Package ‘CARlasso’

October 12, 2022

Title  Conditional Autoregressive LASSO
Version  0.1.2
Maintainer  Yunyi Shen <yshen99@wisc.edu>
Depends  R (>= 3.5.0)
License  GPL-3
Encoding  UTF-8
LazyData  true
ByteCompile  TRUE
Suggests  testthat, knitr, rmarkdown
RoxygenNote  7.1.1
LinkingTo  Rcpp, RcppArmadillo, RcppProgress
Imports  Rcpp, coda, Matrix, igraph, ggraph, ggplot2, MASS, methods
SystemRequirements  C++11
VignetteBuilder  knitr
URL  https://github.com/YunyiShen/CAR-LASSO
BugReports  https://github.com/YunyiShen/CAR-LASSO/issues
NeedsCompilation  yes
Author  Yunyi Shen [aut, cre] (<https://orcid.org/0000-0003-2779-6507>), Claudia Solis-Lemus [aut] (<https://orcid.org/0000-0002-9789-8915>)
Repository  CRAN
Date/Publication  2021-08-11 19:00:02 UTC
R topics documented:

- bGlasso ................................................. 2
- CARlasso ............................................. 4
- horseshoe ............................................. 6
- mgp154 ............................................... 7
- mgp2592 .............................................. 8
- plot.bglasso_out ................................... 8
- plot.carlasso_out ................................... 9
- rCARlasso_ .......................................... 9
- rCARlasso_ .......................................... 11
- simu_AR1 ............................................. 12

Index 14

bGlasso

Gibbs sampler for Bayesian Graphical LASSO and extensions

Description

Main sampling algorithm of Glasso model, note that the mean is in CAR parameterization

Usage

bGlasso(
  data,
  link = "identity",
  r_Omega = 1,
  delta_Omega = 0.01,
  n_iter = 2000,
  n_burn_in = 1000,
  thin_by = 10,
  ns = 1000,
  m = 20,
  emax = 64,
  progress = TRUE,
  verbos = TRUE
)

Arguments

data A data.frame with all response, row as observations
link String name of link function? Currently can be "identity" for normal response, "probit" for binary, "log" for counting, "logit" for compositional. Note that when use "logit", the last response will be used as reference.
r_Omega Hyper-parameter for precision matrix, shape parameter of Gamma. Should be a scalar
delta_Omega  Hyper-parameter for precision matrix, rate parameter of Gamma. Should be a scalar
n_iter   Number of sampling iterations (i.e. after burn in) for the Gibbs sampler
n_burn_in Number of burn in iterations for the Gibbs sampler
thin_by  Final sample was thin by this number
ns       parameter for ARS, maximum number of hulls, only used when link is "log" and "logit"
m       parameter for ARS, initial number of hulls, only used when link is "log" and "logit"
emax     parameter for ARS, tolerance for small values being 0, larger meaning we tolerate smaller values, only used when link is "log" and "logit"
progress   Bool, whether report progress from C++
verbos    Bool, whether show warnings and messages.

Value

A bglasso_out object with elements:

- $point_est
  - $Omega: Posterior mean of precision matrix
- $nodes
  - $responses: node name of responses
- $data
  - $response: response matrix
- $settings: all settings sent to the algorithm, exclude data
- $MCMC_output
  - $mu: A coda::mcmc object, each row was an MCMC sample of the mean vector
  - $Omega: A coda::mcmc object, each row was an MCMC sample of the upper triangular part (with diagonal) of precision matrix Omega
  - $lambda: A coda::mcmc object, first column was the shrinkage parameter lambda for regression coefficient and the second column was shrinkage parameter lambda for precision matrix

Examples

    set.seed(42)
    dt <- simu_AR1()
    glassores <- bGlasso(data = dt[,1:5])
    plot(glassores)
CARlasso  

*Gibbs sampler for Conditional Autoregressive LASSO and extensions*

### Description

Main sampling algorithm of CAR-LASSO model

### Usage

```r
CARlasso(
  formula,
  data,
  link = "identity",
  adaptive = FALSE,
  r_beta = ifelse(adaptive, 0.01, 1),
  delta_beta = ifelse(adaptive, 1e-06, 0.01),
  r_Omega = ifelse(adaptive, 0.01, 1),
  delta_Omega = ifelse(adaptive, 1e-06, 0.01),
  lambda_diag = 0,
  n_iter = 2000,
  n_burn_in = 1000,
  thin_by = 10,
  ns = 1000,
  m = 20,
  emax = 64,
  progress = TRUE,
  verbos = TRUE
)
```

