Package: Libra (via r-universe)

August 26, 2024

Type Package

Title Linearized Bregman Algorithms for Generalized Linear Models

Version 1.7

Date 2022-4-9

Author Feng Ruan, Jiechao Xiong and Yuan Yao

Maintainer Jiechao Xiong <xiongjiechao@pku.edu.cn>

Depends R (>= 3.0), nnls

Suggests lars, MASS, igraph

SystemRequirements GNU Scientific Library (GSL)

Description Efficient procedures for fitting the regularization path for linear, binomial, multinomial, Ising and Potts models with lasso, group lasso or column lasso(only for multinomial) penalty. The package uses Linearized Bregman Algorithm to solve the regularization path through iterations. Bregman Inverse Scale Space Differential Inclusion solver is also provided for linear model with lasso penalty.

License GPL-2

URL https://arxiv.org/abs/1406.7728

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-04-11 09:22:30 UTC

RoxygenNote 7.1.2

Contents

`cv.iss` *CV for ISS*

Description

Cross-validation method to tuning the parameter t for ISS.

Usage

```
cv.iss(
  X,
  y,
  K = 5,
  t,
  intercept = TRUE,
  normalize = TRUE,
  plot.it = TRUE,
  se = TRUE,
  ...
)
```

Arguments

`X`	An n-by-p matrix of predictors
`y`	Response Variable
`K`	Folds number for CV. Default is 5.
`t`	A vector of predecided tuning parameter.
`intercept`	If TRUE, an intercept is included in the model (and not penalized), otherwise no intercept is included. Default is TRUE.
`normalize`	if TRUE, each variable is scaled to have L2 norm square-root n. Default is TRUE.
`plot.it`	Plot it? Default is TRUE
`se`	Include standard error bands? Default is TRUE
`...`	Additonal arguments passing to lb

Details

K-fold cross-validation method is used to tuning the parameter t for ISS. Mean square error is used as prediction error.

Value

A list is returned. The list contains a vector of parameter t, crossvalidation error cv.error, and the estimated standard deviation for it cv.sd

Author(s)

Feng Ruan, Jiechao Xiong and Yuan Yao

References

Ohser, Ruan, Xiong, Yao and Yin, Sparse Recovery via Differential Inclusions, https://arxiv.org/abs/1406.7728

Examples

```
#Examples in the reference paper
library(MASS)
n = 200;p = 100;k = 30;sigma = 1
Sigma = 1/(3*p)*matrix(rep(1,p^2),p,p)
diag(Sigma) = 1
A = mvrnorm(n, rep(0, p), Sigma)
u_ref = rep(0,p)
supp_ref = 1:k
u_ref[supp_ref] = rnorm(k)
u_ref[supp_ref] = u_ref[supp_ref]+sign(u_ref[supp_ref])
b = as.vector(A%*%u_ref + sigma*rnorm(n))
cv.iss(A,b,intercept = FALSE,normalize = FALSE)
```

`cv.lb` *CV for lb*

Description

Cross-validation method to tuning the parameter t for lb.

Usage

```
cv.lb(
  X,
  y,
  kappa,
  alpha,
```

```
  K = 5,
  tlist,
  nt = 100,
  trate = 100,
  family = c("gaussian", "binomial", "multinomial"),
  group = FALSE,
  intercept = TRUE,
  normalize = TRUE,
  plot.it = TRUE,
  se = TRUE,
  ...
)
```

Arguments

X	An n-by-p matrix of predictors
y	Response Variable
kappa	The damping factor of the Linearized Bregman Algorithm that is defined in the reference paper. See details.
alpha	Parameter in Linearized Bregman algorithm which controls the step-length of the discretized solver for the Bregman Inverse Scale Space. See details.
K	Folds number for CV. Default is 5.
tlist	Parameters t along the path.
nt	Number of t. Used only if tlist is missing. Default is 100.
trate	tmax/tmin. Used only if tlist is missing. Default is 100.
family	Response type
group	Whether to use a group penalty, Default is FALSE.
intercept	If TRUE, an intercept is included in the model (and not penalized), otherwise no intercept is included. Default is TRUE.
normalize	if TRUE, each variable is scaled to have L2 norm square-root n. Default is TRUE.
plot.it	Plot it? Default is TRUE
se	Include standard error bands? Default is TRUE
...	Additonal arguments passing to lb

Details

K-fold cross-validation method is used to tuning the parameter t for ISS. Mean square error is used for linear model. Miss-classification error is used for binomial and multinomial model.

