# Package 'extlasso'

February 19, 2015

**Type** Package

**Title** Maximum penalized likelihood estimation with extended lasso penalty

**Version** 0.2

**Date** 2014-19-08

**Author** B N Mandal <mandal.stat@gmail.com> and Jun Ma <jun.ma@mq.edu.au>

**Maintainer** B N Mandal <mandal.stat@gmail.com>

**Depends** R (>= 3.1.1)

**Description** The package estimates coefficients of extended LASSO penalized linear regression and generalized linear models. Currently lasso and elastic net penalized linear regression and generalized linear models are considered. The package currently utilizes an accurate approximation of L1 penalty and then a modified Jacobi algorithm to estimate the coefficients. There is provision for plotting of the solutions and predictions of coefficients at given values of lambda. The package also contains functions for cross validation to select a suitable lambda value given the data. The package also provides a function for estimation in fused lasso penalized linear regression.

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-08-19 07:54:21

## R topics documented:

coef.extlasso *Extract coefficients from a fitted extlasso object*

## Description

The function returns the coefficients from a fitted extlasso object

## Usage

```
## S3 method for class 'extlasso'
coef(object,...)
```

## Arguments

| | |
|---|---|
| object | A 'extlasso' object obtained using 'extlasso' function. |
| ... | Not used |

## Value

Estimated coefficients for different lambdas starting from maximum value of lambda to minimum value of lambda

## Author(s)

B N Mandal and Jun Ma

## References

Mandal, B.N. and Jun Ma, (2014). A Jacobi-Armijo Algorithm for LASSO and its Extensions.

## Examples

```
x=matrix(rnorm(100*30),100,30)
y=sample(c(0,1),100,replace=TRUE)
g1=extlasso(x,y,family="binomial")
coef(g1)
x=matrix(rnorm(100*30),100,30)
y=rnorm(100)
g1=extlasso(x,y,family="normal")
coef(g1)
```

`cv.extlasso` *k-fold cross validation for penalized generalized linear models for normal/binomial/poisson family*

## Description

The function does k-fold cross validation for selecting best value of regularization parameter.

## Usage

```
cv.extlasso(x,y,family=c("binomial","normal","poisson"),k=5,
nlambda=50,tau=1,plot=TRUE, errorbars=TRUE)
```

## Arguments

| | |
|---|---|
| `x` | x is matrix of order n x p where n is number of observations and p is number of predictor variables. Rows should represent observations and columns should represent predictor variables. |
| `y` | y is a vector of response variable of order n x 1. |
| `family` | family is either "normal" or "binomial" or "poisson". |
| `k` | Number of folds for cross validation. Default is k=5. |
| `nlambda` | Number of lambda values to be used for cross validation. Default is nlambda=50. |
| `tau` | Elastic net parameter, $0 \leq \tau \leq 1$ in elastic net penalty $\lambda\{\tau\|\beta\|_1 + (1-\tau)\|beta\|_2^2\}$. Default tau=1 corresponds to LASSO penalty. |
| `plot` | if TRUE, produces a plot of cross validated prediction mean squared errors/ deviances against lambda. Default is TRUE. |
| `errorbars` | If TRUE, error bars are drawn in the plot. Default is TRUE. |

## Value

Produces a plot and returns a list with following components:

| | |
|---|---|
| `lambda` | Value of lambda for which average cross validation error is minimum |
| `pmse` | A vector of average cross validation errors for various lambda values |
| `lambdas` | A vector of lambda values used in cross validation |
| `se` | A vector containing standard errors of cross validation errors |

## Note

This function uses prediction means squared errors for normal family and deviance for binomial and poisson family.

## Author(s)

B N Mandal and Jun Ma

## References

Mandal, B.N. and Jun Ma, (2014). A Jacobi-Armijo Algorithm for LASSO and its Extensions.

## Examples

```
#normal family
x=matrix(rnorm(100*30),100,30)
y=rnorm(100)
cv.extlasso(x,y,family="normal",k=5)

#binomial family
x=matrix(rnorm(100*30),100,30)
y=sample(c(0,1),100,replace=TRUE)
cv.extlasso(x,y,family="binomial",k=5)

#poisson family
x=matrix(rnorm(100*30),100,30)
y=sample(c(1:5),100,replace=TRUE)
cv.extlasso(x,y,family="poisson",k=5)
```

---

`extlasso` *Entire regularization path of penalized generalized linear model for normal/binomial/poisson family using modified Jacobi Algorithm*

---

## Description

The function computes coefficients of a penalized generalized linear model for normal/binomial/poisson family using modified Jacobi Algorithm for a sequence of lambda values. Currently lasso and elastic net penalty are supported.

