Package 'ClustMMDD'

May 30, 2016

Type Package

Title Variable Selection in Clustering by Mixture Models for Discrete Data

Version 1.0.4

Date 2016-05-30

Author Wilson Toussile

Maintainer Wilson Toussile <wilson.toussile@gmail.com>

Description An implementation of a variable selection procedure in clustering by mixture models for discrete data (clustMMDD). Genotype data are examples of such data with two unordered observations (alleles) at each locus for diploid individual. The two-fold problem of variable selection and clustering is seen as a model selection problem where competing models are characterized by the number of clusters K, and the subset S of clustering variables. Competing models are compared by penalized maximum likelihood criteria. We considered asymptotic criteria such as Akaike and Bayesian Information criteria, and a family of penalized criteria with penalty function to be data driven calibrated.

License GPL (>= 2)

Depends Rcpp (>= 0.11.5), R (>= 3.0.0)

Collate ``RcppExports.R'' ``ClustMMDD.R'' ``modelKS.R'' ``zzz.R''

LazyLoad true

Imports methods

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-05-30 20:26:19

R topics documented:

ClustMMDD-package   ClustMMDD *: Clustering by Mixture Models for Discrete Data.*

Description

ClustMMDD stands for "Clustering by Mixture Models for Discrete Data". This package deals with the two-fold problem of variable selection and model-based unsupervised classification in discrete settings. Variable selection and classification are simultaneously solved via a model selection procedure using penalized criteria: Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Integrated Completed Log-likelihood (ICL) or a general criterion with penalty function to be data-driven calibrated.

Details

Package:	ClustMMDD
Type:	Package
Version:	1.0.1
Date:	2015-05-18
License:	GPL (>= 2)

In this package, `K` and `S` are respectively the number of clusters and the subset of variables that are relevant for clustering purposes. We assume that a clustering variable has different probability distributions in at least two clusters, and a non-clustering variable has the same distribution in all clusters. We consider a general situation with data described by P random variables X^l, $l = 1, \cdots, P$, where each variable X^l is an unordered set $\left\{X^{l,1}, \cdots, X^{l,ploidy}\right\}$ of $ploidy$ categorical variables. For all l, the random variables $X^{l,1}, \cdots, X^{l,ploidy}$ take their values in the same set of levels. A typical example of such data comes from population genetics where each genotype of a diploid individual is constituted by $ploidy = 2$ unordered alleles.

The two-fold problem of clustering and variable selection is seen as a model selection problem. A specific collection of competing models associated to different values of `(K, S)` is defined, and are compared using penalized criteria. The penalized criteria are of the form

$$crit\left(K, S\right) = \gamma_n\left(K, S\right) + pen\left(K, S\right),$$

where

- $\gamma_n\left(K, S\right)$ is the maximum log-likelihood,
- and $pen\left(K, S\right)$ the penalty function.

The penalty functions used in this package are the following, where $dim\left(K, S\right)$ is the dimension (number of free parameters) of the model defined by $\left(K, S\right)$:

- Akaike Information Criterion (AIC) :

  $$pen\left(K, S\right) = dim\left(K, S\right)$$

- Bayesian Information (BIC) :

  $$pen\left(K, S\right) = 0.5 * \log(n) * dim\left(K, S\right)$$

- Integrated Complete Likelihood (ICL) :

  $$pen\left(K, S\right) = 0.5 * \log(n) * dim\left(K, S\right) + entropy\left(K, S\right),$$

  where

  $$entropy\left(K, S\right) = -\sum_{i=1}^{N}\sum_{k=1}^{K}\tau_{i,k}\log\left(\tau_{i,k}\right)$$

  and

  $$\tau_{i,k} = P\left(i \in \mathcal{C}_k\right)$$

  .

- More general penalty function :

  $$pen\left(K, S\right) = \alpha * \lambda * dim\left(K, S\right)$$

  where

  - λ is a multiplicative parameter to be calibrated,
  - α a coefficient in $[1.5, 2]$ to be given by the user.

  We propose a data driven procedure based the dimension jumb version of the so called "slope heuristics" (see Dominique Bontemps and Wilson Toussile (2013) and references therein).