Value

A list is returned. The list contains a vector of parameter t, crossvalidation error cv.error, and the estimated standard deviation for it cv.sd

Author(s)

Feng Ruan, Jiechao Xiong and Yuan Yao

References

Ohser, Ruan, Xiong, Yao and Yin, Sparse Recovery via Differential Inclusions, https://arxiv.org/abs/1406.7728

Examples

```
#Examples in the reference paper
library(MASS)
n = 200;p = 100;k = 30;sigma = 1
Sigma = 1/(3*p)*matrix(rep(1,p^2),p,p)
diag(Sigma) = 1
A = mvrnorm(n, rep(0, p), Sigma)
u_ref = rep(0,p)
supp_ref = 1:k
u_ref[supp_ref] = rnorm(k)
u_ref[supp_ref] = u_ref[supp_ref]+sign(u_ref[supp_ref])
b = as.vector(A%*%u_ref + sigma*rnorm(n))
cv.lb(A,b,10,1/20,intercept = FALSE,normalize = FALSE)

#Simulated data, binomial case
X <- matrix(rnorm(500*100), nrow=500, ncol=100)
alpha <- c(rep(1,30), rep(0,70))
y <- 2*as.numeric(runif(500)<1/(1+exp(-X %*% alpha)))-1
cv.lb(X,y,kappa=5,alpha=1,family="binomial",
             intercept=FALSE,normalize = FALSE)
```

---

diabetes *Blood and other measurements in diabetics*

Description

The diabetes data frame has 442 rows and 3 columns. These are the data used in the Efron et al "Least Angle Regression" paper.

Format

This data frame contains the following columns:

x a matrix with 10 columns

y a numeric vector

x2 a matrix with 64 columns

Details

The x matrix has been standardized to have unit L2 norm in each column and zero mean. The matrix x2 consists of x plus certain interactions.

References

Efron, Hastie, Johnstone and Tibshirani (2003) "Least Angle Regression" (with discussion) *Annals of Statistics*

`ggm` *Linearized Bregman solver for composite conditionally likelihood of Gaussian Graphical model with lasso penalty.*

Description

Solver for the entire solution path of coefficients.

Usage

```
ggm(
  X,
  kappa,
  alpha,
  S = NA,
  c = 2,
  tlist,
  nt = 100,
  trate = 100,
  print = FALSE
)
```

Arguments

`X`	An n-by-p matrix of variables.
`kappa`	The damping factor of the Linearized Bregman Algorithm that is defined in the reference paper. See details.
`alpha`	Parameter in Linearized Bregman algorithm which controls the step-length of the discretized solver for the Bregman Inverse Scale Space. See details.
`S`	The covariance matrix can be provided directly if data matrix X is missing.
`c`	Normalized step-length. If alpha is missing, alpha is automatically generated by `alpha=c*n/(kappa*||X^T*X||_2)`. Default is 2. It should be in (0,4). If beyond this range the path may be oscillated at large t values.
`tlist`	Parameters t along the path.
`nt`	Number of t. Used only if tlist is missing. Default is 100.
`trate`	tmax/tmin. Used only if tlist is missing. Default is 100.
`print`	If TRUE, the percentage of finished computation is printed.

Details

The data matrix X is assumed to follow the Gaussian Graohical model which is described as following:

$$X \sim N(\mu, \Theta^{-1})$$

where Θ is sparse p-by-p symmetric matrix. Then conditional on x_{-j}

$$x_j \sim N(\mu_j - \sum_{k \neq j} \Theta_{jk}/\Theta_{jj}(x_k - \mu_k), 1/\Theta_{jj})$$

then the composite conditional likelihood is like this:

$$-\sum_j condloglik(X_j|X_{-j})$$

or in detail:

$$\sum_j \Theta_j^T S\Theta_j/2\Theta_{jj} - ln(\Theta_{jj})/2$$

where S is covariance matrix of data. It is easy to prove that this loss function is convex.