## Usage

```
extlasso(x,y,family=c("normal","binomial","poisson"),intercept=TRUE,
normalize=TRUE,tau=1,alpha=1e-12,eps=1e-6,tol=1e-6,maxiter=1e5, nstep=100,min.lambda=1e-4)
```

## Arguments

| | |
|---|---|
| `x` | x is matrix of order n x p where n is number of observations and p is number of predictor variables. Rows should represent observations and columns should represent predictor variables. |
| `y` | y is a vector of response variable of order n x 1. y should follow either normal/binomial/poisson distribution. |
| `family` | family should be one of these: "normal","binomial","poisson" |
| `intercept` | If TRUE, model includes intercept, else the model does not have intercept. |
| `normalize` | If TRUE, columns of x matrix are normalized with mean 0 and norm 1 prior to fitting the model. The coefficients at end are returned on the original scale. Default is normalize = TRUE. |

| | |
|---|---|
| tau | Elastic net parameter, $0 \leq \tau \leq 1$ in elastic net penalty $\lambda\tau\|\|\beta\|\|_1 + (1-\tau)\|\|\beta\|\|_2^2$. Default tau = 1 corresponds to LASSO penalty. |
| alpha | The quantity in approximating $\|\beta_j\| = \sqrt{(}\beta_j^2 + \alpha)$ Default is alpha = 1e-12. |
| eps | A value which is used to set a coefficient to zero if coefficients value is within - eps to + eps. Default is eps = 1e-6. |
| tol | Tolerance criteria for convergence of solutions. Default is tol = 1e-6. |
| maxiter | Maximum number of iterations permissible for solving optimization problem for a particular lambda. Default is 10000. Rarely you need to change this to higher value. |
| nstep | Number of steps from maximum value of lambda to minimum value of lambda. Default is nstep = 100. |
| min.lambda | Minimum value of lambda. Default is min.lambda=1e-4. |

## Value

An object of class 'extlasso' with following components:

| | |
|---|---|
| beta0 | A vector of order nstep of intercept estimates. Each value denote an estimate for a particular lambda. Corresponding lambda values are available in 'lambdas' element of the 'extlasso' object. |
| coef | A matrix of order nstep x p of slope estimates. Each row denotes solution for a particular lambda. Corresponding lambda values are available in 'lambdas' element of the 'extlasso' object. Here p is number of predictor variables. |
| lambdas | Sequence of lambda values for which coefficients are obtained |
| L1norm | L1norm of the coefficients |
| norm.frac | Fractions of norm computed as L1 norm at current lambda divided by maximum L1 norm |
| lambda.iter | Number of iterations used for different lambdas |
| of.value | Objective function values |
| normx | Norm of x variables |

## Author(s)

B N Mandal and Jun Ma

## References

Mandal, B.N. and Jun Ma, (2014). A Jacobi-Armijo Algorithm for LASSO and its Extensions.

## Examples

```
#LASSO
x=matrix(rnorm(100*30),100,30)
y=rnorm(100)
g1=extlasso(x,y,family="normal")
plot(g1)
```

```
plot(g1,xvar="lambda")

#Elastic net
g2=extlasso(x,y,family="normal",tau=0.6)
plot(g2)
plot(g2,xvar="lambda")

#Ridge regression
g3=extlasso(x,y,family="normal",tau=0)
plot(g3)
plot(g3,xvar="lambda")

#L1 penalized GLM for binomial family
x=matrix(rnorm(100*30),100,30)
y=sample(c(0,1),100,replace=TRUE)
g1=extlasso(x,y,family="binomial")
plot(g1)
plot(g1,xvar="lambda")

#Elastic net with GLM with binomial family
g2=extlasso(x,y,family="binomial",tau=0.8)
plot(g2)
plot(g2,xvar="lambda")
```

---

`fusedlasso` *Fused lasso penalized linear regression*

---

## Description

The function computes coefficients of a fused lasso penalized linear regression model using modified Jacobi gradient descent Algorithm for a pair of lambda1 and lambda2 values.

## Usage

```
fusedlasso(x,y,lambda1,lambda2,intercept=TRUE,normalize=TRUE,
alpha=1e-6,eps=1e-6,tol=1e-8,maxiter=1e5)
```

## Arguments

| | |
|---|---|
| `x` | x is a matrix of order n x p where n is number of observations and p is number of predictor variables. Rows should represent observations and columns should represent predictor variables. |
| `y` | y is a vector of response variable of order n x 1. |
| `lambda1` | The value of lambda1 |
| `lambda2` | The value of lambda2 |
| `intercept` | If TRUE, model includes intercept, else the model does not have intercept. |

| | |
|---|---|
| normalize | If TRUE, columns of x matrix are normalized with mean 0 and norm 1 prior to fitting the model. The coefficients at end are returned on the original scale. Default is normalize = TRUE. |
| alpha | The quantity in approximating $\|\beta\| = \sqrt{(}\beta^2 + \alpha)$ Default is alpha = 1e-12. |
| eps | A value which is used to set a coefficient to zero if coefficients value is within - eps to + eps. Default is eps = 1e-6. |
| tol | Tolerance criteria for convergence of solutions. Default is tol = 1e-6. |
| maxiter | Maximum number of iterations permissible for solving optimization problem for a particular lambda. Default is 10000. Rarely you need to change this to higher value. |