### Arguments

- **formula**: A double sided formula with response at left hand side and predictors at right hand side
- **data**: A data.frame with all response and predictors, row as observations
- **link**: String name of link function? Currently can be "identity" for normal response, "probit" for binary, "log" for counting, "logit" for compositional. Note that when use "logit", the last response will be used as reference.
- **adaptive**: Bool, whether run the adaptive version of the model
- **r_beta**: Hyper-parameter for regression coefficient, shape parameter of Gamma, if adaptive, should have row number same as number of predictors while column number of responses
- **delta_beta**: Hyper-parameter for regression coefficient, rate parameter of Gamma, if adaptive, should have row number same as number of predictors while column number of responses
CARlasso

\[ r_{\Omega} \]
Hyper-parameter for precision matrix, shape parameter of Gamma. If adaptive, can be a matrix with same size as precision matrix, if this is the case, only upper triangular part without diagonal will be used, or can be a vector whose size was the upper triangular part of precision matrix, if non-adaptive, a scalar.

\[ \delta_{\Omega} \]
Hyper-parameter for precision matrix, rate parameter of Gamma. If adaptive, can be a matrix with same size as precision matrix, if this is the case, only upper triangular part without diagonal will be used, or can be a vector whose size was the upper triangular part of precision matrix, if non-adaptive, a scalar.

\[ \lambda_{\text{diag}} \]
adaptive only hyper-parameter for penalties on diagonal entries of Omega, should have dimension \( k \) and non-negative

\[ n_{\text{iter}} \]
Number of sampling iterations (i.e. after burn in) for the Gibbs sampler

\[ n_{\text{burn in}} \]
Number of burn in iterations for the Gibbs sampler

\[ \text{thin by} \]
Final sample was thin by this number

\[ ns \]
parameter for ARS, maximum number of hulls, only used when link is "log" and "logit"

\[ m \]
parameter for ARS, initial number of hulls, only used when link is "log" and "logit"

\[ \text{emax} \]
parameter for ARS, tolerance for small values being 0, larger meaning we tolerate smaller values, only used when link is "log" and "logit"

\[ \text{progress} \]
Bool, whether report progress from C++

\[ \text{verbos} \]
Bool, whether show warnings and messages.

**Value**

A carlasso_out object with elements:

- **$\text{point_est}**
  - $\Omega$: Posterior mean of precision matrix
  - $\beta$: Posterior mean of regression coefficient
  - $\text{CAR}$
    - $\text{C}$: The conditional regression coefficients among responses
    - $\text{B}$: The conditional regression coefficients between response and predictors
    - $\text{M}$: The conditional variance
- **$\text{nodes}**
  - $\text{responses}$: node name of responses
  - $\text{predictors}$: node name of predictors
- **$\text{data}**
  - $\text{response}$: response matrix
  - $\text{design}$: design matrix
- **$\text{settings}$**: all settings sent to the algorithm, exclude data
- **$\text{MCMC_output}**
  - $\beta$: A coda::mcmc object, each row was an MCMC sample of the (column) vectorization of regression coefficient B
- \( \mu \): A coda::mcmc object, each row was an MCMC sample of the mean vector
- \( \Omega \): A coda::mcmc object, each row was an MCMC sample of the upper triangular part (with diagonal) of precision matrix \( \Omega \)
- \( \lambda \): **Non-adaptive only**, A coda::mcmc object, first column was the shrinkage parameter \( \lambda \) for regression coefficients and the second column was shrinkage parameter \( \lambda \) for precision matrix
- \( \lambda_{\beta} \): **Adaptive only**, A coda::mcmc object, each row was an MCMC sample of the (column) vectorization of shrinkage parameter for regression coefficient \( B \)
- \( \lambda_{\Omega} \): **Adaptive only**, A coda::mcmc object, each row was an MCMC sample of the shrinkage parameter for the upper triangular part (without diagonal) of precision matrix \( \Omega \)

Examples

```r
set.seed(42)
dt <- simu_AR1()
car_res <- CARlasso(y1 + y2 + y3 + y4 + y5 ~ x1 + x2 + x3 + x4 + x5, data = dt, adaptive = TRUE)
plot(car_res, tol = 0.05)
# with horseshoe inference

horseshoe(car_res)
plot(car_res)
```

---

**horseshoe**

*Horseshoe method for graphical structure inference*

**Description**

Horseshoe method for graphical structure inference

**Usage**

```r
horseshoe(obj, Bbar = NULL, A = NULL, nu = 3, V = NULL, thr = 0.5)
```

**Arguments**

- `obj`: The carlasso_out object from CARlasso
- `Bbar`: Prior mean of regression coefficients, default all 0s
- `A`: Prior precision of regression coefficients, default 1e-8
- `nu`: Prior degree of freedom of the Wishart on precision matrix
- `V`: prior covariance matrix of the Wishart on precision matrix
- `thr`: threshold for horseshoe inference, default 0.5
Details

This method fits a linear regression with less informative prior on any parameters and compare the posterior mean with the LASSO result. If LASSO is comparably less than result without sparsity prior, we argue that the edge should be absent.

