Package ‘MUSS’

January 2, 2023

Title Spike and Slab Variable Selector under Matrix Uncertainty

Version 1.0.0

Author Shuyu Guo [aut, cre] <guo_sy@outlook.com>,
Bo Ning [aut, ctb] <bycning@ucdavis.edu>

Description In high-dimensional sparse regression with measurement error in variables, assuming the errors are Normally distributed with mean 0 and errors for each variable have a specific variance, then 'MUSS' can conduct variable selection for such case. 'MUSS' selects variables by spike and slab priors for the regression coefficients. It works under EM framework, where the unknown true design matrix and indicators of spike-slab priors are treated as latent variables. To avoid the effect of inappropriate choices of spike and slab scale parameters on variable selection, the final output coefficients are obtained following a decreasing sequence of spike parameters and the path can be displayed by functions from 'MUSS' package.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

LinkingTo Rcpp, RcppEigen

Imports Rcpp, RcppEigen, ggplot2, reshape2, stats, MASS

Suggests knitr, rmarkdown, SSLASSO, glmnet

VignetteBuilder knitr

R topics documented:

gPlot *gPlot*

Description

Plot maximized $g(\boldsymbol{\beta}, \theta, \sigma | \boldsymbol{\beta^{(t)}}, \theta^{(t)}, \sigma^{(t)})$ over iterations at a specified spike parameter. The function g is a concave lower bound for objective log-likelihood function at $\boldsymbol{\beta^{(t)}}$, $\theta^{(t)}$ and $\sigma^{(t)}$ by Jensen's Inequality. $\boldsymbol{\beta^{(t+1)}}$, $\theta^{(t+1)}$ and $\sigma^{(t+1)}$ are obtained by maximizing $g(\boldsymbol{\beta}, \theta, \sigma | \boldsymbol{\beta^{(t)}}, \theta^{(t)}, \sigma^{(t)})$. See Vignette for more details.

Usage

```
gPlot(fit_obj, spike_param)
```

Arguments

`fit_obj`	Fitted object from function 'MUSS'.
`spike_param`	The value of the spike parameter to be specified, which must be in `spike_params` designated in 'MUSS' function. Each `spike_param` corresponds to a unique gPlot. If default `spike_params` is used in function 'MUSS', one can check the value of `spike_params` by calling $spike_params from the returned object from function 'MUSS'.

MUSS *Spike and Slab Variable Selector under Matrix Uncertainty*

Description

For high dimensional sparse regression with additive errors in potential variables, Spike and Slab Variable Selector under Matrix Uncertainty(MUSS) selects variables under EM framework by treating both unknown true variables and spike-slab indicators as latent variables. Following a decreasing list of spike scale parameters, a path of regression coefficients is obtained.

Usage

```
MUSS(
  Z,
  y,
  tauList,
  beta_prior_type = "Laplacian",
  spike_params,
  slab_param,
  beta_init,
  sigma_update = TRUE,
  sigma_init,
  theta_init = 0.5,
  return_g = FALSE,
  a = 1,
  b,
  omega = 1,
  kappa = 1,
  tolerance = 0.01,
  max_iter = 500
)
```

Arguments

Z
$n * p$ covariate matrix with additive error in each column (possibly $p > n$). It is assumed the measurement error model takes the form $Z = X + \Xi$, where X is the unknown true design matrix and Ξ is the matrix of i.i.d Gaussian measurement error of mean 0 and same variance within each column.

y
Numeric response vector from n observations.

tauList
Vector of length p. Corresponding variances of additive measurement error for p columns.

beta_prior_type
Character. Type of prior distribution of regression parameter β. beta_prior_type = "Laplacian" or "Gaussian". Default is "Laplacian".

spike_params
Vector of of length L. Decreasing scale parameters of spike prior for beta. spike_params should be less than slab_param. If not specified, spike_params will be assigned default values. See 'Details' for more information.

slab_param
Numeric. Scale parameter λ_1 of slab prior for beta. If not specified, slab_param = spike_params[1] * 10 if spike_params is given; Otherwise, slab_param = 1 by default.

beta_init
Vector. Initial value of regression coefficients beta. beta_init = 0 by default.

sigma_update
Logical. Whether the variance of model error is updated or not. Default is TRUE.

sigma_init
Numeric. The initial value of standard deviation of model error. If not specified, sigma is initialized according to sd(y).

theta_init
Numeric. The initial value of prior proportion of nonzero beta. theta_init must be in (0,1]. Default is 0.5.

return_g
Logical. Default is FALSE. If specified TRUE, return a list containing the expected value of log likelihood function w.r.t the current conditional distribution of latent variables.

a, b
Numeric parameters of Beta prior distribution of theta, where theta ~ Beta(a,b). a = 1 and b = p by default.

omega, kappa
Numeric parameters of Inverse Gamma prior distribution of sigma^2, where sigma^2 ~ IG(omega/2,omega*kappa/2). omega = 1 and kappa = 1 by default.

tolerance
Numeric. Criterion for early stopping at each spike_param. If $||\beta_{old} - \beta_{new}||_2 <$ tolerance, then break the iteration at the current spike_param.

max_iter
Integer. The maximum iteration number at each spike_param.

Details

Since 'MUSS' is built based on EM algorithm, it is possible to converge to local but not global maximum values. Thus the result can be sensitive to the initial choices of beta, sigma and theta.

