Package 'BayesRecipe'

November 20, 2023

Type Package

Title Bayesian Reciprocal Regularization

Version 0.1

Date 2019-10-11

Authors Himel Mallick (Merck Research Laboratories) <himel.mallick@merck.com>, Rahim Alhamzawi (University of Al-Qadisiyah) <rahim.alhamzawi@qu.edu.iq>, Vladimir Svetnik (Merck Research Laboratories) <vladimir_svetnik@merck.com>

Maintainer Himel Mallick <himel.mallick@merck.com>

Description Implementation of Bayesian reciprocal regularization for regression and classification.

Depends R (>= 3.5.0), msm, miscTools, glmnet, timeSeries, Bolstad2, mombf, coda, MCMCglmm, VGAM, truncdist, mvtnorm, SuppDists, MCMCpack, MASS, bayeslm

License MIT

LazyData TRUE

RoxygenNote 7.2.3

R topics documented:

BayesRLasso 1
BayesRLassoSlice 4
BayesRLassoSMN 7
BayesRLassoSMU 9
Goodrich 12
rrLASSO.S5 12
TGFB 14

Index **15**

`BayesRLasso` *Bayesian Reciprocal Regularization*

Description

Wrapper for Bayesian Reciprocal LASSO using MCMC

Usage

```
BayesRLasso(
  x,
  y,
  method = "slice",
  lambda.estimate = "AP",
  update.sigma2 = TRUE,
  max.steps = 11000,
  n.burn = 1000,
  n.thin = 1,
  ridge.CV = TRUE,
  a = 0.001,
  b = 0.001,
  posterior.summary.beta = "mean",
  posterior.summary.lambda = "median",
  beta.ci.level = 0.95,
  lambda.ci.level = 0.95,
  seed = 1234,
  na.rm = TRUE
)
```

Arguments

x	A numeric matrix with standardized predictors in columns and samples in rows.
y	A mean-centered continuous response variable with matching rows with x.
method	If Gibbs sampling is desired, 'SMU' or 'SMN' must be selected as the data augmentation scheme or mixture representation, although not recommended when the data size is as big as hundreds of covariates. By default it uses an ellipitical slice sampler ('slice') which is scalable for large-scale problems.
lambda.estimate	Estimating lambda by empirical bayes ('EB'), MCMC ('MCMC'), or apriori ('AP'). Default is 'AP' (currently only available option when method = 'slice').
update.sigma2	Whether sigma2 should be updated. Default is TRUE (currently only available option when method = 'slice').
max.steps	Number of MCMC iterations. Default is 11000.
n.burn	Number of burn-in iterations. Default is 1000.
n.thin	Lag at which thinning should be done. Default is 1 (no thinning).
ridge.CV	When method = 'SMU' and X is rank deficient, a ridge parameter is added to the diagonal of the crossproduct (XtX) to allow for proper calculation of the inverse. If TRUE, the ridge parameter is estimated by cross-validation using glmnet. Otherwise, it falls back to adding a small number (1e-05) without the cross-validation. Default is TRUE. This parameter is ignored when method = 'SMN' or 'slice'.
a	If lambda.estimate = 'MCMC', shape hyperparameter for the Gamma prior on lambda. Default is 0.001. This parameter is ignored when method = 'slice'.
b	If lambda.estimate = 'MCMC', rate hyperparameter for the Gamma prior on lambda. Default is 0.001. This parameter is ignored when method = 'slice'.
posterior.summary.beta	Posterior summary measure for beta (mean, median, or mode). Default is 'mean'.

`posterior.summary.lambda`
: Posterior summary measure for lambda (mean, median, or mode). Default is 'median'.

`beta.ci.level`
: Credible interval level for beta. Default is 0.95 (95%).

`lambda.ci.level`
: Credible interval level for lambda. Default is 0.95 (95%).

`seed`
: Seed value for reproducibility. Default is 1234.

`na.rm`
: Logical. Should missing values (including NaN) be omitted from the calculations?

Value

A list containing the following components is returned:

`time`
: Computational time in minutes.

`beta`
: Posterior estimates of beta.