## Value

An object of class 'extlasso' with following components:

| | |
|---|---|
| intercept | Value of intercept: TRUE or FALSE as used in input |
| coef | A vector of order (p+1) if intercept is TRUE, first element being estimates of intercept or a vector of order p if intercept is FALSE. Here p is number of predictor variables. |
| lambda1 | The value of lambda1 |
| lambda2 | The value of lambda2 |
| L1norm | L1norm of the coefficients |
| lambda.iter | Number of iterations |
| of.value | Objective function value |

## Author(s)

B N Mandal and Jun Ma

## References

Mandal, B.N. and Jun Ma, (2014). A Jacobi-Armijo Algorithm for LASSO and its Extensions.

## Examples

```
n=50
p=100
rho=0
beta=rep(0,p)
beta[1:20]=1
beta[11:15]=2
beta[25]=3
beta[41:45]=1
x=matrix(rnorm(n*p),n,p)
y=x%*%beta+rnorm(n,0,0.5)
f1<-fusedlasso(x,y,lambda1=0.1,lambda2=1)
plot(beta,col="blue",type="b",pch=1,ylim=range(beta,f1$coef))
lines(f1$coef,type="b",lty=1,col="black")
legend("topright",pch=1,lty=1,merge=TRUE,text.col=c("blue","black"),legend=c("True","Fitted"))
```

plot.extlasso *Plot of regularization path*

## Description

Produces a plot of entire regularization path from a 'extlasso' object obtained using 'extlasso' function.

## Usage

```
## S3 method for class 'extlasso'
plot(x,xvar=c("lambda","L1norm","fraction of norm"),...)
```

## Arguments

| | |
|---|---|
| x | A 'extlasso' object obtained using 'extlasso' function. |
| xvar | What should be on x-axis? xvar="lambda" produces a plot of regularization path with respect to lambda, xvar="L1norm" produces a plot of regularization path with respect to L1 norm of coefficients and xvar="fraction of norm" produces a plot of regularization path with respect to fraction of norm of coefficients. Default is xvar="L1norm". |
| ... | Optional graphical parameters to matplot() function |

## Value

A plot of regularization path is produced.

## Author(s)

B N Mandal and Jun Ma

## References

Mandal, B.N. and Jun Ma, (2014). A Jacobi-Armijo Algorithm for LASSO and its Extensions.

## Examples

```
x=matrix(rnorm(100*30),100,30)
y=rnorm(100)
g1=extlasso(x,y,family="normal")
plot(g1)
plot(g1,xvar="lambda")
x=matrix(rnorm(100*30),100,30)
y=sample(c(0,1),100,replace=TRUE)
g1=extlasso(x,y,family="binomial")
plot(g1)
plot(g1,xvar="lambda")
```

predict.extlasso *Prediction of coefficients of a penalized linear regression or generalized linear models*

## Description

The function computes estimated coefficients value at a given lambda or L1 norm or fraction of norm using a 'extlasso' object obtained using 'extlasso' function.

## Usage

```
## S3 method for class 'extlasso'
predict(object,mode=c("fraction","norm","lambda"),at=0,...)
```

## Arguments

| | |
|---|---|
| object | A 'extlasso' object obtained using 'extlasso' function. |
| mode | If mode="lambda", prediction is made for a given lambda, if mode="norm", prediction is made for a given L1 norm and if mode="fraction", prediction is made for a fraction of norm value. Default is mode="lambda" |
| at | A value at which prediction is to be made. Default is at = 0. |
| ... | Not used. Other arguments to predict. |

## Value

A vector of estimated coefficients of length p or p+1 at the given value of lambda or L1 norm or fraction of norm, depending on intercept=TRUE or FALSE in 'extlasso' object. Here p is number of predictor variables.

## Author(s)

B N Mandal and Jun Ma

## References

Mandal, B.N. and Jun Ma, (2014). A Jacobi-Armijo Algorithm for LASSO and its Extensions.

## Examples

```
x=matrix(rnorm(100*30),100,30)
y=sample(c(0,1),100,replace=TRUE)
g1=extlasso(x,y,family="binomial")
predict(g1,mode="lambda",at=0.1)
predict(g1,mode="L1norm",at=1)
predict(g1,mode="fraction",at=0.5)
x=matrix(rnorm(100*30),100,30)
y=rnorm(100)
g1=extlasso(x,y,family="normal")
```

```
predict(g1,mode="lambda",at=0.09)
predict(g1,mode="L1norm",at=0.6)
predict(g1,mode="fraction",at=0.8)
```

# Index