Package ‘PARSE’

June 11, 2016

Type Package

Title Model-Based Clustering with Regularization Methods for High-Dimensional Data

Version 0.1.0

Date 2016-06-10

Author Lulu Wang, Wen Zhou, Jennifer Hoeting

Maintainer Lulu Wang <wanglulu@stat.colostate.edu>

Description Model-based clustering and identifying informative features based on regularization methods. The package includes three regularization methods - PAirwise Reciprocal fuSE (PARSE) penalty proposed by Wang, Zhou and Hoeting (2016), the adaptive L1 penalty (APL1) and the adaptive pairwise fusion penalty (APFP). Heatmaps are included to shown the identification of informative features.

LazyData TRUE

Depends R (>= 3.0.0)

Imports stats, mvtnorm, gplots, foreach, doParallel, grDevices, utils

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-06-11 09:42:05

R topics documented:

apfp *Model-based Clustering with APFP*

Description

The adaptive pairwise fusion penalty (APFP) was proposed by Guo (2010). Under the framework of the model-based clustering, APFP aims to identify the pairwise informative variables for clustering high-dimenisonal data.

Usage

```
apfp(tuning, K = NULL, lambda = NULL, y, N = 100, kms.iter = 100, kms.nstart = 100,
      adapt.kms = FALSE, eps.diff = 1e-5, eps.em = 1e-5,
      iter.LQA = 20, eps.LQA = 1e-5, model.crit = 'gic')
apfp(tuning = NULL, K, lambda, y, N = 100, kms.iter = 100, kms.nstart = 100,
      adapt.kms = FALSE, eps.diff = 1e-5, eps.em = 1e-5,
      iter.LQA = 20, eps.LQA = 1e-5, model.crit = 'gic')
```

Arguments

`tuning`	A 2-dimensional vector or a matrix with 2 columns, the first column is the number of clusters K and the second column is the tuning parameter λ in the penalty term. If this is missing, then `K` and `lambda` must be provided.
`K`	The number of clusters K.
`lambda`	The tuning parameter λ in the penalty term.
`y`	A p-dimensional data matrix. Each row is an observation.
`N`	The maximum number of iterations in the EM algorithm. The default value is 100.
`kms.iter`	The maximum number of iterations in kmeans algorithm for generating the starting value for the EM algorithm.
`kms.nstart`	The number of starting values in K-means.
`adapt.kms`	A indicator of using the cluster means estimated by K-means to calculate the adaptive parameters in APFP. The default value is FALSE.
`eps.diff`	The lower bound of pairwise difference of two mean values. Any value lower than it is treated as 0.
`eps.em`	The lower bound for the stopping criterion in the EM algorithm.
`iter.LQA`	The number of iterations in the estimation of cluster means by using the local quadratic approximation (LQA).
`eps.LQA`	The lower bound for the stopping criterion in the estimation of cluster means.
`model.crit`	The criterion used to select the number of clusters K. It is either 'bic' for Bayesian Information Criterion or 'gic' for Generalized Information Criterion.

Details

The j-th variable is defined as pairwise informative for a pair of clusters C_k and $C_{k'}$ if $\mu_{kj} \neq \mu_{k'j}$. Also, a variable is globally informative if it is pairwise informative for at least one pair of clusters. Here we assume that each cluster has the same diagonal variance in the model-based clustering. APFP is in the following form,

$$\sum_{j=1}^{d} \sum_{k<k'} \tau_{kk'j} |\mu_{kj} - \mu_{k'j}|,$$

where d is the number of variables in the data, $\tau_{kk'j} = |\tilde{\mu}_{kj} - \tilde{\mu}_{k'j}|^{-1}$ is the adaptive parameters. Here we provide two choices for $\tilde{\mu_{kj}}$. If `adapt.kms == TRUE`, $\tilde{\mu}_{kj}$ is the estimates from the K-mean algorithm; otherwise, $\tilde{\mu}_{kj}$ is the estimates from the model-based clustering without penalty.

The estimation uses the EM algorithm. Since the EM algorithm depends on the starting values. We use the estimates from K-means with multiple starting points as the starting values. For estimating the cluster means, APFP uses the local quadratic approximation.

