

Package ‘fmerPack’

July 22, 2025

Title Tools of Heterogeneity Pursuit via Finite Mixture Effects Model
Version 0.0-1
Date 2021-01-18
Maintainer Yan Li <yan.4.li@uconn.edu>
Description Heterogeneity pursuit methodologies for regularized finite mixture regression by effects-model formulation proposed by Li et al. (2021) <[doi:10.48550/arXiv.2003.04787](https://doi.org/10.48550/arXiv.2003.04787)>.
Depends R (>= 3.4.0)
Imports utils, flexmix, glmnet, MASS, Rcpp (>= 0.12.0), abind
LinkingTo Rcpp, RcppArmadillo
License GPL (>= 3.0)
Encoding UTF-8
RoxygenNote 7.1.1
NeedsCompilation yes
Author Yan Li [aut, cre],
Kun Chen [aut]
Repository CRAN
Date/Publication 2021-02-01 08:20:02 UTC

Contents

fmrHP	2
fmrReg	4
path.fmrHP	5
path.fmrReg	8
select.tuning	10
Index	11

fmrHP

*Finite Mixture Effects Model with Heterogeneity Pursuit***Description**

Produce solution for specified lambda of regularized finite mixture effects model with lasso or adaptive lasso; compute the degrees of freedom, likelihood and information criteria (AIC, BIC and GIC) of the estimators. Model fitting is conducted by EM algorithm and Bregman coordinate descent.

Usage

```
fmrHP(y, X, m, intercept = FALSE, lambda, equal.var = FALSE,
      ic.type = c("ALL", "BIC", "AIC", "GIC"),
      B = NULL, prob = NULL, rho = NULL, w = NULL,
      control = list(), report = FALSE)
```

Arguments

y	a vector of response ($n \times 1$)
X	a matrix of covariate ($n \times p$)
m	number of components
intercept	indicating whether intercept should be included
lambda	value of tuning parameter
equal.var	indicating whether variances of different components are equal
ic.type	the information criterion to be used; currently supporting "AIC", "BIC", and "GIC".
B	initial values for the rescaled coefficients with first column being the common effect, and the rest m columns being the heterogeneity for corresponding components
prob	initial values for prior probabilities for different components
rho	initial values for rho vector ($1/\sigma$), the reciprocal of standard deviation
w	weight matrix for penalty function. Default option is NULL
control	a list of parameters for controlling the fitting process
report	indicating whether printing the value of objective function during EM algorithm for validation checking of initial value.

Details

The available elements for argument control include

- epsilon: Convergence threshold for generalized EM algorithm. Defaults value is 1E-6.
- maxit: Maximum number of passes over the data for all lambda values. Default is 1000.

- `inner.eps`: Convergence threshold for Bregman coordinate descent algorithm. Defaults value is 1E-6.
- `inner.maxit`: Maximum number of iteration for Bregman coordinate descent algorithm. Defaults value is 200.
- `n.ini`: Number of initial values for EM algorithm. Default is 10. In EM algorithm, it is preferable to start from several different initial values.

Value

A list consisting of

<code>y</code>	vector of response
<code>X</code>	matrix of covariates
<code>m</code>	number of components
<code>B.hat</code>	estimated rescaled coefficient ($p \times m + 1 \times nlambda$)
<code>pi.hat</code>	estimated prior probabilities ($m \times nlambda$)
<code>rho.hat</code>	estimated rho values ($m \times nlambda$)
<code>lambda</code>	lambda used in model fitting
<code>plik</code>	value of penalized log-likelihood
<code>loglik</code>	value of log-likelihood
<code>conv</code>	indicator of convergence of EM algorithm
<code>IC</code>	values of information criteria
<code>df</code>	degree of freedom

