = c(1:5)
true.beta<-rep(0,p)
true.beta[nzfixed] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

crossval<-cv.l1.reg(x,y,10,(0:20)*2)
plot(crossval)
out<-l1.reg(x,y,lambda=crossval$lam.opt)
out
```

`plot.cv.l2.reg` *Cross validation plot*

Description

Plot cross validation results across lambdas for greedy coordinate descent for L2 regression

Usage

```
## S3 method for class 'cv.l2.reg'
plot(x, ...)
```

Arguments

x	Output of cv.l2.reg. Must be of class "cv.l2.reg"
...	N/A

Details

`plot.cv.l2.reg` plots the prediction error of k fold cross validation across lambda values.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

cv.l2.reg

l2.reg

Examples

```
set.seed(100)
n=50
p=200
nzfixed = c(1:5)
true.beta<-rep(0,p)
true.beta[nzfixed] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

crossval2<-cv.l2.reg(x,y,10,exp((1:15))/1000)
plot(crossval2)
out2<-l2.reg(x,y,crossval2$lam.opt)
```

---

plot.cv.logit.reg *Cross validation plot*

Description

Plot cross validation results across lambdas for greedy coordinate descent for logistic regression

Usage

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

Arguments

x	Output of cv.logit.reg. Must be of class "cv.logit.reg"
...	N/A

Details

plot.cv.logit.reg plots the prediction error of k fold cross validation across lambda values.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T., Chen, Y.F., Hastie, T., Sobel E. and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics, Volume 25, No 6, 714-721.

See Also

cv.logit.reg

logit.reg

Examples

```
set.seed(101)
n=250;p=50
beta=c(1,1,1,1,1,rep(0,p-5))
x=matrix(rnorm(n*p),p,n)
xb = t(x) %*% beta
logity=exp(xb)/(1+exp(xb))
y=rbinom(n=length(logity),prob=logity,size=1)

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)
lam.vec = (0:15)*2

#K-fold cross validation
cv <- cv.logit.reg(x,y,5,lam.vec)
plot(cv)

#Lasso penalized logistic regression using optimal lambda
out<-logit.reg(x,y,cv$lam.opt)

#Re-estimate parameters without penalization
out2<-logit.reg(x[out$selected,],y,0)
out2$estimate
```

---

print.cv.l1.reg *Print results of Greedy Coordinate Descent for L1 Regression*

Description

Print short summary of results of cross validation for Greedy Coordinate Descent for L1 Regression.

Usage

```
## S3 method for class 'cv.l1.reg'
print(x, ...)
```

Arguments

x	Output of cv.l1.reg. Must be of class "cv.l1.reg"
...	N/A

Details

print.cv.l1.reg produces output from cv.l1.reg.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

cv.l1.reg

Examples

```
set.seed(100)
n=50
p=200
nzfixed = c(1:5)
true.beta<-rep(0,p)
true.beta[nzfixed] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

crossval<-cv.l1.reg(x,y,10,(0:20)*2)
print(crossval)
```

---

print.cv.l2.reg *Print results of Greedy Coordinate Descent for L2 Regression*

Description

Print short summary of results of cross validation for Greedy Coordinate Descent for L2 Regression.

Usage

```
## S3 method for class 'cv.l2.reg'
print(x, ...)
```

Arguments

x	Output of cv.l2.reg. Must be of class "cv.l2.reg"
...	N/A

Details

print.cv.l2.reg produces output from cv.l2.reg.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

cv.l2.reg

Examples

```
set.seed(100)
n=50
p=200
nzfixed = c(1:5)
true.beta<-rep(0,p)
true.beta[nzfixed] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

crossval2<-cv.l2.reg(x,y,10,exp((1:15))/1000)
plot(crossval2)
out2<-l2.reg(x,y,crossval2$lam.opt)
```

print.cv.logit.reg *Print results of Greedy Coordinate Descent for Logistic Regression*

Description

Print short summary of results of cross validation for Greedy Coordinate Descent for Logistic Regression.

Usage

```
## S3 method for class 'cv.logit.reg'
print(x, ...)
```

Arguments

x	Output of cv.logit.reg. Must be of class "cv.logit.reg"
...	N/A

Details

print.cv.logit.reg produces output from cv.logit.reg.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T., Chen, Y.F., Hastie, T., Sobel E. and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics, Volume 25, No 6, 714-721.

See Also

cv.logit.reg

Examples

