Package ‘dlbayes’

November 14, 2018

Type Package

Title Use Dirichlet Laplace Prior to Solve Linear Regression Problem and Do Variable Selection

Version 0.1.0

Author Shijia Zhang; Meng Li

Maintainer Shijia Zhang <zsj27@mail.ustc.edu.cn>

Description The Dirichlet Laplace shrinkage prior in Bayesian linear regression and variable selection, featuring:
utility functions in implementing Dirichlet-Laplace priors such as visualization;
scalability in Bayesian linear regression;
penalized credible regions for variable selection.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports GIGrvg, expm, glmnet, MASS, LaplacesDemon, stats, graphics

RoxygenNote 6.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-14 09:20:09 UTC

R topics documented:

dl *Implement the Dirichlet Laplace shrinkage prior in Bayesian linear regression*

Description

This function is the baysian linear regression version of the algorithm proposed in Bhattacharya et al. (2015). The function is fast because we use fast sampling method compute posterior samples. The method proposed in Bhattacharya et al. (2015) is used in the second step perfectly solving the large p problem. The local shrinkage controlling parameter psi_j are updated via a slice sampling scheme given by Polson et al. (2014). And the parameters phi_j have various inverse gaussian distribution. We generate variates with transformation into multiple roots by Michael et al. (1976).

Usage

```
dl(x, y, burn = 5000, nmc = 5000, thin = 1, hyper = 1/2)
```

Arguments

`x`	input matrix, each row is an observation vector, dimension n*p.
`y`	Response variable, a n*1 vector.
`burn`	Number of burn-in MCMC samples. Default is 5000.
`nmc`	Number of posterior draws to be saved. Default is 5000.
`thin`	Thinning parameter of the chain. Default is 1 means no thinning.
`hyper`	The value of hyperparameter in the prior, can be [1/max(n,p),1/2]. It controls local shrinkage scales through psi. Small values of hyperparameter would lead most of the result close to zero; while large values allow small singularity at zero. We give a method and a function to tuning this parameter. See the function called "dlhyper" for details.

Value

`betamatrix`	Posterior samples of beta. A large matrix (nmc/thin)*p

Examples

```
{
p=50
n=5
#generate x
x=matrix(rnorm(n*p),nrow=n)
#generate beta
beta=c(rep(0,10),runif(n=5,min=-1,max=1),rep(0,10),runif(n=5,min=-1,max=1),rep(0,p-30))
#generate y
y=x%*%beta+rnorm(n)
hyper=dlhyper(x,y)
dlresult=dl(x,y,hyper=hyper)}
```

dlanalysis *dlanalysis*

Description

This is a function that analyse the MCMC sampling result by computing the posterior mean, median and credible intervals

Usage

```
dlanalysis(dlresult, alpha = 0.05)
```

Arguments

`dlresult`	Posterior samples of beta. A large matrix (nmc/thin)*p
`alpha`	Level for the credible intervals. For example,the default is alpha = 0.05 means 95% credible intervals

Value

`betamean`	Posterior mean of beta, a p*1 vector.
`LeftCI`	The left bounds of the credible intervals.
`RightCI`	The right bounds of the credible intervals.
`betamedian`	Posterior median of Beta, a p*1 vector.

Examples

```
p=50
n=5
#generate x
x=matrix(rnorm(n*p),nrow=n)
#generate beta
beta=c(rep(0,10),runif(n=5,min=-1,max=1),rep(0,10),runif(n=5,min=-1,max=1),rep(0,p-30))
#generate y
y=x%*%beta+rnorm(n)
hyper=dlhyper(x,y)
dlresult=dl(x,y,hyper=hyper)
da=dlanalysis(dlresult,alpha=0.05)
da$betamean
da$betamedian
da$LeftCI
da$RightCI
```

dlhyper *Tune the hyperparameter in the prior distribtuion*

Description

This function is to tune the value of hyperparameter in the prior, which can be [1/max(n,p),1/2]. We use the method proposed by Zhang et al. (2018). This method tune the hyperparameter by incorporating a prior on R^2. And they give a direct way to minimize KL directed divergence for special condition.

Usage

```
dlhyper(x, y)
```

Arguments

x	input matrix, each row is an observation vector, dimension n*p. Same as the argument in dlmain
y	Response variable, a n*1 vector. Same as the argument in dlmain

Value

hyper	A value that can use in the following posterior computation

Examples

```
p=50
n=6
#generate x
x=matrix(rnorm(n*p),nrow=n)
#generate beta
beta=c(rep(0,10),runif(n=5,min=-1,max=1),rep(0,10),runif(n=5,min=-1,max=1),rep(0,p-30))
#generate y
y=x%*%beta+rnorm(n)
hyper=dlhyper(x,y)
```

dlprior *Title Simulate the dirichlet laplace shrinkage prior*

Description

This function generates random deviates from dirichlet laplace shrinkage prior and can plot the distribution function.

Usage

```
dlprior(hyper = 1/2, p = 1e+05, plt = TRUE, min = -5, max = 5,
  sigma = 1)
```

Arguments

`hyper`	important hyperparameter that related to posterior shrinkage scales and prior distribution
`p`	number of observations
`plt`	whether to plot the dirichlet laplace prior. default TRUE means plot the distribution
`min`	left point of the plot graph
`max`	right point of the plot graph
`sigma`	the value equals to normal noises' standard deviations

Value

`beta`	A p*1 vector. p observations from the distribution

Examples

```
{theta=dlprior(hyper=1/2,p=100000,plt=TRUE,min=-5,max=5,sigma=1)}
```

`dlvs` *Title Do Bayesian variable selection via penalized credible region*

Description

This is a function using the algorithm doing variable selection via penalized credible interval proposed by Bondell et al. (2012). The computation of the proposed sequence is doing matrix computing and using existing LASSO software.

Usage

```
dlvs(dlresult)
```

Arguments

`dlresult`	Posterior samples of beta. A large matrix (nmc/thin)*p

Value

`betatil`	Variable selection result of beta, a p*1 vector. Most of the values shrinks to 0

Examples

```
{
p=30
n=5
#generate x
x=matrix(rnorm(n*p),nrow=n)
#generate beta
beta=c(rep(0,10),runif(n=5,min=-1,max=1),rep(0,10),runif(n=5,min=-1,max=1))
#generate y
y=x%*%beta+rnorm(n)
hyper=dlhyper(x,y)
dlresult=dl(x,y,hyper=hyper)
dlvs(dlresult)
}
```

Index