The maximum log-likelihood is estimated via the Expectation and Maximisation algorithm. The maximum a posteriori classification is derived from the estimated parameters of the selected model.

Author(s)

Wilson Toussile

Maintainer: Wilson Toussile <wilson.toussile@gmail.com>

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

See Also

The main functions :

`em.cluster.R` Compute an approximation of the maximum likelihood estimates of parameters using Expectation and Maximization algorithm, for a given value of (K, S). The maximum a posteriori classification is then derived.

`backward.explorer` Gather the most competitive models using a backward-stepwise strategy.

`dimJump.R` Perform the data driven calibration of the penalty function via an estimation of λ. Two values are proposed and a graphic is proposed to help user in making a choice.

`selectK.R` Perform the selection of the number K of clusters for a given subset of clustering variables.

`model.selection.R` Perform a model selection from a collection of competing models.

Examples

```
data(genotype2)
head(genotype2)
data(genotype2_ExploredModels)
head(genotype2_ExploredModels)

#Calibration of the penalty function
outDimJump = dimJump.R(genotype2_ExploredModels, N = 1000, h = 5, header = TRUE)
cte1 = outDimJump[[1]][1]
outSlection = model.selection.R(genotype2_ExploredModels, cte = cte1, header = TRUE)
outSlection
```

`==-methods` *Methods for Function ==*

Description

Check if two objects of class `modelKS` are equal.

Methods

```
signature(e1 = "modelKS", e2 = "modelKS")
```

Author(s)

Wilson Toussile.

See Also

slotNames, new, methods, show

Examples

```
showClass("modelKS")
slotNames("modelKS")
data(exModelKS)
exModelKS
exModelKS == exModelKS
```

backward.explorer *Gather a set of the most competitive models.*

Description

This function gathers a set of the most competitive models using a backward-stepwise strategy. The visited models are gathered in a file with suffix "_ExploredModels.txt". The algorithm used is described in Wilson Toussile and Elisabeth Gassiat (2009).

Usage

```
backward.explorer(x, Kmax, Criterion, ploidy = 1,
  ForceExclusion = FALSE, emOptions = list(epsi = NULL, nberSmallEM = NULL,
  nberIterations = NULL, nberMaxIterations = NULL, typeSmallEM = NULL, typeEM =
  NULL, putThreshold = NULL), Kmin = 1, Smin = NULL,
  project = deparse(substitute(x)))
```

Arguments

x	A matrix of string that contains data.
Kmax	The maximum number of clusters to be explored.
Criterion	The model selection criterion in c("BIC", "AIC", "ICL", "CteDim") used for exploration (see details).
ploidy	The number of columns for each variable in the data. For example, $ploidy = 2$ for genotypic data from diploid individual.
ForceExclusion	The indication of whether to force exclusion or not. The default value is set to FALSE.

`emOptions` A list of EM options (see `EmOptions` and `setEmOptions`).

`Kmin` The minimum number of clusters. The default value is set to 1.

`Smin` A logical vector that indicates the variables to include in the selected set of clustering variables. The default value NULL: no variable is preselected.

`project` The name of the project. The default value is the name of the dataset.

Details

If the penalized criteria is `CteDim`, a sequence of penalty functions of the form $pen\left(K,S\right) = \lambda * dim\left(K,S\right)$ is used. In this shape of penalty function, λ is in $[0.5, log(N)]$, where N is the number of individuals in the sample data. Thus, `AIC` and `BIC` penalties are in the sequence of candidate penalties.

Value

A data.frame of selected models for the choosen proposed criteria.

Author(s)

Wilson Toussile

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

See Also

`dimJump.R` for the data driven calibration of the penalty function, and `model.selection.R` for the final model selection.

Examples

```
data(genotype1)
head(genotype1)
genotype2 = cutEachCol(genotype1[, -11], ploidy = 2)
head(genotype2)

# The following command create a file "genotype2_ExploredModels.txt"
# that contains the most competitive models.

#output = backward.explorer(genotype2, Kmax = 10, ploidy = 2, Kmin = 1, Criterion = "CteDim")

data(genotype2_ExploredModels)
head(genotype2_ExploredModels)
```

cutEachCol *Retrieve data from strings in the dataset.*

Description

It is assumed that each string in the data frame submitted represents a set of $ploidy$ unordered observations from the same set of levels. For example, for $ploidy = 2$, the data "101102" represents "101", "102".