```
set.seed(101)
n=250;p=50
beta=c(1,1,1,1,1,rep(0,p-5))
x=matrix(rnorm(n*p),p,n)
xb = t(x) %*% beta
logity=exp(xb)/(1+exp(xb))
y=rbinom(n=length(logity),prob=logity,size=1)

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)
```

```
lam.vec = (0:15)*2

#K-fold cross validation
cv <- cv.logit.reg(x,y,5,lam.vec)
plot(cv)

#Lasso penalized logistic regression using optimal lambda
out<-logit.reg(x,y,cv$lam.opt)

#Re-estimate parameters without penalization
out2<-logit.reg(x[out$selected,],y,0)
out2$estimate
```

`print.l1.reg` *Print results of Greedy Coordinate Descent for L1 Regression*

Description

Print short summary of results of Greedy Coordinate Descent for L1 Regression. Includes number of cases and predictors, lambda used, estimate of coeffcients produced, the number of selected predictors, and the names of selected predictors.

Usage

```
## S3 method for class 'l1.reg'
print(x, ...)
```

Arguments

`x`	Output of l1.reg. Must be of class "l1.reg"
`...`	N/A

Details

`print.l1.reg` produces selected output from `l1.reg`. For more output, see summary.l1.reg.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

summary.l1.reg

l1.reg

Examples

```
set.seed(100)
n=500
p=2000
nz = c(1:5)
true.beta<-rep(0,p)
true.beta[nz] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

#Lasso penalized L1 regression
out<-l1.reg(x,y,lambda=50)

#Re-estimate parameters without penalization
out2<-l1.reg(x[out$selected,],y,lambda=0)
print(out2)
```

---

print.l2.reg *Print results of Cyclic Coordinate Descent for L2 Regression*

Description

Print short summary of results of Cyclic Coordinate Descent for L2 Regression. Includes number of cases and predictors, lambda used, estimate of coeffcients produced, the number of selected predictors, and the names of selected predictors.

Usage

```
## S3 method for class 'l2.reg'
print(x, ...)
```

Arguments

x	Output of l2.reg. Must be of class "l2.reg"
...	N/A

Details

print.L1_REG produces selected output from l2.reg. For more output, see summary.l2.reg.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

summary.l2.reg

l2.reg

Examples

```
set.seed(100)
n=500
p=2000
nzfixed = c(1:5)
true.beta<-rep(0,p)
true.beta[nzfixed] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

#Lasso penalized L2 regression
out<-l2.reg(x,y,lambda=2)

#Re-estimate parameters without penalization
out2<-l2.reg(x[out$selected,],y,lambda=0)
print(out2)
```

---

print.logit.reg *Print results of Cyclic Coordinate Descent for Logistic Regression*

Description

Print short summary of results of Cyclic Coordinate Descent for Logistic Regression. Includes number of cases and predictors, lambda used, estimate of coeffcients produced, the number of selected predictors, and the names of selected predictors.

Usage

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

Arguments

x	Output of logit.reg. Must be of class "logit.reg"
...	N/A

Details

`print.logit.reg` produces selected output from logit.reg. For more output, see summary.logit.reg.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T., Chen, Y.F., Hastie, T., Sobel E. and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics, Volume 25, No 6, 714-721.

See Also

summary.logit.reg

logit.reg

Examples

```
set.seed(1001)
n=500;p=5000
beta=c(1,1,1,1,1,rep(0,p-5))
x=matrix(rnorm(n*p),p,n)
xb = t(x) %*% beta
logity=exp(xb)/(1+exp(xb))
y=rbinom(n=length(logity),prob=logity,size=1)

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

#Lasso penalized logistic regression using optimal lambda
out<-logit.reg(x,y,50)
print(out)

#Re-estimate parameters without penalization
out2<-logit.reg(x[out$selected,],y,0)
print(out2)
```

summary.l1.reg *Print all results of Greedy Coordinate Descent for L1 Regression*

Description

Print full summary of results of Greedy Coordinate Descent for L1 Regression.

Usage