In addition, the value of spike and slab parameters can be crucial to variable selection result in practice. We set some default values for spike_params and slab_param. If spike_param is not specified and slab_param is given, spike_params = exp(seq(log(slab_L),by=-0.3,length.out=20))[-1] for beta_prior_type = "Laplacian", and spike_params = exp(seq(log(slab_G),by=-0.5,length.out=20))[-1] for beta_prior_type = "Gaussian" by default. If slab_param is also not specified, the slab_param = 1 by default and the spike_param is set the same as the previous way. One can assign any valid values to spike_params and slab_param.

Value

MUSS returns a list containing the following values:

beta_path — $L * p$ matrix. Each row is the beta fitted at corresponding spike_param. For Gaussian case, it is not thresholded.

beta_indices — Vector. Indices of selected nonzero regression parameters.

beta_values — Vector. Values of selected nonzero regression parameters.

beta_output — Vector of length p. Full output beta including zeros. For Laplacian case, it is the last row of beta_path. For Gaussian case, it is the thresholded result from last row of beta_path.

beta_thresholds — Vector of length L. Threshold at each spike_param, returned only when "beta_prior_type" = "Gaussian".

sigma_path — Vector of Length L. Estimated sigma at each spike_param.

theta_path — Vector of Length L. Estimated theta at each spike_param.

g_List — List of Length L. Each element of g_List is a list containing values of maximized function g over iterations at corresponding spike_param.

iter_nums — Vector of Length L. Number of Iterations at each spike_param.

Examples

```
require(MASS)
n = 200
p = 500

set.seed(1234)
beta_true = c(-3, 2, -1.5, -2, 3, rep(0,p-5))
X = matrix(rnorm(n*p, 0, 1), nrow = n)
epsilon = rnorm(n, 0, 1)
y = X %*% beta_true+epsilon

tau = sample(seq(0.5,0.9,by = 0.1), size = p, replace = TRUE)
Xi = mvrnorm(n, rep(0,p), diag(tau))
Z = X + Xi

muss_L = MUSS(Z, y, tauList = tau, beta_prior_type = "Laplacian")
```

pathPlot *pathPlot*

Description

Plot the path of beta, theta or sigma over spike_params. If beta_prior_type = "Gaussian", the thresholds will be displayed as well.

Usage

```
pathPlot(fit_obj, path_of = "beta")
```

Arguments

`fit_obj` The fitted object from 'MUSS' function.

`path_of` Character. `path_of` = `"beta"`, `"theta"` or `"sigma"`. Default is `"beta"`.

`posteriorX` *posteriorX*

Description

In ordinary regression model, assuming each column of the design matrix `Z` contains additive Gaussian measurement errors with known variance collected in `tauList`, function 'posteriorX' returns posterior expectation as well as some related values for the true design matrix X.

More specifically, if we denote the measurement error matrix as Ξ, then the measurement error model takes the form: $Z = X + \Xi$. Since we have $y_i|\boldsymbol{x_i}, \boldsymbol{\beta}, \sigma^2 \sim N(\boldsymbol{x_i}^\top\boldsymbol{\beta}, \sigma^2)$ and $\boldsymbol{x_i}|\boldsymbol{z_i}, \Lambda \sim N(\boldsymbol{z_i}, \Lambda)$ where Λ is the diagonal matrix consists of `tauList`, `posteriorX` computes the expectation of X, variance of $\boldsymbol{x_i}$ and expectation of $X^\top X$ conditioning on $\boldsymbol{\beta}$, $\boldsymbol{y}$, Z, Λ and σ.

This is part of E-step in the 'MUSS' function.

Usage

```
posteriorX(Z, y, beta, sigma, tauList)
```

Arguments

`Z` $n * p$ covariate matrix with additive errors in each column (possibly $p > n$). It is assumed the measurement error model takes the form $Z = X + \Xi$, where X is the unknown true design matrix and Ξ is the matrix of i.i.d Gaussian measurement errors of mean 0 and same variance within each column.

`y` Numeric response vector from n observations.

`beta` Regression parameter vector of length p.

`sigma` Numeric. The standard deviance of regression model error terms.

`tauList` Vector of length p. Corresponding variances of additive measurement error for p columns.

Value

'posteriorX' returns a list containing the following values:

`hatX` $n * p$ matrix. The expectation of X conditional on $\boldsymbol{\beta}$, $\boldsymbol{y}$, Z, Λ and σ.

`hatSigma` $p * p$ matrix. The variance of $\boldsymbol{x_i}$ conditional on $\boldsymbol{\beta}$, $\boldsymbol{y}$, Z, Λ and σ, which is equivalent for each $\boldsymbol{x_i}$.

`hatXX` $p * p$ matrix. The conditional expectation of $X^\top X$. `hatXX = t(hatX) * hatX + hatSigma`.

Examples

```
require(MASS)
n = 200
p = 500

set.seed(1234)
beta_true = c(-3, 2, -1.5, -2, 3, rep(0,p-5))
X = matrix(rnorm(n*p, 0, 1), nrow = n)
epsilon = rnorm(n, 0, 1)
y = X %*% beta_true+epsilon

tau = sample(seq(0.5,0.9,by = 0.1), size = p, replace = TRUE)
Xi = mvrnorm(n, rep(0,p), diag(tau))
Z = X + Xi

post_X = posteriorX(Z, y, beta = beta_true, sigma = 1, tauList = tau)
```

Index