Value

A carlasso_out object with learned binary adjacency matrix and multi-response linear regression MCMC output.

Examples

```r
set.seed(42)
dt <- simu_AR1()
car_res <- CARlasso(y1+y2+y3+y4+y5~x1+x2+x3+x4+x5, data = dt, adaptive = TRUE)
car_res <- horseshoe(car_res)
plot(car_res)
```

Description

This study is based on pyrosequencing of 16S rDNA amplicons from faecal samples collected from 178 elderly Irish citizens and 13 healthy young control subjects. A subset of these samples were also subjected to shotgun sequencing using Illumina HiSeq 2000 2x91bp reads. Antibiotic treatment was an exclusion criterion.

Usage

```r
data(mgp154)
```

Format

An data.frame with genus and predictors.

Source

MG-RAST:mgp154

References

**mgp2592**  
*Hofmockel Soil Aggregate COB KBASE*

**Description**

This study is to examine soil microbial community composition and structure of both bacteria and fungi at a microbiologically-relevant scale. The researchers isolated soil aggregates from three land management systems in central Iowa to test if the aggregate-level microbial responses are related to plant community and management practices. The clean dataset has 120 samples with 17 genus under consideration.

**Usage**

```r
data(mgp2592)
```

**Format**

A data frame with genus and predictors.

**Source**

MG-RAST-mgp2592

**References**


---

**plot.bglasso_out**

*plot the graph estimated by graphical lasso with threshold method using ggraph*

**Description**

plot the graph estimated by graphical lasso with threshold method using ggraph

**Usage**

```r
## S3 method for class 'bglasso_out'
plot(x, ...)
```

**Arguments**

- `x` - The bglasso_out
- `...`
  - `tol`: threshold for plotting default 0.01, if horseshoe, then horseshoe result is used
plot.carlasso_out

Value
A ggplot xect

plot.carlasso_out  plot the chain graph estimated by CAR-LASSO with threshold or horseshoe method using ggraph

Description
plot the chain graph estimated by CAR-LASSO with threshold or horseshoe method using ggraph

Usage
## S3 method for class 'carlasso_out'
plot(x, ...)

Arguments
x The carlasso_out xect
...
  • tol: threshold for plotting default 0.01, if horseshoe, then horseshoe result is used

Value
A ggplot xect

rCARAlasso_ Block Gibbs sampler for adaptive CAR-LASSO

Description
This function is for advanced users to build their own sampler use adaptive CARlasso as core. It will execute one round of Gibbs sampler of adaptive CAR-LASSO model. Be aware that the function is a void function implemented in C++, and all updated parameters e.g. Omega will be manipulate directly in memory to save space. Users should manage to do their own work to save the state. Also be aware that R uses shallow copy by default, which means one cannot save the state by simply give it to another object e.g. first Omega_old <- Omega_curr then update Omega_curr. Omega_old will also change. This function will NOT check dimensions of input. Below we assume n samples, k responses and p predictors.
Usage

rCARAlasso_(
  Z_curr,
  design,
  lambda2_beta,
  tau2_curr,
  beta_curr,
  lambda_Omega,
  Omega_curr,
  mu_curr,
  r_beta,
  delta_beta,
  r_Omega,
  delta_Omega,
  lambda_diag,
  k,
  p,
  n
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z_curr</td>
<td>the current (latent) normal Z_curr, should be n*k. Will not be changed</td>
</tr>
<tr>
<td>design</td>
<td>the design matrix, should be n*p. Will not be changed</td>
</tr>
<tr>
<td>lambda2_beta</td>
<td>the current shrinkage parameter of regression coefficients, should be a vector with p*k entries. Will be updated</td>
</tr>
<tr>
<td>tau2_curr</td>
<td>the current latent scale parameter in the normal mixture representation of Laplace, for regression coefficients, should be a vector with p*k entries. Will be updated.</td>
</tr>
<tr>
<td>beta_curr</td>
<td>the current regression coefficients, should be a matrix sized p*k (p row and k columns). Will be updated.</td>
</tr>
<tr>
<td>lambda_Omega</td>
<td>the current shrinkage parameter for Omega, should be a vector with k*(k-1)/2 entries. Will be updated.</td>
</tr>
<tr>
<td>Omega_curr</td>
<td>the current Omega matrix, should be a matrix of size k*k. Will be updated.</td>
</tr>
<tr>
<td>mu_curr</td>
<td>the current mu, intercept, should be a vector of size k. Will be updated.</td>
</tr>
<tr>
<td>r_beta</td>
<td>hyperprior's parameter of shrinkage for regression coefficients, should be a scalar of type 'double' and positive. Will not be updated.</td>
</tr>
<tr>
<td>delta_beta</td>
<td>hyperprior's parameter of shrinkage for regression coefficients, should be a scalar of type 'double' and positive. Will not be updated.</td>
</tr>
<tr>
<td>r_Omega</td>
<td>hyperprior's parameter of shrinkage for precision Omega, should be a scalar of type 'double' and positive. Will not be updated.</td>
</tr>
<tr>
<td>delta_Omega</td>
<td>hyperprior's parameter of shrinkage for precision Omega, should be a scalar of type 'double' and positive. Will not be updated.</td>
</tr>
<tr>
<td>lambda_diag</td>
<td>shrinkage parameter of the diagonal of Omega, should be a vector of size k, should be non-negative. Will not be updated.</td>
</tr>
</tbody>
</table>
rCARlasso_  