Value

A "ggm" class object is returned. The list contains the call, the path, value for alpha, kappa, t.

Author(s)

Jiechao Xiong

Examples

```
library(MASS)
p = 20
Omega = diag(1,p,p)
Omega[0:(p-2)*(p+1)+2] = 1/3
Omega[1:(p-1)*(p+1)] = 1/3
S = solve(Omega)
X = mvrnorm(n=500,rep(0,p),S)
obj = ggm(X,10,trate=10)
obj$path[,,50]
```

`ising` *Linearized Bregman solver for composite conditionally likelihood of Ising model with lasso penalty.*

Description

Solver for the entire solution path of coefficients.

Usage

```
ising(
  X,
  kappa,
  alpha,
  c = 2,
  tlist,
  responses = c(-1, 1),
  nt = 100,
  trate = 100,
  intercept = TRUE,
  print = FALSE
)
```

Arguments

`X`	An n-by-p matrix of variables.
`kappa`	The damping factor of the Linearized Bregman Algorithm that is defined in the reference paper. See details.
`alpha`	Parameter in Linearized Bregman algorithm which controls the step-length of the discretized solver for the Bregman Inverse Scale Space. See details.
`c`	Normalized step-length. If alpha is missing, alpha is automatically generated by `alpha=c*n/(kappa*||X^T*X||_2)`. Default is 2. It should be in (0,4). If beyond this range the path may be oscillated at large t values.
`tlist`	Parameters t along the path.
`responses`	The type of data. c(0,1) or c(-1,1), Default is c(-1,1).
`nt`	Number of t. Used only if tlist is missing. Default is 100.
`trate`	tmax/tmin. Used only if tlist is missing. Default is 100.
`intercept`	if TRUE, an intercept is included in the model (and not penalized), otherwise no intercept is included. Default is TRUE.
`print`	If TRUE, the percentage of finished computation is printed.

Details

The data matrix X is assumed in {1,-1}. The Ising model here used is described as following:

$$P(x) \sim \exp(\sum_i \frac{a_{0i}}{2} x_i + x^T \Theta x/4)$$

where Θ is p-by-p symmetric and 0 on diagnal. Then conditional on x_{-j}

$$\frac{P(x_j = 1)}{P(x_j = -1)} = exp(\sum_i a_{0i} + \sum_{i \neq j} \theta_{ji} x_i)$$

then the composite conditional likelihood is like this:

$$-\sum_j condloglik(X_j | X_{-j})$$

Value

A "ising" class object is returned. The list contains the call, the path, the intercept term a0 and value for alpha, kappa, t.

Author(s)

Jiechao Xiong

Examples

```
library('Libra')
library('igraph')
data('west10')
X <- as.matrix(2*west10-1);
obj = ising(X,10,0.1,nt=1000,trate=100)
g<-graph.adjacency(obj$path[,,770],mode="undirected",weighted=TRUE)
E(g)[E(g)$weight<0]$color<-"red"
E(g)[E(g)$weight>0]$color<-"green"
V(g)$name<-attributes(west10)$names
plot(g,vertex.shape="rectangle",vertex.size=35,vertex.label=V(g)$name,
edge.width=2*abs(E(g)$weight),main="Ising Model (LB): sparsity=0.51")
```

---

isingdata *Simulation data for Ising model*

Description

The `isingdata` data list contains 2 variables. One is 5000 samples from the ising model on the 10-by-10 grid using Gibbs sampling. The other is the groupdtruth parameter.

Format

This data list contains the following two variables:

J a 100-by-100 matrix, the groudtruth of ising model.

X a 5000-by-100 matrix, each entry is in -1,1.

`iss` *ISS solver for linear model with lasso penalty*

Description

Solver for the entire solution path of coefficients for ISS.

Usage

```
iss(X, y, intercept = TRUE, normalize = TRUE, nvar = min(dim(X)))
```

Arguments

`X`	An n-by-p matrix of predictors
`y`	Response Variable
`intercept`	if TRUE, an intercept is included in the model (and not penalized), otherwise no intercept is included. Default is TRUE.
`normalize`	if normalize, each variable is scaled to have L2 norm square-root n. Default is TRUE.
`nvar`	Maximal number of variables allowed in the model.