`lowerbeta`
: Lower limit of the credible interval of beta.

`upperbeta`
: Upper limit of the credible interval of beta.

`lambda`
: Posterior estimate of lambda.

`lambdaci`
: Posterior credible interval of lambda.

`beta.post`
: Post-burn-in posterior samples of beta.

`sigma2.post`
: Post-burn-in posterior samples of sigma2.

`lambda.post`
: Post-burn-in posterior samples of lambda.

Examples

```
## Not run:

#########################
# Load Prostate dataset #
#########################

library(ElemStatLearn)
prost<-prostate

########################################
# Scale data and prepare train/test split #
########################################

prost.std <- data.frame(cbind(scale(prost[,1:8]),prost$lpsa))
names(prost.std)[9] <- 'lpsa'
data.train <- prost.std[prost$train,]
data.test <- prost.std[!prost$train,]

################################
# Extract standardized variables #
################################

y.train   = data.train$lpsa - mean(data.train$lpsa)
y.test <- data.test$lpsa - mean(data.test$lpsa)
x.train = scale(as.matrix(data.train[,1:8], ncol=8))
x.test = scale(as.matrix(data.test[,1:8], ncol=8))
```

```
############################
# Reciprocal Bayesian LASSO #
############################

fit_BayesRLasso<- BayesRLasso(x.train, y.train)
y.pred.BayesRLasso<-x.test%*%fit_BayesRLasso$beta
mean((y.pred.BayesRLasso - y.test)^2) # Performance on test data

####################################
# Visualization of Posterior Samples #
####################################

##############
# Trace Plot #
##############

library(coda)
plot(mcmc(fit_BayesRLasso$beta.post),density=FALSE,smooth=TRUE)

#############
# Histogram #
#############

library(psych)
multi.hist(fit_BayesRLasso$beta.post,density=TRUE,main="")


## End(Not run)
```

BayesRLassoSlice *Bayesian Reciprocal Regularization*

Description

Elliptical slice sampler for Bayesian Reciprocal LASSO using inverse Laplace prior

Usage

```
BayesRLassoSlice(
  x,
  y,
  lambda.estimate = "AP",
  update.sigma2 = TRUE,
  max.steps = 11000,
  n.burn = 1000,
  n.thin = 1,
  ridge.CV = TRUE,
  a = 0.001,
  b = 0.001,
  posterior.summary.beta = "mean",
  posterior.summary.lambda = "median",
```

```
  beta.ci.level = 0.95,
  lambda.ci.level = 0.95,
  seed = 1234,
  na.rm = TRUE
)
```

Arguments

x	A numeric matrix with standardized predictors in columns and samples in rows.
y	A mean-centered continuous response variable with matching rows with x.
lambda.estimate	Estimating lambda by empirical bayes ('EB'), MCMC ('MCMC'), or apriori ('AP'). Default is 'AP'.
update.sigma2	Whether sigma2 should be updated. Default is TRUE.
max.steps	Number of MCMC iterations. Default is 11000.
n.burn	Number of burn-in iterations. Default is 1000.
n.thin	Lag at which thinning should be done. Default is 1 (no thinning).
ridge.CV	If X is rank deficient, a ridge parameter is added to the diagonal of the crossproduct (XtX) to allow for proper calculation of the inverse. If TRUE, the ridge parameter is estimated by cross-validation using glmnet. Otherwise, it falls back to adding a small number (1e-05) without the cross-validation. Default is TRUE.
a	If lambda.estimate = 'MCMC', shape hyperparameter for the Gamma prior on lambda. Default is 0.001.
b	If lambda.estimate = 'MCMC', rate hyperparameter for the Gamma prior on lambda. Default is 0.001.
posterior.summary.beta	Posterior summary measure for beta (mean, median, or mode). Default is 'mean'.
posterior.summary.lambda	Posterior summary measure for lambda (mean, median, or mode). Default is 'median'.
beta.ci.level	Credible interval level for beta. Default is 0.95 (95%).
lambda.ci.level	Credible interval level for lambda. Default is 0.95 (95%).
seed	Seed value for reproducibility. Default is 1234.
na.rm	Logical. Should missing values (including NaN) be omitted from the calculations?