Examples

```
library(fmerPack)
## problem settings
n <- 100; m <- 3; p <- 5;
sigma2 <- c(0.1, 0.1, 0.4); rho <- 1 / sqrt(sigma2)
phi <- rbind(c(1, 1, 1), c(1, 1, 1), c(0, -3, 3), c(-3, 3, 0), c(3, 0, -3))
beta <- t(t(phi) / rho)
## generate response and covariates
z <- rmultinom(n, 1, prob= rep(1 / m, m))
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- MASS::mvrnorm(1, mu = rowSums(t(z) * X[, 1:(nrow(beta))] %*% beta),
                  Sigma = diag(colSums(z * sigma2)))
fmrHP(y, X, m = m, lambda = 0.01, control = list(n.ini = 10))
```

fmrReg

*Finite Mixture Model with lasso and adaptive penalty***Description**

Produce solution for specific lambda of regularized finite mixture model with lasso or adaptive lasso penalty; compute the degrees of freedom, likelihood and information criteria (AIC, BIC and GIC) of the estimators. Model fitting is conducted by EM algorithm and coordinate descent.

Usage

```
fmrReg(y, X, m, intercept = FALSE, lambda, equal.var = FALSE, common.var = NULL,
       ic.type = c("ALL", "BIC", "AIC", "GIC"),
       B = NULL, prob = NULL, rho = NULL, w = NULL,
       control = list(), report = FALSE)
```

Arguments

y	a vector of response ($n \times 1$)
X	a matrix of covariate ($n \times p$)
m	number of components
intercept	indicating whether intercept should be included
lambda	value of tuning parameter
equal.var	indicating whether variances of different components are equal
common.var	indicating whether the effects over different components are the same for specific covariates
ic.type	the information criterion to be used; currently supporting "AIC", "BIC", and "GIC".
B	initial values for the rescaled coefficients with columns being the coefficients for different components
prob	initial values for prior probabilities for different components
rho	initial values for rho vector ($1/\sigma$), the reciprocal of standard deviation
w	weight matrix for penalty function. Default option is NULL
control	a list of parameters for controlling the fitting process
report	indicating whether printing the value of objective function during EM algorithm for validation checking of initial value.

Details

The available elements for argument control include

- epsilon: Convergence threshold for generalized EM algorithm. Defaults value is 1E-6.
- maxit: Maximum number of passes over the data for all lambda values. Default is 1000.

- `inner.maxit`: Maximum number of iteration for flexmix package to compute initial values. Defaults value is 200.
- `n.ini`: Number of initial values for EM algorithm. Default is 10. In EM algorithm, it is preferable to start from several different initial values.

Value

A list consisting of

<code>y</code>	vector of response
<code>X</code>	matrix of covariates
<code>m</code>	number of components
<code>B.hat</code>	estimated rescaled coefficient ($p \times m \times nlambda$)
<code>pi.hat</code>	estimated prior probabilities ($m \times nlambda$)
<code>rho.hat</code>	estimated rho values ($m \times nlambda$)
<code>lambda</code>	lambda used in model fitting
<code>plik</code>	value of penalized log-likelihood
<code>loglik</code>	value of log-likelihood
<code>conv</code>	indicator of convergence of EM algorithm
<code>IC</code>	values of information criteria
<code>df</code>	degree of freedom

Examples

```
library(fmerPack)
## problem settings
n <- 100; m <- 3; p <- 5;
sigma2 <- c(0.1, 0.1, 0.4); rho <- 1 / sqrt(sigma2)
phi <- rbind(c(1, 1, 1), c(1, 1, 1), c(0, -3, 3), c(-3, 3, 0), c(3, 0, -3))
beta <- t(t(phi) / rho)
## generate response and covariates
z <- rmultinom(n, 1, prob= rep(1 / m, m))
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- MASS::mvrnorm(1, mu = rowSums(t(z) * X[, 1:(nrow(beta))]) %*% beta),
               Sigma = diag(colSums(z * sigma2)))
fmrReg(y, X, m = m, lambda = 0.01, control = list(n.ini = 10))
```

Description

Produce solution paths of regularized finite mixture effects model with lasso or adaptive lasso penalty; compute the degrees of freedom, likelihood and information criteria (AIC, BIC and GIC) of the estimators. Model fitting is conducted by EM algorithm and Bregman coordinate descent.