Value

This function returns the esimated parameters and some statistics of the optimal model within the given K and λ, which is selected by BIC when `model.crit = 'bic'` or GIC when `model.crit = 'gic'`.

`mu.hat.best`	The estimated cluster means in the optimal model
`sigma.hat.best`	The estimated covariance in the optimal model
`p.hat.best`	The estimated cluster proportions in the optimal model
`s.hat.best`	The clustering assignments using the optimal model
`lambda.best`	The value of λ that provide the optimal model
`K.best`	The value of K that provide the optimal model
`llh.best`	The log-likelihood of the optimal model
`gic.best`	The GIC of the optimal model
`bic.best`	The BIC of the optimal model
`ct.mu.best`	The degrees of freedom in the cluster means of the optimal model

References

Guo, J., Levina, E., Michailidis, G., and Zhu, J. (2010) Pairwise variable selection for high-dimensional model-based clustering. *Biometrics* **66(3)**, 793–804.

See Also

`nopenalty` `apL1` `parse`

Examples

```
y <- rbind(matrix(rnorm(100,0,1),ncol=2), matrix(rnorm(100,4,1), ncol=2))
output <- apfp(K = c(1:2), lambda = c(0,1), y=y)
output$mu.hat.best
```

apL1 *Model-based Clustering with APL1*

Description

The adaptive L_1 penalty was proposed by Pan and Shen (2007). Under the framework of the model-based clustering, APL1 aims to identify the globally informative variables for clustering high-dimensional data.

Usage

```
apL1(tuning, K = NULL, lambda = NULL, y, N = 100, kms.iter = 100, kms.nstart = 100,
      adapt.kms = FALSE, eps.diff = 1e-5, eps.em = 1e-5, model.crit = 'gic')
apL1(tuning = NULL, K, lambda, y, N = 100, kms.iter = 100, kms.nstart = 100,
      adapt.kms = FALSE, eps.diff = 1e-5, eps.em = 1e-5, model.crit = 'gic')
```

Arguments

`tuning`	A 2-dimensional vector or a matrix with 2 columns, the first column is the number of clusters K and the second column is the tuning parameter λ in the penalty term. If this is missing, then `K` and `lambda` must be provided.
`K`	The number of clusters K.
`lambda`	The tuning parameter λ in the penalty term.
`y`	A p-dimensional data matrix. Each row is an observation.
`N`	The maximum number of iterations in the EM algorithm. The default value is 100.
`kms.iter`	The maximum number of iterations in kmeans algorithm for generating the starting value for the EM algorithm.
`kms.nstart`	The number of starting values in K-means.
`adapt.kms`	A indicator of using the cluster means estimated by K-means to calculate the adaptive parameters in APFP. The default value is FALSE.
`eps.diff`	The lower bound of pairwise difference of two mean values. Any value lower than it is treated as 0.
`eps.em`	The lower bound for the stopping criterion.
`model.crit`	The criterion used to select the number of clusters K. It is either 'bic' for Bayesian Information Criterion or 'gic' for Generalized Information Criterion.

Details

A variable is defined as globally informative if there exists at least one pair of clusters such that $\mu_{kj} \neq \mu_{k'j}$. Here we assume that each cluster has the same diagonal variance in the model-based clustering. APL1 is in the following form,

$$\sum_{j=1}^{d} \tau_{kj} \sum_{k=1}^{K} |\mu_{kj}|,$$

where d is the number of variables in the data, K is the number of clusters, $\tau_{kj} = \tilde{\mu}_{kj}$ is the adaptive parameters. Here we provide two choices for τ_{kj}. If `adapt.kms == TRUE`, $\tilde{\mu}_{kj}$ is the estimates from the K-mean algorithm; otherwise, $\tilde{\mu}_{kj}$ is the estimates from the model-based clustering without penalty.

The EM algorithm is used for estimating parameters. Since the EM algorithm depends on the starting values. We use the estimates from K-means with multiple starting points as the starting values.

Value

This function returns the esimated parameters and some statistics of the optimal model within the given K and λ, which is selected by BIC when `model.crit = 'bic'` or GIC when `model.crit = 'gic'`.