k  integer, number of responses  
p  integer, number of predictors  
n  integer, number of Z_curr points  

Value  

Again this is a void function and will not return anything. All update happened in memory directly.  

rCARlasso_  

Block Gibbs sampler for CAR-LASSO  

Description  

This function is for advanced users to build their own sampler use CARlasso as core. It will execute one round of Gibbs sampler of CAR-LASSO model. Be aware that the function is a void function implemented in C++, and all updated parameters e.g. Omega will be manipulate directly in memory to save space. Users should manage to do their own work to save the state. Also be aware that R uses shallow copy by default, which means one cannot save the state by simply give it to another object e.g. first Omega_old <- Omega_curr then update Omega_curr, Omega_old will also change. This function will NOT check dimensions of input. Below we assume n samples, k responses and p predictors.  

Usage  

rCARlasso_(  
Z_curr,  
design,  
lambda2_beta,  
tau2_curr,  
beta_curr,  
lambda_Omega,  
Omega_curr,  
mu_curr,  
r_beta,  
delta_beta,  
r_Omega,  
delta_Omega,  
k,  
p,  
n  
)  

Arguments  

Z_curr  the current (latent) normal data, should be n*k. Will not be changed  
design  the design matrix, should be n*p. Will not be changed
lambda2_beta: the current shrinkage parameter of regression coefficients, should be a scalar of type double. Will be updated.

tau2_curr: the current latent scale parameter in the normal mixture representation of Laplace, for regression coefficients, should be a vector with p*k entries. Will be updated.

beta_curr: the current regression coefficients, should be a matrix sized p*k (p row and k columns). Will be updated.

lambda_Omega: the current shrinkage parameter for Omega, should be a scalar of type double. Will be updated.

Omega_curr: the current Omega matrix, should be a matrix of size k*k. Will be updated.

mu_curr: the current mu, intercept, should be a vector of size k. Will be updated.

r_beta: hyperprior’s parameter of shrinkage for regression coefficients, should be a scalar of type ‘double’ and positive. Will not be updated.

delta_beta: hyperprior’s parameter of shrinkage for regression coefficients, should be a scalar of type ‘double’ and positive. Will not be updated.

r_Omega: hyperprior’s parameter of shrinkage for precision Omega, should be a scalar of type ‘double’ and positive. Will not be updated.

delta_Omega: hyperprior’s parameter of shrinkage for precision Omega, should be a scalar of type ‘double’ and positive. Will not be updated.

k: integer, number of responses

p: integer, number of predictors

n: integer, number of data points

Value

Again this is a void function and will not return anything. All update happened in memory directly.

---

**simu_AR1**

*Simulate a simple AR1 model with specific predictor*

---

**Description**

Simulate a simple AR1 model with specific predictor

**Usage**

`simu_AR1(n = 100, k = 5, rho = 0.7)`

**Arguments**

- **n**: sample size
- **k**: number of responses
- **rho**: partial correlation in AR1
**Details**

Simulate a simple AR1 model with k responses and k predictors, each predictor has effect on exact one response node

**Value**

a dataframe, with y1 to yk as responses and x1 to xk as predictors
Index

* datasets
  mgp154, 7
  mgp2592, 8

bGlasso, 2

CARlasso, 4

horseshoe, 6

mgp154, 7
mgp2592, 8

plot.bglasso_out, 8
plot.carlasso_out, 9

rCARAlasso, 9
rCARlasso, 11

simu_AR1, 12