Usage

```
cutEachCol(xdata, ploidy)
```

Arguments

`xdata`	A data.frame or a matrix of strings.
`ploidy`	The number of unordered observations represented by a string in `xdata`. For example, for genotypic data from diploid individual, $ploidy = 2:$ a data such as "ab" represents {`"a", "b"`} observed alleles.

Value

A matrix of strings compatible with the main functions of `ClustMMDD`. The number of columns in the outcome data frame is equal to $ploidy * ncol(xdata)$.

Author(s)

Wilson Toussile

See Also

`dataR2C` for `ClustMMDD` data format.

Examples

```
data(genotype1)
head(genotype1)
genotype2 = cutEachCol(genotype1[,-11], ploidy = 2)
head(genotype2)
```

dataR2C *Transform a (normal) data frame to be compatible with ClustMMDD main functions*

Description

`dataR2C(x, ploidy)` returns a list.

Usage

```
dataR2C(x, ploidy = 1)
```

Arguments

`x`	A data frame or a matrix with the number of columns equal to $number(variables) * ploidy$.
`ploidy`	The number of unordered observations represented by a string in `xdata`. For example, for genotypic data from diploid individual, $ploidy = 2$: two columns for one variable.

Value

A list of elements needed for `ClustMMDD` main functions :

- data : A matrix compatible with `ClustMMDD` main functions.
- ploidy : The number of columns for each variable. It is the ploidy for genotypic data
- N : The number of lines in x.
- P : The number of categorical variables describing the dataset : $P = ncol(x)/ploidy$.
- N_LEVELS : The vector of the numbers of levels for the variables.
- LEVELS : The levels for the variables.
- COUNT : The observed counts of the levels.
- FREQ : The observed frequencies.

Author(s)

Wilson Toussile

See Also

`cutEachCol` in `ClustMMDD` package.

Examples

```
data(genotype1)
head(genotype1)
genotype2 = cutEachCol(genotype1[, -11], 2)
head(genotype2)
genotype3 = dataR2C(genotype2, ploidy = 2)
head (genotype3$data)
str(genotype3)
```

---

dimJump.R *Data driven calibration of the penalty function*

Description

Data driven calibration of the penalty function using the dimension jump version of the "slope heuristics".

Usage

```
dimJump.R(fileOrData, h = integer(), N = integer(), header = logical())
```

Arguments

fileOrData	A character string or a data frame (see details). If a data frame, it must contain columns named logLik and dim. If a file, it must be as the one produced by backward.explorer.
h	An integer defining the size of the sliding window used to find the biggest jump.
N	The size of the sample data (number of rows).
header	The indication of whether the file contains header or not.

Details

This function is a dimension jump version of the so called *slope heuristics* for the calibration of penalty function using the data.

Value

Assume that the penalty function is in the form

$$pen\left(K,S\right)=\alpha*\lambda*dim\left(K,S\right)$$

, where

- λ is the penalty parameter to be calibrated,
- and α a coeffcient belonging to $[1.5, 2]$, to be given by the user in model.selection.R for the final selection.

It returns a list containing two candidate values of λ and their bounds. It also produces a graphic that illustrates the "slope heuristics".

Author(s)

Wilson Toussile

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

See Also

backward.explorer for exploration of competing models space, model.selection.R for final selection.

Examples

```
# genotype2_ExploredModels was obtained via backward.explorer.
data(genotype2_ExploredModels)
outDimJump = dimJump.R(genotype2_ExploredModels, N = 1000, h = 5, header = TRUE)
outDimJump[[1]]
```

---

em.cluster.R *Compute estimates of the parameters by Expectation and Maximization algorithm.*

Description

Compute an approximation of the maximum likelihood estimates of parameters using Expectation and Maximization (EM) algorithm. A maximum a posteriori classification is then derived from the estimated set of parameters.