Details

The ISS solver computes the whole regularization path for lasso-penalty for linear model. It gives the piecewise constant solution path for Bregman Inverse Scale Space Differential Inclusion. It is the asymptotic limit of LB method with kaapa goes to infinity and alpha goes to zero.

Value

An "LB" class object is returned. The list contains the call, the family, the path, the intercept term a0 and value for alpha, kappa, iter, and meanvalue, scale factor of X, meanx and normx.

Author(s)

Feng Ruan, Jiechao Xiong and Yuan Yao

References

Ohser, Ruan, Xiong, Yao and Yin, Sparse Recovery via Differential Inclusions, https://arxiv.org/abs/1406.7728

Examples

```
#Examples in the reference paper
library(MASS)
library(lars)
library(MASS)
library(lars)
n = 80;p = 100;k = 30;sigma = 1
Sigma = 1/(3*p)*matrix(rep(1,p^2),p,p)
diag(Sigma) = 1
A = mvrnorm(n, rep(0, p), Sigma)
u_ref = rep(0,p)
supp_ref = 1:k
u_ref[supp_ref] = rnorm(k)
u_ref[supp_ref] = u_ref[supp_ref]+sign(u_ref[supp_ref])
b = as.vector(A%*%u_ref + sigma*rnorm(n))
lasso = lars(A,b,normalize=FALSE,intercept=FALSE,max.steps=100)
par(mfrow=c(3,2))
matplot(n/lasso$lambda, lasso$beta[1:100,], xlab = bquote(n/lambda),
     ylab = "Coefficients", xlim=c(0,3),ylim=c(range(lasso$beta)),type='l', main="Lasso")
object = iss(A,b,intercept=FALSE,normalize=FALSE)
plot(object,xlim=c(0,3),main=bquote("ISS"))
kappa_list = c(4,16,64,256)
alpha_list = 1/10/kappa_list
for (i in 1:4){
  object <- lb(A,b,kappa_list[i],alpha_list[i],family="gaussian",group=FALSE,
               trate=20,intercept=FALSE,normalize=FALSE)
  plot(object,xlim=c(0,3),main=bquote(paste("LB ",kappa,"=",.(kappa_list[i]))))
}
```

`lb` *Linearized Bregman solver for linear, binomial, multinomial models with lasso, group lasso or column lasso penalty.*

Description

Solver for the entire solution path of coefficients for Linear Bregman iteration.

Usage

```
lb(
  X,
  y,
  kappa,
  alpha,
  c = 1,
  tlist,
  nt = 100,
  trate = 100,
```

```
  family = c("gaussian", "binomial", "multinomial"),
  group = FALSE,
  index = NA,
  intercept = TRUE,
  normalize = TRUE,
  print = FALSE
)
```

Arguments

X	An n-by-p matrix of predictors
y	Response Variable
kappa	The damping factor of the Linearized Bregman Algorithm that is defined in the reference paper. See details.
alpha	Parameter in Linearized Bregman algorithm which controls the step-length of the discretized solver for the Bregman Inverse Scale Space. See details.
c	Normalized step-length. If alpha is missing, alpha is automatically generated by alpha=n*c/(kappa*||X^T*X||_2). It should be in (0,2) for family = "gaussian"(Default is 1), (0,8) for family = "binomial"(Default is 4), (0,4) for family = "multinomial"(Default is 2). If beyond these range the path may be oscillated at large t values.
tlist	Parameters t along the path.
nt	Number of t. Used only if tlist is missing. Default is 100.
trate	tmax/tmin. Used only if tlist is missing. Default is 100.
family	Response type
group	Whether to use a group penalty, Default is FALSE.
index	For group models, the index is a vector that determines the group of the parameters. Parameters of the same group should have equal value in index. Be careful that multinomial group model default assumes the variables in same column are in the same group, and a empty value of index means each variable is a group.
intercept	if TRUE, an intercept is included in the model (and not penalized), otherwise no intercept is included. Default is TRUE.
normalize	if TRUE, each variable is scaled to have L2 norm square-root n. Default is TRUE.
print	If TRUE, the percentage of finished computation is printed.