Usage

```
path.fmrHP(y, X, m, equal.var = FALSE,
           ic.type = "ALL", B = NULL, prob = NULL, rho = NULL,
           control = list(), modstr = list(), report = FALSE)
```

Arguments

y	a vector of response ($n \times 1$)
X	a matrix of covariate ($n \times p$)
m	number of components
equal.var	indicating whether variances of different components are equal
ic.type	the information criterion to be used; currently supporting "AIC", "BIC", and "GIC".
B	initial values for the rescaled coefficients with first column being the common effect, and the rest m columns being the heterogeneity for corresponding components
prob	initial values for prior probabilities for different components
rho	initial values for rho vector ($1/\sigma$), the reciprocal of standard deviation
control	a list of parameters for controlling the fitting process
modstr	a list of model parameters controlling the model fitting
report	indicating whether printing the value of objective function during EM algorithm for validation checking of initial value.

Details

Model parameters can be specified through argument `modstr`. The available include

- `lambda`: A vector of user-specified lambda values with default NULL.
- `lambda.min.ratio`: Smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value.
- `nlambda`: The number of lambda values.
- `w`: Weight matrix for penalty function. Default option is NULL, which means lasso penalty is used for model fitting.
- `intercept`: Should intercept(s) be fitted (default=TRUE) or set to zero (FALSE).
- `common.only`: A vector of user-specified indicators of the variables only with common effects.
- `common.no.penalty`: A vector of user-specified indicators of the variables with no penalty on the common effect.
- `cluster.no.penalty`: A vector of user-specified indicators of the variables with no penalty on the cluster-specific effects.
- `select.ratio`: A user-specified ratio indicating the ratio of variables to be selected.

The available elements for argument `control` include

- `epsilon`: Convergence threshold for generalized EM algorithm. Defaults value is 1E-6.

- `maxit`: Maximum number of passes over the data for all lambda values. Default is 1000.
- `inner.eps`: Convergence threshold for Bregman coordinate descent algorithm. Defaults value is 1E-6.
- `inner.maxit`: Maximum number of iteration for Bregman coordinate descent algorithm. Defaults value is 200.
- `n.ini`: Number of initial values for EM algorithm. Default is 10. In EM algorithm, it is preferable to start from several different initial values.

Value

A list consisting of

<code>lambda</code>	vector of lambda used in model fitting
<code>lambda.used</code>	vector of lambda in model fitting after truncation by <code>select.ratio</code>
<code>B.hat</code>	estimated rescaled coefficient ($p \times m + 1 \times nlambda$)
<code>pi.hat</code>	estimated prior probabilities ($m \times nlambda$)
<code>rho.hat</code>	estimated rho values ($m \times nlambda$)
<code>IC</code>	values of information criteria

References

Li, Y., Yu, C., Zhao, Y., Yao, W., Aseltine, R. H., & Chen, K. (2021). Pursuing Sources of Heterogeneity in Modeling Clustered Population.

Examples

```
library(fmerPack)
## problem settings
n <- 100; m <- 3; p <- 5;
sigma2 <- c(0.1, 0.1, 0.4); rho <- 1 / sqrt(sigma2)
phi <- rbind(c(1, 1, 1), c(1, 1, 1), c(1, 1, 1), c(-3, 3, 0), c(3, 0, -3))
beta <- t(t(phi) / rho)
## generate response and covariates
z <- rmultinom(n, 1, prob= rep(1 / m, m))
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- MASS::mvrnorm(1, mu = rowSums(t(z) * X[, 1:(nrow(beta))]) %*% beta),
               Sigma = diag(colSums(z * sigma2)))
## lasso
fit1 <- path.fmrHP(y, X, m = m, modstr = list(nlambda = 10), control = list(n.ini = 1))
## adaptive lasso
fit2 <- path.fmrHP(y, X, m = m,
                  modstr = list(w = abs(select.tuning(fit1)$B + 1e-6)^2))
```

path.fmrReg	<i>Finite Mixture Model with lasso and adaptive penalty</i>
-------------	---

Description

Produce solution paths of regularized finite mixture model with lasso or adaptive lasso penalty; compute the degrees of freedom, likelihood and information criteria (AIC, BIC and GIC) of the estimators. Model fitting is conducted by EM algorithm and coordinate descent.