Usage

```
em.cluster.R(xdata, K, S, ploidy = 1, emOptions = list(epsi = NULL,
  typeSmallEM = NULL, typeEM = NULL, nberSmallEM = NULL, nberIterations = NULL,
  nberMaxIterations = NULL, putThreshold = NULL), cte = 1)
```

Arguments

xdata	A matrix of strings with the number of columns equal to ploidy * (number of variables).
K	The number of clusters (or populations).
S	The subset of clustering variables in the form of a vector of logicals indicating the selected variables. S gathers variables that are not identically distributed in at least two clusters.

`ploidy`	The number of unordered observations represented by a string in `xdata`. For example, for genotypic data from diploid individual, $ploidy = 2$.
`emOptions`	A list of EM options (see `EmOptions` and `setEmOptions`).
`cte`	A double used as a value of λ in the penalty function $pen(K, S) = \lambda * dim\,(K, S)$, where $dim\,(K, S)$ is the number of free parameters in the model defined by (K, S).

Value

A list of

- N : The size (number of lines) of the dataset.
- K : The number of clusters (populations).
- S : A vector of logicals indicating the selected variables for clustering.
- dim : The number of free parameters.
- pi_K : The vector of mixing proportions.
- prob : A list of matrices, each matrix being the probabilities of a variable in different clusters.
- logLik : The log-likelihood.
- entropy : The entropy.
- criteria : Criteria values c(BIC, AIC, ICL, CteDim).
- Tik : A stochastic matrix given the a posteriori membership probabilities.
- mapClassif : Maximum a posteriori classification.
- NbersLevels : The numbers of observed levels of the considered categorical variables.
- levels : The observed levels.

Author(s)

Wilson Toussile.

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

See Also

`dataR2C` for transformation of a classic data frame, `backward.explorer`, `selectK.R`, `dimJump.R`, `model.selection.R` for both model selection and classification.

Examples

```
data(genotype1)
head(genotype1)
genotype2 = cutEachCol(genotype1[, -11], ploidy = 2)
head(genotype2)

#See the EM options
EmOptions() # Options can be set by \code{\link{setEmOptions()}}
par5 = em.cluster.R (genotype2, K = 5, S = c(rep(TRUE, 8), rep(FALSE, 2)), ploidy = 2)
slotNames(par5)
head(par5["membershipProba"])
par5["mixingProportions"]
par5
```

EmOptions *Display the current Expectation and Maximization options.*

Description

Display the Expectation and Maximization algorithm current options.

Usage

```
EmOptions()
```

Value

A list of EM options :

- epsi : The upper bound of the relative increasing on log-likelihood.
- nberSmallEM : The number of random parameter points from which to run small EMs. The estimated parameter point associated to the higher maximum log-likelihood is then used to initialise the final EM run.
- nberIterations : The number of iterations in each small EM.
- typeSmallEM : 0 = classic EM, 1 = SEM and 2 = CEM.
- typeEM : 0 = classic EM, 1 = SEM and 2 = CEM.
- nberMaxIterations : The maximum number of iterations in the final EM if the convergence is slow.
- putThreshold : The indication of whether all parameter estimates are positive.

Author(s)

Wilson Toussile.

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

See Also

`setEmOptions` for setting EM options.

Examples

```
EmOptions()
setEmOptions(list(epsi = 1e-6))
EmOptions()
setEmOptions() # To set default values
EmOptions()
```

exModelKS *An example of* `modelKS`.

Description

An example of a set of parameters given by an instance of `modelKS`.

Format

An instance of `modelKS`.

Author(s)

Wilson Toussile

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

See Also

`modelKS`

Examples

```
data(exModelKS)
slotNames("modelKS")
head(exModelKS["membershipProba"])
exModelKS["mixingProportions"]
exModelKS
```

---

genotype1 — `genotype1` *is a data frame of genotype data with* `ploidy = 2`.

Description

A simulated data frame of genotype data with `N = 1000` individuals genotyped at `P = 10` loci. Each string represents two alleles : `ploidy = 2`. For example, "109107" represents $\{"109", "107"\}$. The last column of the data frame contains integers that represent the population membership.

Format

The format is: chr [1:1000, 1:10] "109107" "105101" "106106" ... and the 11 th column contains integers representing the prior classification in 5 sub-populations.