Value

A list containing the following components is returned:

time	Computational time in minutes.
beta	Posterior estimates of beta.
lowerbeta	Lower limit of the credible interval of beta.
upperbeta	Upper limit of the credible interval of beta.
lambda	Posterior estimate of lambda.
lambdaci	Posterior credible interval of lambda.
beta.post	Post-burn-in posterior samples of beta.
sigma2.post	Post-burn-in posterior samples of sigma2.
lambda.post	Post-burn-in posterior samples of lambda.

Examples

```
## Not run:

#########################
# Load Prostate dataset #
#########################

library(ElemStatLearn)
prost<-prostate

########################################
# Scale data and prepare train/test split #
########################################

prost.std <- data.frame(cbind(scale(prost[,1:8]),prost$lpsa))
names(prost.std)[9] <- 'lpsa'
data.train <- prost.std[prost$train,]
data.test <- prost.std[!prost$train,]

#################################
# Extract standardized variables #
#################################

y.train   = data.train$lpsa - mean(data.train$lpsa)
y.test <- data.test$lpsa - mean(data.test$lpsa)
x.train = scale(as.matrix(data.train[,1:8], ncol=8))
x.test = scale(as.matrix(data.test[,1:8], ncol=8))

#################################
# Reciprocal Bayesian LASSO (slice) #
#################################

fit_BayesRLasso_slice<- BayesRLasso(x.train, y.train, method = 'slice')
y.pred.BayesRLasso_slice<-x.test%*%fit_BayesRLasso_slice$beta
mean((y.pred.BayesRLasso_slice - y.test)^2) # Performance on test data

#####################################
# Visualization of Posterior Samples #
#####################################

##############
# Trace Plot #
##############

library(coda)
plot(mcmc(fit_BayesRLasso_slice$beta.post),density=FALSE,smooth=TRUE)

#############
# Histogram #
#############

library(psych)
multi.hist(fit_BayesRLasso_slice$beta.post,density=TRUE,main="")


## End(Not run)
```

BayesRLassoSMN *Bayesian Reciprocal Regularization*

Description

MCMC for Bayesian Reciprocal LASSO using truncated SMN mixture

Usage

```
BayesRLassoSMN(
  x,
  y,
  lambda.estimate = "AP",
  update.sigma2 = TRUE,
  max.steps = 11000,
  n.burn = 1000,
  n.thin = 1,
  ridge.CV = TRUE,
  a = 0.001,
  b = 0.001,
  posterior.summary.beta = "mean",
  posterior.summary.lambda = "median",
  beta.ci.level = 0.95,
  lambda.ci.level = 0.95,
  seed = 1234,
  na.rm = TRUE
)
```

Arguments

x	A numeric matrix with standardized predictors in columns and samples in rows.
y	A mean-centered continuous response variable with matching rows with x.
lambda.estimate	Estimating lambda by empirical bayes ('EB'), MCMC ('MCMC'), or apriori ('AP'). Default is 'AP'.
update.sigma2	Whether sigma2 should be updated. Default is TRUE.
max.steps	Number of MCMC iterations. Default is 11000.
n.burn	Number of burn-in iterations. Default is 1000.
n.thin	Lag at which thinning should be done. Default is 1 (no thinning).
ridge.CV	If X is rank deficient, a ridge parameter is added to the diagonal of the crossproduct (XtX) to allow for proper calculation of the inverse. If TRUE, the ridge parameter is estimated by cross-validation using glmnet. Otherwise, it falls back to adding a small number (1e-05) without the cross-validation. Default is TRUE
a	If lambda.estimate = 'MCMC', shape hyperparameter for the Gamma prior on lambda. Default is 0.001.
b	If lambda.estimate = 'MCMC', rate hyperparameter for the Gamma prior on lambda. Default is 0.001.

`posterior.summary.beta`
: Posterior summary measure for beta (mean, median, or mode). Default is 'mean'.

`posterior.summary.lambda`
: Posterior summary measure for lambda (mean, median, or mode). Default is 'median'.