Usage

```
path.fmrReg(y, X, m, equal.var = FALSE,
            ic.type = "ALL", B = NULL, prob = NULL, rho = NULL,
            control = list(), modstr = list(), report = FALSE)
```

Arguments

y	a vector of response ($n \times 1$)
X	a matrix of covariate ($n \times p$)
m	number of components
equal.var	indicating whether variances of different components are equal
ic.type	the information criterion to be used; currently supporting "ALL", "AIC", "BIC", and "GIC".
B	initial values for the rescaled coefficients with columns being the columns being the coefficient for different components
prob	initial values for prior probabilities for different components
rho	initial values for rho vector ($1/\sigma$), the reciprocal of standard deviation
control	a list of parameters for controlling the fitting process
modstr	a list of model parameters controlling the model fitting
report	indicating whether printing the value of objective function during EM algorithm for validation checking of initial value.

Details

Model parameters can be specified through argument modstr. The available include

- lambda: A vector of user-specified lambda values with default NULL.
- lambda.min.ratio: Smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value.
- nlambdas: The number of lambda values.
- w: Weight matrix for penalty function. Default option is NULL, which means lasso penalty is used for model fitting.
- intercept: Should intercept(s) be fitted (default=TRUE) or set to zero (FALSE).

- no.penalty: A vector of user-specified indicators of the variables with no penalty.
- common.var: A vector of user-specified indicators of the variables with common effect among different components.
- select.ratio: A user-specified ratio indicating the ratio of variables to be selected.

The available elements for argument control include

- epsilon: Convergence threshold for generalized EM algorithm. Defaults value is 1E-6.
- maxit: Maximum number of passes over the data for all lambda values. Default is 1000.
- inner.maxit: Maximum number of iteration for flexmix package to compute initial values. Defaults value is 200.
- n.ini: Number of initial values for EM algorithm. Default is 10. In EM algorithm, it is preferable to start from several different initial values.

Value

A list consisting of

lambda	vector of lambda used in model fitting
B.hat	estimated rescaled coefficient ($p \times m \times nlambda$)
pi.hat	estimated prior probabilities ($m \times nlambda$)
rho.hat	estimated rho values ($m \times nlambda$)
IC	values of information criteria

Examples

```
library(fmerPack)
## problem settings
n <- 100; m <- 3; p <- 5;
sigma2 <- c(0.1, 0.1, 0.4); rho <- 1 / sqrt(sigma2)
phi <- rbind(c(1, 1, 1), c(1, 1, 1), c(1, 1, 1), c(-3, 3, 0), c(3, 0, -3))
beta <- t(t(phi) / rho)
## generate response and covariates
z <- rmultinom(n, 1, prob= rep(1 / m, m))
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- MASS::mvrnorm(1, mu = rowSums(t(z) * X[, 1:(nrow(beta))]) %*% beta),
                Sigma = diag(colSums(z * sigma2)))
## lasso
fit1 <- path.fmrReg(y, X, m = m, modstr = list(nlambda = 10), control = list(n.ini = 1))
## adaptive lasso
fit2 <- path.fmrReg(y, X, m = m,
                    modstr = list(w = abs(select.tuning(fit1)$B + 1e-6)^2))
```

select.tuning	<i>Tuning parameter selection</i>
---------------	-----------------------------------

Description

Select tuning parameter via AIC, BIC or GIC from objects generated by `path.fmrHP`.

Usage

```
select.tuning(object, figure = FALSE, criteria = c("BIC", "GIC", "AIC"))
```

Arguments

<code>object</code>	Object generated from <code>path.fmrHP</code> .
<code>figure</code>	incidator for showing plot of information criteria.
<code>criteria</code>	information criteria for selection of tuning parameter.

Value

list of parameters of selected model.

Index

fmrHP, [2](#)

fmrReg, [4](#)

path.fmrHP, [5](#)

path.fmrReg, [8](#)

select.tuning, [10](#)