Author(s)

Wilson Toussile

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

See Also

`genotype2`

Examples

```
data(genotype1)
head(genotype1)
```

genotype2 *A genotype data frame compatible with ClustMMDD main functions.*

Description

This data frame can be obtained using `cutEachCol(genotype1[, -11], ploidy = 2)` (see `genotype1`).

Usage

```
data(genotype2)
```

Format

The format is: chr [1:1000, 1:20] "109" "107" "105" "101" "106" "106" "107" ..., representing observed alleles for the considered 10 loci, 2 column per locus.

Details

`ploidy = 2` for diploid individual.

Author(s)

Wilson Toussile

Source

Simulated data.

See Also

`genotype1`.

Examples

```
data(genotype2)
head(genotype2)
data(genotype1)
genotype3 = cutEachCol(genotype1[,-11], ploidy = 2)
head(genotype3)
```

genotype2_ExploredModels
A data frame of competing models gathered by `backward.explorer`.

Description

A data frame of competing models gathered by `backward.explorer` for $Kmax = 10$. Such data file can be used for a final model selection process.

Usage

```
data("genotype2_ExploredModels")
```

Format

A data frame with 2667 explored models on the following 16 variables.

`N`**:** The size of the data

`P`**:** The number of variables

`K`**:** the number of clusters

`S1`**:** 1st variable

`S2`**:** 2nd variable

`S3`**:** 3th variable

`S4`**:** 4th variable

`S5`**:** 5th variable

`S6`**:** 6th variable

`S7`**:** 7th variable

`S8`**:** 8th variable

`S9`**:** 9th variable

`S10`**:** 10th variable

`logLik`**:** The log-likelihood

`dim`**:** The dimension = number of free parameters

`entropy`**:** Entropy.

Details

TODO

Source

Wilson Toussile

See Also

dimJump.R and model.selection.R.

Examples

```
data(genotype2_ExploredModels)
head(genotype2_ExploredModels)
plot(genotype2_ExploredModels[, c("dim", "logLik")],
 col = "blue", xlab = "Dimension", ylab = "Log-likelihood")

# Data-driven calibration of the penalty
dimJump.R(genotype2_ExploredModels, h = 5, N=1000, header=T)
```

---

is.element-methods *Check if a modelKS object is in a set of such objects.*

Description

Return TRUE if an instance of modelKS belongs to a set.

Arguments

el	An instance of modelKS class.
set	A set of instances of modelKS class.

Value

TRUE if the object el belongs to a given set of modelKS.

Methods

signature(el = c("modelKS"), set = c("modelKS")) The two arguments must be vectors (see examples)

Author(s)

Wilson Toussile

Examples

```
data(exModelKS)
is.element(c(exModelKS), c(exModelKS))
is.element(c(exModelKS, 1, c(1:5)), c(exModelKS))
is.element(c(exModelKS), c(exModelKS, 1, list(1:5, 0)))
```

`is.modelKS-methods` *Is an object from class `modelKS`?*

Description

Function to test inheritance relationships between an object and a class `modelKS`.

Arguments

`object`	Any `R` object.

Value

TRUE if `object` is from class `modelKS`, and FALSE if not.

Methods

`signature(object = "modelKS")` Is an `object` from class `modelKS`?

Author(s)

Wilson Toussile

Examples

```
data(exModelKS)
is.modelKS(exModelKS)
is.modelKS(1:7)
```

`isInFile.R` *Find a model in a file.*

Description

Find a given model defined by (K, S) in a file.

Usage

```
isInFile.R(K, S, file, header)
```

Arguments

`K`	The number of clusters.
`S`	A vector of logicals of length equal to the number of variables, that indicates the clustering variables.
`file`	A file where to find the model.
`header`	A logical indicating if the file contains a header or not.

Value

A list :

- TrueFalse : A logical indicating if the given model was found and the following if TRUE.
- line : The line where the given is in the file.
- N : The size of the dataset from which the model was estimated.
- logLik : The log-likelihood.
- dim : The dimension of the model = number of free parameters.
- entropy : The entropy associated to estimated parameters of the models.