`beta.ci.level` Credible interval level for beta. Default is 0.95 (95%).

`lambda.ci.level`
: Credible interval level for lambda. Default is 0.95 (95%).

`seed` Seed value for reproducibility. Default is 1234.

`na.rm` Logical. Should missing values (including NaN) be omitted from the calculations?

Value

A list containing the following components is returned:

`time`	Computational time in minutes.
`beta`	Posterior estimates of beta.
`lowerbeta`	Lower limit of the credible interval of beta.
`upperbeta`	Upper limit of the credible interval of beta.
`lambda`	Posterior estimate of lambda.
`lambdaci`	Posterior credible interval of lambda.
`beta.post`	Post-burn-in posterior samples of beta.
`sigma2.post`	Post-burn-in posterior samples of sigma2.
`lambda.post`	Post-burn-in posterior samples of lambda.

Examples

```
## Not run:

#########################
# Load Prostate dataset #
#########################

library(ElemStatLearn)
prost<-prostate

########################################
# Scale data and prepare train/test split #
########################################

prost.std <- data.frame(cbind(scale(prost[,1:8]),prost$lpsa))
names(prost.std)[9] <- 'lpsa'
data.train <- prost.std[prost$train,]
data.test <- prost.std[!prost$train,]

################################
# Extract standardized variables #
################################

y.train   = data.train$lpsa - mean(data.train$lpsa)
y.test <- data.test$lpsa - mean(data.test$lpsa)
```

```
x.train = scale(as.matrix(data.train[,1:8], ncol=8))
x.test = scale(as.matrix(data.test[,1:8], ncol=8))

###################################
# Reciprocal Bayesian LASSO (SMN) #
###################################

fit_BayesRLasso_SMN<- BayesRLasso(x.train, y.train, method = 'SMN')
y.pred.BayesRLasso_SMN<-x.test%*%fit_BayesRLasso_SMN$beta
mean((y.pred.BayesRLasso_SMN - y.test)^2) # Performance on test data

#####################################
# Visualization of Posterior Samples #
#####################################

##############
# Trace Plot #
##############

library(coda)
plot(mcmc(fit_BayesRLasso_SMN$beta.post),density=FALSE,smooth=TRUE)

#############
# Histogram #
#############

library(psych)
multi.hist(fit_BayesRLasso_SMN$beta.post,density=TRUE,main="")


## End(Not run)
```

BayesRLassoSMU *Bayesian Reciprocal Regularization*

Description

MCMC for Bayesian Reciprocal LASSO using inverse SMU mixture

Usage

```
BayesRLassoSMU(
  x,
  y,
  lambda.estimate = "AP",
  update.sigma2 = TRUE,
  max.steps = 11000,
  n.burn = 1000,
  n.thin = 1,
  ridge.CV = TRUE,
  a = 0.001,
  b = 0.001,
  posterior.summary.beta = "mean",
```

```
  posterior.summary.lambda = "median",
  beta.ci.level = 0.95,
  lambda.ci.level = 0.95,
  seed = 1234,
  na.rm = TRUE
)
```

Arguments

x	A numeric matrix with standardized predictors in columns and samples in rows.
y	A mean-centered continuous response variable with matching rows with x.
lambda.estimate	Estimating lambda by empirical bayes ('EB'), MCMC ('MCMC'), or apriori ('AP'). Default is 'AP'.
update.sigma2	Whether sigma2 should be updated. Default is TRUE.
max.steps	Number of MCMC iterations. Default is 11000.
n.burn	Number of burn-in iterations. Default is 1000.
n.thin	Lag at which thinning should be done. Default is 1 (no thinning).
ridge.CV	If X is rank deficient, a ridge parameter is added to the diagonal of the crossproduct (XtX) to allow for proper calculation of the inverse. If TRUE, the ridge parameter is estimated by cross-validation using glmnet. Otherwise, it falls back to adding a small number (1e-05) without the cross-validation. Default is TRUE.
a	If lambda.estimate = 'MCMC', shape hyperparameter for the Gamma prior on lambda. Default is 0.001.
b	If lambda.estimate = 'MCMC', rate hyperparameter for the Gamma prior on lambda. Default is 0.001.
posterior.summary.beta	Posterior summary measure for beta (mean, median, or mode). Default is 'mean'.
posterior.summary.lambda	Posterior summary measure for lambda (mean, median, or mode). Default is 'median'.
beta.ci.level	Credible interval level for beta. Default is 0.95 (95%).
lambda.ci.level	Credible interval level for lambda. Default is 0.95 (95%).
seed	Seed value for reproducibility. Default is 1234.
na.rm	Logical. Should missing values (including NaN) be omitted from the calculations?