```
## S3 method for class 'l1.reg'
summary(object, ...)
```

Arguments

object	Output of l1.reg. Must be of class "l1.reg"
...	N/A

Details

summary.l1.reg produces full output from l1.reg. For selected output, see print.l1.reg.

Value

X	The design matrix.
Y	The outcome variable for cases.
cases	The number of cases
predictors	The number of predictors
lambda	The value of penalization parameter lambda used.
objective	The value of the objective function
residual	A vector of length p listing the residuals
L1	The sum of the residuals
estimate	The estimate of the coefficients
nonzeros	The name of "selected" variables included in the model.
selected	The name of the "selected" variables included in the model.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

summary.l1.reg

l1.reg

Examples

```
set.seed(100)
n=500
p=2000
nz = c(1:5)
true.beta<-rep(0,p)
true.beta[nz] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

#Lasso penalized L1 regression
out<-l1.reg(x,y,lambda=50)

#Re-estimate parameters without penalization
out2<-l1.reg(x[out$selected,],y,lambda=0)
summary(out2)
```

---

summary.l2.reg *Print all results of Cyclic Coordinate Descent for L2 Regression*

Description

Print full summary of results of Cyclic Coordinate Descent for L2 Regression.

Usage

```
## S3 method for class 'l2.reg'
summary(object, ...)
```

Arguments

object	Output of l2.reg. Must be of class "l2.reg"
...	N/A

Details

summary.l2.reg produces full output from l2.reg. For selected output, see print.l2.reg.

Value

X	The design matrix.
Y	The outcome variable for cases.
cases	The number of cases
predictors	The number of predictors
lambda	The value of penalization parameter lambda used.
objective	The value of the objective function
residual	A vector of length p listing the residuals
L2	The sum of the residuals
estimate	The estimate of the coefficients
nonzeros	The name of "selected" variables included in the model.
selected	The name of the "selected" variables included in the model.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. Annals of Applied Statistics, Volume 2, No 1, 224-244.

See Also

summary.l2.reg

l2.reg

Examples

```
set.seed(100)
n=500
p=2000
nz = c(1:5)
true.beta<-rep(0,p)
true.beta[nz] = c(1,1,1,1,1)

x=matrix(rnorm(n*p),p,n)
y = t(x) %*% true.beta

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

#Lasso penalized L2 regression
out<-l2.reg(x,y,lambda=2)
```

```
#Re-estimate parameters without penalization
out2<-l2.reg(x[out$selected,],y,lambda=0)
summary(out2)
```

`summary.logit.reg` *Print all results of Cyclic Coordinate Descent for Logistic Regression*

Description

Print full summary of results of Cyclic Coordinate Descent for Logistic Regression.

Usage

```
## S3 method for class 'logit.reg'
summary(object, ...)
```

Arguments

`object`	Output of logit.reg. Must be of class "logit.reg"
`...`	N/A

Details

`summary.logit.reg` produces full output from logit.reg. For selected output, see print.logit.reg.

Value

`X`	The design matrix.
`Y`	The outcome variable for cases.
`cases`	The number of cases
`predictors`	The number of predictors
`lambda`	The value of penalization parameter `lambda` used.
`objective`	The value of the objective function
`residual`	A vector of length `p` listing the residuals
`estimate`	The estimate of the coefficients
`nonzeros`	The number of "selected" variables included in the model.
`selected`	The name of the "selected" variables included in the model.

Author(s)

Edward Grant, Kenneth Lange, Tong Tong Wu

Maintainer: Edward Grant <edward.m.grant@gmail.com>

References

Wu, T.T., Chen, Y.F., Hastie, T., Sobel E. and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics, Volume 25, No 6, 714-721.

See Also

summary.logit.reg

logit.reg

Examples

```
set.seed(1001)
n=500;p=5000
beta=c(1,1,1,1,1,rep(0,p-5))
x=matrix(rnorm(n*p),p,n)
xb = t(x) %*% beta
logity=exp(xb)/(1+exp(xb))
y=rbinom(n=length(logity),prob=logity,size=1)

rownames(x)<-1:nrow(x)
colnames(x)<-1:ncol(x)

#Lasso penalized logistic regression
out<-logit.reg(x,y,lambda=50)

#Re-estimate parameters without penalization
out2<-logit.reg(x[out$selected,],y,lambda=0)
summary(out2)
```

Index