Author(s)

Wilson Toussile

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

Examples

```
data(genotype1)
head(genotype1)
genotype2 = cutEachCol(genotype1[, -11], ploidy = 2)
head(genotype2)

S = c(rep(TRUE, 8), rep(FALSE, 2))
## Not run:
outPut = selectK.R(genotype2, S, Kmax = 6, ploidy = 2, Kmin=1)
isInFile.R(K = 5, S, "genotype2_ExploredModels.txt", header = TRUE)
isInFile.R(K = 5, rep(TRUE, 10), "genotype2_ExploredModels.txt", header = TRUE)

file.remove("genotype2_ExploredModels.txt")

## End(Not run)
```

model-methods *Retrieve a list of model (K, S) from a* modelKS *object.*

Description

Recall that a model is defined by (K, S) where K is the number of clusters and S that indicates the clustering variable. This method retrieves a list of model (K, S) from a modelKS object.

Methods

`signature(object = "modelKS")` Retrieve a list of model (K, S) from a `modelKS` object.

Author(s)

Wilson Toussile.

See Also

`modelKS`, `slotNames`, `new`, `methods`, `show`

Examples

```
data(exModelKS)
showClass("modelKS")
slotNames("modelKS")
exModelKS
exModelKS["K"]
exModelKS["S"]
model(exModelKS)
```

`model.selection.R` *Selection of both the number K of clusters and the subset S of clustering variables.*

Description

The inference on both the number K of clusters and the subset S of clustering variables is seen as a model selection problem. Each competing model is characterized by one value of (K, S). The competing models are compared using penalized criteria AIC, BIC, ICL and a more general penalized criterion with a penalty function on the form

$$pen\left(K,S\right) = \alpha * \lambda * dim\left(K,S\right),$$

where

- λ is a parameter that can be calibrated using "slope-heuristics" (see `backward.explorer`, `dimJump.R`),
- and α is a coefficient in $[1.5, 2]$ to be given by the user.

Usage

```
model.selection.R(fileOrData, cte = as.double(1), alpha = as.double(2.0), header = TRUE,
  lines = integer())
```

Arguments

fileOrData — A character string or a data frame (see backward.explorer). If fileOrData is a data frame, it must contains a column named logLik and another named dim (see details).

cte — A penalty function parameter. The associated criterion is $-log(likelihood) + cte * dim$.

alpha — A coefficient in $[1.5, 2]$. The default value is 2.

header — Indication of the presence of header in the file.

lines — A vector of integer. If not empty and fileOrData is the name of a file, only models defined in lines are compared.

Value

A data frame of the selected models for the proposed penalized criteria.

Author(s)

Wilson Toussile

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

See Also

backward.explorer, dimJump.R.

Examples

```
data(genotype2_ExploredModels)
outDimJump = dimJump.R(genotype2_ExploredModels, N = 1000, h = 5, header = TRUE)
cte1 = outDimJump[[1]][1]
outSlection = model.selection.R(genotype2_ExploredModels, cte = cte1, header = TRUE)
outSlection
```

modelKS-class *modelKS is a class of parameters of (K, S) model.*

Description

modelKS is a class that can contain the set of parameters associated to a model given by (K, S).

Objects from this class

Objects can be created by calling `new("modelKS", ...)`. See `new` for more details.

Slots

`N`: The number of individuals in the datatset.

`P`: The number of random variables considered in the dataset.

`N_levels`: A vector of the numbers of levels for the considered variables.

`levels`: A `"list"` of the observed levels for the variables.

`K`: The number of clusters.

`S`: A vector of `"logical"` indicating the clustering variables.

`dim`: The dimension of a model (K, S) defined as the number of free parmaters.

`mixingProportions`: The numeric vector of the mixing proportions.

`count`: A `"list"` of the counts of levels for each variable.

`frequencies`: A `"list"` of the observed frequencies for each variable.

`proba`: A `"list"` of `"matrix"` that contains the estimates of the levels probabilities in each clusters.

`logLik`: An approximation of the maximum log-likelihood obtained by the EM algorithm.

`entropy`: The entropy given by $-\sum_{i=1}^{N}\sum_{k=1}^{K}\tau_{i,k}log\left(\tau_{i,k}\right)$, where $\tau_{i,k}$ is the probability that individual i belongs to cluster k

`membershipProba`: The `"numeric"` matrix of membership probabilities.