Value

A list containing the following components is returned:

time	Computational time in minutes.
beta	Posterior estimates of beta.
lowerbeta	Lower limit of the credible interval of beta.
upperbeta	Upper limit of the credible interval of beta.
lambda	Posterior estimate of lambda.
lambdaci	Posterior credible interval of lambda.

beta.post	Post-burn-in posterior samples of beta.
sigma2.post	Post-burn-in posterior samples of sigma2.
lambda.post	Post-burn-in posterior samples of lambda.

Examples

```
## Not run:

########################
# Load Prostate dataset #
########################

library(ElemStatLearn)
prost<-prostate

########################################
# Scale data and prepare train/test split #
########################################

prost.std <- data.frame(cbind(scale(prost[,1:8]),prost$lpsa))
names(prost.std)[9] <- 'lpsa'
data.train <- prost.std[prost$train,]
data.test <- prost.std[!prost$train,]

#################################
# Extract standardized variables #
#################################

y.train   = data.train$lpsa - mean(data.train$lpsa)
y.test <- data.test$lpsa - mean(data.test$lpsa)
x.train = scale(as.matrix(data.train[,1:8], ncol=8))
x.test = scale(as.matrix(data.test[,1:8], ncol=8))

#################################
# Reciprocal Bayesian LASSO (SMU) #
#################################

fit_BayesRLasso_SMU<- BayesRLasso(x.train, y.train, method = 'SMU')
y.pred.BayesRLasso_SMU<-x.test%*%fit_BayesRLasso_SMU$beta
mean((y.pred.BayesRLasso_SMU - y.test)^2) # Performance on test data

#####################################
# Visualization of Posterior Samples #
#####################################

##############
# Trace Plot #
##############

library(coda)
plot(mcmc(fit_BayesRLasso_SMU$beta.post),density=FALSE,smooth=TRUE)

#############
# Histogram #
#############
```

```
library(psych)
multi.hist(fit_BayesRLasso_SMU$beta.post,density=TRUE,main="")


## End(Not run)
```

`Goodrich` *Microbiome taxonomy dataset from Zhou and Gallins (2019).*

Description

16S genus-level abundances of 99 OTUs related to BMI in n = 414 human subjects, as described in Goodrich et al. (2014).

Usage

```
data(Goodrich)
```

Format

A data frame of taxonomic abundances of 99 OTUs in n = 414 human subjects. From a pool of 11,225 candidate features, OTUs were collapsed to the genus level by summing their respective relative abundances, discarding any OTUs which were unannotated at the genus level.

References

Zhou Y, Gallins D (2019). A Review and Tutorial of Machine Learning Methods for Microbiome Host Trait Prediction. Frontiers in Genetics 10, 579.

Goodrich JK et al. (2014). Human Genetics Shape the Gut Microbiome. Cell 159(4): 789-799.

`rrLASSO.S5` *Bayesian Reciprocal Regularization*

Description

Simplified Shotgun Stochastic Search with Screening (S5) algorithm to solve the reduced reciprocal LASSO, based on a source code adapted from 'BayesS5' by Minsuk Shin.

Usage

```
rrLASSO.S5(
  X,
  y,
  S = 30,
  lam = 1,
  intercept = TRUE,
  verbose = FALSE,
  seed = 1234
)
```

Arguments

X	A numeric matrix with standardized predictors in columns and samples in rows.
y	A mean-centered continuous response variable with matching rows with X.
S	A screening size of variables. Default is 30.
lam	A tuning parameter for the rrLASSO objective function. Default is 1.
intercept	If TRUE, intercept is included in the final OLS fit. The default is TRUE.
verbose	If TRUE, the function prints the current status of the S5 in each temperature. The default is FALSE.
seed	Seed value for reproducibility. Default is 1234.