Details

The Linearized Bregman solver computes the whole regularization path for different types of lasso-penalty for gaussian, binomial and multinomial models through iterations. It is the Euler forward discretized form of the continuous Bregman Inverse Scale Space Differential Inclusion. For binomial models, the response variable y is assumed to be a vector of two classes which is transformed in to {1,-1}. For the multinomial models, the response variable y can be a vector of k classes or a n-by-k matrix that each entry is in {0,1} with 1 indicates the class. Under all circumstances, two parameters, kappa and alpha need to be specified beforehand. The definitions of kappa and alpha are the same as that defined in the reference paper. Parameter alpha is defined as stepsize and kappa is the damping factor of the Linearized Bregman Algorithm that is defined in the reference paper.

Value

A "lb" class object is returned. The list contains the call, the type, the path, the intercept term a0 and value for alpha, kappa, iter, and meanvalue, scale factor of X, meanx and normx. For gaussian and bonomial, path is a p-by-nt matrix, and for multinomial, path is a k-by-p-by-nt array, each dimension represents class, predictor and parameter t.

Author(s)

Feng Ruan, Jiechao Xiong and Yuan Yao

References

Ohser, Ruan, Xiong, Yao and Yin, Sparse Recovery via Differential Inclusions, https://arxiv.org/abs/1406.7728

Examples

```
#Examples in the reference paper
library(MASS)
n = 80;p = 100;k = 30;sigma = 1
Sigma = 1/(3*p)*matrix(rep(1,p^2),p,p)
diag(Sigma) = 1
A = mvrnorm(n, rep(0, p), Sigma)
u_ref = rep(0,p)
supp_ref = 1:k
u_ref[supp_ref] = rnorm(k)
u_ref[supp_ref] = u_ref[supp_ref]+sign(u_ref[supp_ref])
b = as.vector(A%*%u_ref + sigma*rnorm(n))
kappa = 16
alpha = 1/160
object <- lb(A,b,kappa,alpha,family="gaussian",group=FALSE,
             trate=20,intercept=FALSE,normalize=FALSE)
plot(object,xlim=c(0,3),main=bquote(paste("LB ",kappa,"=",.(kappa))))


#Diabetes, linear case
library(Libra)
data(diabetes)
attach(diabetes)
object <- lb(x,y,100,1e-3,family="gaussian",group=FALSE)
plot(object)
detach(diabetes)

#Simulated data, binomial case
data('west10')
y<-2*west10[,1]-1;
X<-as.matrix(2*west10[,2:10]-1);
path <- lb(X,y,kappa = 1,family="binomial",trate=100,normalize = FALSE)
plot(path,xtype="norm",omit.zeros=FALSE)

#Simulated data, multinomial case
```

```
X <- matrix(rnorm(500*100), nrow=500, ncol=100)
alpha <- matrix(c(rnorm(30*3), rep(0,70*3)),nrow=3)
P <- exp(alpha%*%t(X))
P <- scale(P,FALSE,apply(P,2,sum))
y <- rep(0,500)
rd <- runif(500)
y[rd<P[1,]] <- 1
y[rd>1-P[3,]] <- -1
result <- lb(X,y,kappa=5,alpha=0.1,family="multinomial",
 group=TRUE,intercept=FALSE,normalize = FALSE)
plot(result)
```

`plot.lb` *Plot method for lb objects*

Description

Produce a plot of an LB fit. The default is a complete coefficient path.

Usage

```
## S3 method for class 'lb'
plot(x, xtype = c("t", "norm"), omit.zeros = TRUE, eps = 1e-10, ...)
```

Arguments

`x`	lb object
`xtype`	The x-axis type. "t" or "norm". Default is "t".
`omit.zeros`	When the number of variables is much greater than the number of observations, many coefficients will never be nonzero; this logical (default `TRUE`) avoids plotting these zero coefficents
`eps`	Definition of zero above, default is `1e-10`
`...`	Additonal arguments for generic plot. Can be used to set xlims, change colors, line widths, etc

Details

The default plot uses the fraction of L1 norm as the x. For multinomial case, the sum of absolute values of different class's coefficients are caculated to represent each variable. The intercept term is not ploted

Author(s)

Feng Ruan, Jiechao Xiong and Yuan Yao

potts *Linearized Bregman solver for composite conditionally likelihood of Potts model with lasso penalty and block-group penalty.*

Description

Solver for the entire solution path of coefficients.