`mapClassification`: The maximum a posteriori classification given by a vector of `"integers"`.

Methods

== `signature(e1 = "modelKS", e2 = "ANY")`: ...

[<- `signature(x = "modelKS")`: ...

[`signature(x = "modelKS")`: ...

is.element `signature(el = "modelKS", set = "modelKS")`: ...

show `signature(object = "modelKS")`: ...

simulData `signature(object = "modelKS", N = "numeric", ploidy = "numeric")`: ...

read.modelKS `signature(file = "character")`: ...

is.modelKS `signature(object = "modelKS")`: ...

Author(s)

Wilson Toussile.

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

See Also

slotNames, new, methods, showClass.

Examples

```
data(exModelKS)
showClass("modelKS")
slotNames("modelKS")
exModelKS
exModelKS["K"]
exModelKS["S"]
model(exModelKS)
```

Rcpp Modules Examples *Functions and Objects created by Rcpp Modules Example*

Description

These function and objects are accessible from R via the Rcpp Modules mechanism which creates them based on the declaration in the C++ file.

See Also

The Rcpp Modules vignette.

---

read.modelKS-methods *Read the parameters of a model (K, S) from a file.*

Description

Read the parameters of a model (K, S) from a file, and return an instance of modelKS.

Methods

`signature()` Generic.

`signature(file = "character")` Read a set of parameters of a model (K, S) from a file.

Author(s)

Wilson Toussile.

See Also

modelKS, slotNames, new, methods, show

read.or.compute *Read a given model from a file or compute the estimates of paramaters if not found.*

Description

Read a given model from a file or compute the estimates of the parameters if not found. This function is not available for users.

Usage

```
read.or.compute(xdata, xK, xS, xReferenceModel, xReferenceModelsIndex,
  xNberExploredModels, xFileNameExploredModels, cte = as.double(1),
  header = TRUE)
```

Arguments

xdata	A list of dataset and several description paramaters such as frequencies.
xK	The number of components (clusters or populations).
xS	The subset of relevant variables.
xReferenceModel	The indicator of if the model is a reference model in an exclusion step of the backward-stepwise explorer.
xReferenceModelsIndex	The vector indicating the models that have once been a reference at an exclusion step.
xNberExploredModels	The current number of explored models.
xFileNameExploredModels	The explored models.
cte	A constant real.
header	Indication of the presence of header in the file.

Details

Not available for users.

Author(s)

Wilson Toussile

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

See Also

dimJump.R for data driven calibration of the penality function, and model.selection.R for model selection.

---

selectK.R *Selection of the number K of clusters.*

Description

Perform a selection of the number K of clusters for a given subset S of clustering variables.

Usage

```
selectK.R(xdata, S, Kmax, ploidy = 1, Kmin = 1,
  emOptions = list(epsi = 1e-05, nberSmallEM = 20, nberIterations = 15,
  nberMaxIterations = 5000, typeSmallEM = 0, typeEM = 0, putThreshold = FALSE),
  cte = 1, project = deparse(substitute(xdata)))
```

Arguments

xdata	A dataset in which data of each variable are in $ploidy$ column(s).
S	A subset of clustering variables on the form of logical vector of the same length P as the number of variables in xdata.
Kmax	The maximum number of clusters to be explored.
ploidy	The number of occurrences for each variable in the data. For example, $ploidy = 2$ for genotype
Kmin	The minimum number of clusters to be explored. The default value is set to 1.
emOptions	A list of EM options (see EmOptions and setEmOptions).
cte	A double used for the selection criterion named CteDim in which the penalty function is $pen(K, S) = cte * dim$, where dim is the number of free parameters.
project	The name of the project. The default value is the name of the dataset.

Value

A list of estimated paramaters for each selection criteria.

Author(s)

Wilson Toussile

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

See Also

backward.explorer for more exploration of the competing models space, dimJump.R for data driven calibration of the penality function, and model.selection.R for model selection.