Value

A list containing the following components is returned:

hppm	Index of the highest posterior probablity model.
marg.prob	Marginal posterior probablities of the coefficients.
beta	OLS coefficients from the final rLASSO model.
time	Computation time in seconds.

Examples

```
## Not run:

#########################
# Load Prostate dataset #
#########################

library(ElemStatLearn)
prost<-prostate

########################################
# Scale data and prepare train/test split #
########################################

prost.std <- data.frame(cbind(scale(prost[,1:8]),prost$lpsa))
names(prost.std)[9] <- 'lpsa'
data.train <- prost.std[prost$train,]
data.test <- prost.std[!prost$train,]

#################################
# Extract standardized variables #
#################################

y.train   = data.train$lpsa - mean(data.train$lpsa)
y.test <- data.test$lpsa - mean(data.test$lpsa)
x.train = scale(as.matrix(data.train[,1:8], ncol=8))
x.test = scale(as.matrix(data.test[,1:8], ncol=8))

#################################
# Reduced Reciprocal LASSO (S5) #
#################################

rrLasso<- rrLASSO.S5(x.train, y.train)
```

```
y.pred.rrLasso<-x.test%*%rrLasso$beta[-1]
mean((y.pred.rrLasso - y.test)^2) # Performance on test data


## End(Not run)
```

TGFB *TGFB gene expression colon cancer dataset from Calon et al. (2012).*

Description

Gene expression profiles of 172 genes related to TGF-beta in n = 262 human colon cancer patients. TGF-beta is one of the key cytokines implicated in various disease processes including colon cancer.

Usage

```
data(TGFB)
```

Format

A data frame of gene expression profiles of 172 genes related to TGF-beta in n = 262 human patients and the amount of TGF-beta production measured in each patient (y).

References

Rossell D, Telesca D (2017). Nonlocal Priors for High-Dimensional Estimation. Journal of the American Statistical Association 112(517): 254-265.

Calon et al. (2012). Dependency of Colorectal Cancer On A TGF-beta-driven Program in Stromal Cells for Metastasis Initiation. Cancer cell 22(5): 571-584.

Index

∗ **Bayesian**
BayesRLasso, 1
BayesRLassoSlice, 4
BayesRLassoSMN, 7
BayesRLassoSMU, 9
rrLASSO.S5, 12
∗ **Ellipitical**
BayesRLasso, 1
BayesRLassoSlice, 4
∗ **LASSO,**
rrLASSO.S5, 12
∗ **LASSO**
BayesRLasso, 1
BayesRLassoSlice, 4
BayesRLassoSMN, 7
BayesRLassoSMU, 9
∗ **MCMC,**
BayesRLasso, 1
BayesRLassoSlice, 4
BayesRLassoSMN, 7
BayesRLassoSMU, 9
∗ **Reciprocal**
BayesRLasso, 1
BayesRLassoSlice, 4
BayesRLassoSMN, 7
BayesRLassoSMU, 9
rrLASSO.S5, 12
∗ **S5**
rrLASSO.S5, 12
∗ **SMN,**
BayesRLasso, 1
BayesRLassoSMN, 7
∗ **SMU,**
BayesRLasso, 1
BayesRLassoSMU, 9
∗ **datasets**
Goodrich, 12
TGFB, 14
∗ **regularization,**
BayesRLasso, 1
BayesRLassoSlice, 4
BayesRLassoSMN, 7
BayesRLassoSMU, 9
rrLASSO.S5, 12
∗ **sampler,**
BayesRLasso, 1
BayesRLassoSlice, 4
∗ **slice**
BayesRLasso, 1
BayesRLassoSlice, 4

BayesRLasso, 1
BayesRLassoSlice, 4
BayesRLassoSMN, 7
BayesRLassoSMU, 9

Goodrich, 12

rrLASSO.S5, 12

TGFB, 14