Usage

```
potts(
  X,
  kappa,
  alpha,
  c = 1,
  tlist,
  nt = 100,
  trate = 100,
  group = FALSE,
  intercept = TRUE,
  print = FALSE
)
```

Arguments

X	An n-by-p matrix of variables.
kappa	The damping factor of the Linearized Bregman Algorithm that is defined in the reference paper. See details.
alpha	Parameter in Linearized Bregman algorithm which controls the step-length of the discretized solver for the Bregman Inverse Scale Space. See details.
c	Normalized step-length. If alpha is missing, alpha is automatically generated by alpha=c*n/(kappa*||XX^T*XX||_2), where XX is 0-1 indicator matrix induced by the class of each Xi. Default is 1. It should be in (0,2). If beyond this range the path may be oscillated at large t values.
tlist	Parameters t along the path.
nt	Number of t. Used only if tlist is missing. Default is 100.
trate	tmax/tmin. Used only if tlist is missing. Default is 100.
group	Whether to use a block-wise group penalty, Default is FALSE
intercept	if TRUE, an intercept is included in the model (and not penalized), otherwise no intercept is included. Default is TRUE.
print	If TRUE, the percentage of finished computation is printed.

Details

The data matrix X is transformed into a 0-1 indicator matrix D with each column D_{jk} means $1(X_j) == k$. The Potts model here used is described as following:

$$P(x) \sim \exp(\sum_{jk} a_{0,jk} 1(x_i = 1) + d^T \Theta d/2)$$

where Θ is p-by-p symmetric and 0 on diagnal. Then conditional on x_{-j}

$$P(x_j = k) \sim exp(\sum_{k} a_{0,jk} + \sum_{i \neq j, r} \theta_{jk,ir} d_{ir})$$

then the composite conditional likelihood is like this:

$$-\sum_{j} condloglik(X_j | X_{-j})$$

Value

A "potts" class object is returned. The list contains the call, the path, the intercept term a0 and value for alpha, kappa, t.

Author(s)

Jiechao Xiong

Examples

```
X = matrix(floor(runif(200*10)*3),200,10)
obj = potts(X,10,nt=100,trate=10,group=TRUE)
```

`predict.lb` *Predict method for lb objects*

Description

Predict response variable for new data given a lb object

Usage

```
## S3 method for class 'lb'
predict(object, newx, t, type = c("fit", "coefficients"), ...)
```

Arguments

object	lb object
newx	New data matrix that each row is a data or a vector. If missing, type switched to coefficients
t	The parmeter for object to determin which coeffiecients used for prediction. Linear interpolation is used if t is not in object\$t. If missing, all the coeffiecients along the path is used to predict.
type	To predict response of newx or just fit coeffients on the path.
...	Additonal arguments for generic predict.

Details

The default plot uses the fraction of L1 norm as the x. For multinomial case, the sum of absolute values of different class's coefficients are caculated to represent each variable. The intercept term is not ploted

Value

A list containing t and other variables. For type="fit", the rediction response "fit" is returned. For "binomial", a vector of the probabilities for newx falling into class +1 is redurned. For "multinomial", a matrix with each column means the probabilities for newx falling into the corresponding class. If type="coefficients" coefficients "beta" and intercepts "a0" are returned.

Author(s)

Feng Ruan, Jiechao Xiong and Yuan Yao

---

west10 *Journey to the West, one of the Four Great Classical Novels of Chinese Literature by WU, Cheng'en.*

Description

The west10 data frame has 408 rows and 10 columns. It records the appearance of top 10 characters in 408 scenes, the characters who appeared ("1") or not ("0") in each of the scenes. The dataset was collected via crowdsourcing in the classes of Mathematical Introduction to Data Analysis and Statistical Learning, taught by Prof. Yuan YAO at Peking University. he original data contains all the 302 characters, which can be downloaded at the sourse website.

Format

This file contains the following data frame:

west10 a data frame of 408 rows and 10 columns

Details

It records the appearance of top 10 characters in 408 scenes, the characters who appeared ("1") or not ("0") in each of the scenes. All the character names are in Chinese Pinyin.

References

Yuan YAO, A Mathematical Introduction to Data Analysis, Lecture Notes in *Peking University*

Index