`mu.hat.best`	The estimated cluster means in the optimal model
`sigma.hat.best`	The estimated covariance in the optimal model
`p.hat.best`	The estimated cluster proportions in the optimal model
`s.hat.best`	The clustering assignments using the optimal model
`lambda.best`	The value of λ that provide the optimal model
`K.best`	The value of K that provide the optimal model
`llh.best`	The log-likelihood of the optimal model
`gic.best`	The GIC of the optimal model
`bic.best`	The BIC of the optimal model
`ct.mu.best`	The degrees of freedom in the cluster means of the optimal model

References

Pan, W. and Shen, X. (2007). Penalized model-based clustering with application to variable selection. *The Journal of Machine Learning Research* **8**, 1145–1164.

See Also

`nopenalty` `apfp` `parse`

Examples

```
y <- rbind(matrix(rnorm(100,0,1),ncol=2), matrix(rnorm(100,4,1), ncol=2))
output <- apL1(K = c(1:2), lambda = c(0,0.1), y=y)
output$mu.hat.best
```

`heatmap_fit` *summary plot of globally and pairwise informative variables*

Description

Heatmaps of the data with estimated informative variables and the indicator for pairwise informativeness of each globally informative variables.

Usage

```
heatmap_fit(output, y, plot_type = 'info.data', eps.diff = 1e-5,
margins = c(5,5), cexRow = 0.5, cexCol = 0.4, lhei = c(0.8,5),
lwid=c(0.8,5), adjCol = c(0.8,0.4), sepwidth=c(0.05,0.05))
```

Arguments

`output`	results from parse, apfp, apL1 or nopenalty functions. For the ‘nopenalty’ function, the ‘short.output’ should be FALSE.
`y`	data.
`plot_type`	takes two values, ’info.data’ or ’info.pair’. ’info.data’ is the heatmap of the data with informative variables; ’info.pair’ indicates which globally informative variable is pairwise informative for each pair of clusters.
`eps.diff`	The lower bound of pairwise difference of two mean values. Any value lower than it is treated as 0. The default value is 1e-5.
`margins`	parameter in ’heatmap.2’ function, 2-dimensional numeric vector containing the margins for column and row names, respectively.
`cexRow`	parameter in ’heatmap.2’ function, positive numbers for the row axis labeling.
`cexCol`	parameter in ’heatmap.2’ function, positive numbers for the column axis labeling.
`lhei`	parameters in ’heatmap.2’ function, visual layout of column height.
`lwid`	parameters in ’heatmap.2’ function, visual layout of column weight.
`adjCol`	parameters in ’heatmap.2’ function, justification of column labels (variables names).
`sepwidth`	parameters in ’heatmap.2’ function, 2-dimensional vector giving the width and height of the separator box

Value

heatmap of the data with informative variables or heatmap of whether the globally informative variables are pairwise informative for each pair of clusters or not.

References

Gregory R. Warnes, Ben Bolker, Lodewijk Bonebakker, Robert Gentleman, Wolfgang Huber Andy Liaw, Thomas Lumley, Martin Maechler, Arni Magnusson, Steffen Moeller, Marc Schwartz and Bill Venables (2015). gplots: Various R Programming Tools for Plotting Data. R package version 2.17.0. https://CRAN.R-project.org/package=gplots

See Also

heatmap.2

Examples

```
y <- rbind(matrix(rnorm(120,0,1),ncol=4),
matrix(rnorm(120,4,1), ncol=4), matrix(rnorm(120,0,1),ncol=4))
output <- parse(K = 3, lambda = 1, y=y)
output$mu.hat.best
heatmap_fit(output, y, cexRow=1)
```

---

nopenalty — *Classical Model-based Clustering*

Description

This function estimates the model-based clustering which is under the framework of finite mixture models.

Usage

```
nopenalty(K, y, N = 100, kms.iter = 100, kms.nstart = 100,
           eps.diff = 1e-5, eps.em = 1e-5,
           model.crit = 'gic', short.output = FALSE)
```

Arguments

K	A vector of the number of clusters
y	A p-dimensional data matrix. Each row is an observation
N	The maximum number of iterations in the EM algorithm. The default value is 100.
kms.iter	The maximum number of iterations in the K-means algorithm whose outputs are the starting values for the EM algorithm
kms.nstart	The number of starting values in K-means
eps.diff	The lower bound of pairwise difference of two mean values. Any value lower than it is treated as 0
eps.em	The lower bound for the stopping criterion.