Examples

```
data(genotype1)
head(genotype1)
genotype2 = cutEachCol(genotype1[, -11], ploidy = 2)
head(genotype2)
S = c(rep(TRUE, 8), rep(FALSE, 2))
## Not run:
outPut = selectK.R(genotype2, S, Kmax = 6, ploidy = 2, Kmin=1)
outPut[["BIC"]]

file.remove("genotype2_ExploredModels.txt")

## End(Not run)
```

setEmOptions *Set Expectation and Maximization options.*

Description

Set Expectation and Maximization options.

Usage

```
setEmOptions(emOptions = list(epsi = NULL, typeSmallEM = NULL, typeEM = NULL,
  nberSmallEM = NULL, nberIterations = NULL, nberMaxIterations = NULL,
  putThreshold = NULL))
```

Arguments

emOptions A list of options needed by the Expectation and maximization algorithm :

- epsi : In [1e-5, 1e-20], it is the upper bound of the relative increase in the log-likelihood.
- typeSmallEM : In c(0, 1, 2) : 0 = classic EM, 1 = SEM, 2 = CEM.
- typeEM : In c(0, 1, 2) : 0 = classic EM, 1 = SEM, 2 = CEM.
- nberSmallEM : The number of random parameter points from which to perform nberIterations EM runs.
- nberIterations : The number of iterations for each small EM.
- nberMaxIterations : The maximum number of iterations if EM algorithm converge hardly.
- putThreshold : If TRUE, the probabilities of levels are assumed to be positive in all clusters.

Details

Use `setEmOptions()` to set all options to default.

Author(s)

Wilson Toussile.

References

- Dominique Bontemps and Wilson Toussile (2013) : Clustering and variable selection for categorical multivariate data. Electronic Journal of Statistics, Volume 7, 2344-2371, ISSN.
- Wilson Toussile and Elisabeth Gassiat (2009) : Variable selection in model-based clustering using multilocus genotype data. Adv Data Anal Classif, Vol 3, number 2, 109-134.

See Also

`EmOptions` for getting the current EM options.

Examples

```
EmOptions()
setEmOptions(list(epsi = 1e-6))
EmOptions()
setEmOptions() # To set default values
EmOptions()
```

`setModelKS-methods` *Set an instance of class `modelKS` from a list.*

Description

Set an object of class `modelKS` from a list.

Arguments

`x`	A list from which to retrieve the slots of `modelKS`.

Value

An object of class `modelKS`.

Warning

This function is not available for users.

Author(s)

Wilson Toussile

show-methods — show *method for an object of class* modelKS

Description

Show an object of class modelKS.

Arguments

this	An object of class modelKS.

Author(s)

Wilson Toussile.

See Also

slotNames, new, methods, show

simulData-methods — *Simulate a dataset from a given set of parameters in an instance of* modelKS.

Description

Simulate a dataset from a given instance of modelKS containing a set of parameters.

Arguments

object	An instance of modelKS.
N	The size of the sample to simulate.
ploidy	The number of columns for each variable in the data. For example, $ploidy = 2$ for genotypic data from diploid individual.

Value

A list :

["data"] : The simulated dataset.

["class"] : The membership class.

Methods

`signature(object = "modelKS", N = "numeric", ploidy = "numeric")` Simulate a dataset for a given set of parameters in a modelKS object.

Author(s)

Wilson Toussile

See Also

modelKS, exModelKS.

Examples

```
data(exModelKS)
exModelKS
exData = simulData(exModelKS, 1000, 2)
str(exData)
head(exData$data)
head(exData$class)
```

[-methods *Get a slot from* modelKS.

Description

Get a slot from an object of class modelKS.

Methods

signature(x = "modelKS") See examples.

Author(s)

Wilson Toussile.

See Also

modelKS, slotNames, new, methods, show

Examples

```
data(exModelKS)
slotNames(exModelKS)
exModelKS["K"]
exModelKS["S"]
```

`[<--methods` *Get or set a slot from* `modelKS`.

Description

Get or set a slot from `modelKS`.

Methods

`signature(x = "modelKS")` See examples.

Author(s)

Wilson Toussile.

See Also

`modelKS`, `slotNames`, `new`, `methods`, `show`

Examples

```
data(exModelKS)
slotNames(exModelKS)
exModelKS["K"]
exModelKS["S"]
```

Index