`model.crit`	The criterion used to select the number of clusters K. It is either 'bic' for Bayesian Information Criterion or 'gic' for Generalized Information Criterion.
`short.output`	A short version of output is needed or not. A short version is used for computing the adaptive parameters in APFP or APL1 methods. The default value is FALSE.

Details

This function estimates parameters μ, Σ, π and the clustering assignments in the model-based clustering using the mixture model,

$$y \sim \sum_{k=1}^{K} \pi_k f(y|\mu_k, \Sigma)$$

where $f(y|\mu_k, \Sigma_k)$ is the density function of Normal distribution with mean μ_k and variance Σ. Here we assume that each cluster has the same diagonal variance.

This function is also used to compute the adaptive parameters for functions `apfp` and `apL1`.

Value

This function returns the esimated parameters and some statistics of the optimal model within the given K and λ, which is selected by BIC when `model.crit = 'bic'` or GIC when `model.crit = 'gic'`.

`mu.hat.best`	The estimated cluster means.
`sigma.hat.best`	The estimated covariance.
`p.hat.best`	The estimated cluster proportions.
`s.hat.best`	The clustering assignments.
`K.best`	The value of K that provides the optimal model
`llh.best`	The log-likelihood of the optimal model
`gic.best`	The GIC of the optimal model
`bic.best`	The BIC of the optimal model
`ct.mu.best`	The degrees of freedom in the cluster means of the optimal model

References

Fraley, C., & Raftery, A. E. (2002) Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association* **97(458)**, 611–631.

See Also

`apfp` `apL1` `parse`

Examples

```
y <- rbind(matrix(rnorm(100,0,1),ncol=2), matrix(rnorm(100,4,1), ncol=2))
output <- nopenalty(K = c(1:2), y)
output$mu.hat.best
```

parse *Model-based Clustering with PARSE*

Description

The PAirwise Reciprocal fuSE (PARSE) penalty was proposed by Wang, Zhou and Hoeting (2016). Under the framework of the model-based clustering, PARSE aims to identify the pairwise informative variables for clustering, especially for high-dimensional data.

Usage

```
parse(tuning, K = NULL, lambda = NULL, y, N = 100, kms.iter = 100, kms.nstart = 100,
     eps.diff = 1e-5, eps.em = 1e-5, model.crit = 'gic', backward = TRUE, cores=2)
parse(tuning = NULL, K, lambda, y, N = 100, kms.iter = 100, kms.nstart = 100,
     eps.diff = 1e-5, eps.em = 1e-5, model.crit = 'gic', backward = TRUE, cores=2)
```

Arguments

`tuning`	A 2-dimensional vector or a matrix with 2 columns, the first column is the number of clusters K and the second column is the tuning parameter λ in the penalty term. If this is missing, then `K` and `lambda` must be provided.
`K`	The number of clusters K.
`lambda`	The tuning parameter λ in the penalty term.
`y`	A p-dimensional data matrix. Each row is an observation.
`N`	The maximum number of iterations in the EM algorithm. The default value is 100.
`kms.iter`	The maximum number of iterations in kmeans algorithm for generating the starting value for the EM algorithm.
`kms.nstart`	The number of starting values in K-means.
`eps.diff`	The lower bound of pairwise difference of two mean values. Any value lower than it is treated as 0.
`eps.em`	The lower bound for the stopping criterion.
`model.crit`	The criterion used to select the number of clusters K. It is either 'bic' for Bayesian Information Criterion or 'gic' for Generalized Information Criterion.
`backward`	Use the backward selection algorithm when it equals to "TRUE", otherwise select all the possible subsets.
`cores`	The number of cores which can be used in parallel computing.

Details

The j-th variable is defined as pairwise informative for a pair of clusters C_k and $C_{k'}$ if $\mu_{kj} \neq \mu_{k'j}$. Also, a variable is globally informative if it is pairwise informative for at least one pair of clusters.

Here we assume that each cluster has the same diagonal variance in the model-based clustering. PARSE is in the following form,

$$\sum_{j=1}^{d} \sum_{k<k'} |\mu_{kj} - \mu_{k'j}|^{-1} \mathbf{I}(\mu_{kj} \neq \mu_{k'j}).$$

where d is the number of variables in the data.

The estimation uses the backward searching algorithm embedded in the EM algorithm. Since the EM algorithm depends on the starting values. We use the estimates from K-means with multiple starting points as the starting values. Please check the paper for details of the algorithm. In this function we use parallel computing to estimate cluster means for each dimension. The default number of cores to be used is 2, which can be specified by users.

Value

This function returns the esimated parameters and some statistics of the optimal model within the given K and λ, which is selected by BIC when `model.crit = 'bic'` or GIC when `model.crit = 'gic'`.

`mu.hat.best`	The estimated cluster means in the optimal model
`sigma.hat.best`	The estimated covariance in the optimal model
`p.hat.best`	The estimated cluster proportions in the optimal model
`s.hat.best`	The clustering assignments using the optimal model
`lambda.best`	The value of λ that provide the optimal model
`K.best`	The value of K that provide the optimal model
`llh.best`	The log-likelihood of the optimal model
`gic.best`	The GIC of the optimal model
`bic.best`	The BIC of the optimal model
`ct.mu.best`	The degrees of freedom in the cluster means of the optimal model

References

Wang, L., Zhou, W. and Hoeting, J. (2016) Identification of Pairwise Informative Features for Clustering Data. *preprint*.

See Also

`optim` `nopenalty` `apL1` `apfp` `foreach` `doParallel`

Examples

```
y <- rbind(matrix(rnorm(120,0,1),ncol=3), matrix(rnorm(120,4,1), ncol=3))
output <- parse(K = c(1:2), lambda = c(0,1), y=y, cores=2)
output$mu.hat.best
```

`response2drug` *Gene-expression Data for Asthma Disease*

Description

The data contains gene expression data of 108 objects.

Usage

```
data(response2drug)
```

Format

Rows Objects

Columns Genes

Details

This is the microarray gene expression data from NCBI's Gene Expression Omnibus database with Gene Expression Omnibus Series accession number GSE43696. The data consist of 108 samples with 20 healthy, 50 moderate asthma and 38 severe asthma patients; and 405 genes. Each row represents one observation and each column represents one gene.

Source

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE43696

References

Voraphani, N., Gladwin, M.T., Contreras, A.U., Kaminski, N., Tedrow, J.R., Milosevic, J., Bleecker, E.R., Meyers, D.A., Ray, A., Ray, P. and Erzurum, S.C. (2014) An airway epithelial iNOS-DUOX2-thyroid peroxidase metabolome drives Th1/Th2 nitrative stress in human severe asthma. *Mucosal immunology* **7(5)**, 1175-1185.

Examples

```
data(response2drug)
output1 = parse (K=2, lambda = 1, y = response2drug, N = 100,
kms.iter = 50, kms.nstart = 50, eps.diff = 1e-5, eps.em = 1e-5,
model.crit = 'gic', backward = TRUE, cores = 2)
output1$mu.hat.best[, 1:5]
```

`summary` *summary of the clustering results*

Description

Summary of the globally informative variables

Usage

```
summary(output, y, eps.diff = 1e-5)
```

Arguments

`output`	results from parse, apfp, apL1 or nopenalty functions. For the ‘nopenalty’ function, the ‘short.output’ should be FALSE.
`y`	data.
`eps.diff`	The lower bound of pairwise difference of two mean values. Any value lower than it is treated as 0.

Value

`num.info`	the number of globally informative variables
`perc.info`	the percentage of globally informative variables
`info.name`	the variable names of the globally informative variables, if the data have no variable name, ’info.name’ is the index of the variable

Examples

```
y <- rbind(matrix(rnorm(120,0,1),ncol=4),
matrix(rnorm(120,4,1), ncol=4), matrix(rnorm(120,0,1),ncol=4))
output <- parse(K = c(1:2), lambda = c(0,1), y=y)
output$mu.hat.best
summary(output, y)
```

Index