Package ‘LCAvarsel’

October 12, 2022

Type Package

Title Variable Selection for Latent Class Analysis

Description Variable selection for latent class analysis for model-based clustering of multivariate categorical data. The package implements a general framework for selecting the subset of variables with relevant clustering information and discard those that are redundant and/or not informative. The variable selection method is based on the approach of Fop et al. (2017) <doi:10.1214/17-AOAS1061> and Dean and Raftery (2010) <doi:10.1007/s10463-009-0258-9>. Different algorithms are available to perform the selection: stepwise, swap-stepwise and evolutionary stochastic search. Concomitant covariates used to predict the class membership probabilities can also be included in the latent class analysis model. The selection procedure can be run in parallel on multiple cores machines.

Version 1.1

Date 2017-11-19

Author Michael Fop [aut, cre],
Thomas Brendan Murphy [ctb]

Maintainer Michael Fop <michael.fop@ucd.ie>

URL https://michaelfop.github.io/

Depends R (>= 3.4), poLCA (>= 1.4.1)

License GPL (>= 2)

Imports nnet, MASS, foreach, parallel, doParallel, GA, memoise

Suggests knitr (>= 1.12), rmarkdown (>= 1.2)

ByteCompile true

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2018-01-04 10:01:02 UTC

R topics documented:

`compareCluster` *Clustering comparison criteria*

Description

Computes some criteria for comparing two classifications of the data points.

Usage

```
compareCluster(class1, class2)
```

Arguments

`class1` A numeric or character vector of class labels.

`class2` A numeric or character vector of class labels. Must be same length of `class1`.

Details

The Jaccard, Rand and adjusted Rand indices measure the agreement between two partitions of the units. These indices vary in the interval $[0, 1]$ and a value of 1 corresponds to a perfect correspondence. Note that sometimes the adjusted Rand index could take negative values (see Hubert, Arabie, 1985). The variation of information is a measure of the distance between the two clusterings and a small value is indication of closeness.

Value

A list containing:

`tab` The confusion matrix between the two clusterings.

`jaccard` Jaccard index.

`RI` Rand index.

`ARI` Adjusted Rand index.

`varInfo` Variation of information between the two clusterings.

References

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2193-218.

Meila, M. (2007). Comparing clusterings - an information based distance. *Journal of Multivariate Analysis*, 98, 873-895.

Examples

```
cl1 <- sample(1:3, 100, replace = TRUE)
cl2 <- sample(letters[1:4], 100, replace = TRUE)
compareCluster(cl1, cl2)
compareCluster(cl1, cl1)   # perfect matching
```

`control-parameters` *Set control parameters for various purposes*

Description

Set control parameters for the EM algorithm for latent class model estimation, multinomial logistic regression estimation in the regression step, and genetic algorithm for variable selection procedure.

Usage

```
controlLCA(maxiter = 1e05, tol = 1e-04, nrep = 5)

controlReg(maxiter = 5000, tol = 1e-05)

controlGA(popSize = 20, maxiter = 100, run = maxiter/2,
          pcrossover = 0.8, pmutation = 0.2,
          elitism = base::max(1, round(popSize*0.05)))
```

Arguments

`maxiter` Maximum number of iterations in the EM algorithm, the multinomial logistic regression and the genetic algorithm.

`tol` Tolerance value for judging when convergence has been reached. Used in the EM algorithm and the multinomial logistic regression.

`nrep` Number of times to estimate the latent class analysis model, using different starting values for the matrix `z` of a posteriori probabilities. Each time, a different random initialization is used in order to search for the global maximum of the log-likelihood.

`popSize` Population size. This number corresponds to the number of different models to be considered at each iteration of the genetic algorithm.

`run` Number of consecutive generations without any improvement in the best fitness value of the variable selection procedure before the genetic algorithm is stopped.

`pcrossover` Probability of crossover between pairs of models.

`pmutation` Probability of mutation in a parent model.

`elitism` Number of best fitness models to survive at each iteration of the genetic algorithm in the variable selection procedure.

Details

Function `controlLCA` is used to set control parameters of the EM algorithm employed to estimate the latent class analysis model.

Function `controlReg` controls tolerance and maximum number of iterations in the estimation of the multinomial logistic regression. This regression is used to model the conditional distribution of a proposed variable given the current set of clustering variables in the variable selection procedure.

Function `controlGA` sets parameters of the genetic algorithm used for variable selection.

Value

A list of parameters values.

See Also

`poLCA`, `ga`

Examples

```
data(carcinoma, package = ”poLCA”)
# increase number of replicates and decrease tolerance value
fit <- fitLCA(carcinoma, ctrlLCA = controlLCA(nrep = 10, tol = 1e-07))
```

`fitLCA` *Latent class analysis model*

Description

Estimation and model selection for latent class analysis and latent class regression model for clustering multivariate categorical data. The best model is automatically selected using BIC.

Usage

```
fitLCA(Y, G = 1:3, X = NULL, ctrlLCA = controlLCA())
```

Arguments

`Y` A dataframe with (response) categorical variables. The categorical variables used to fit the latent class analysis model are converted to `factor`.

`G` An integer vector specifying the numbers of latent classes for which the BIC is to be calculated.

`X` A vector or dataframe of concomitant covariates used to predict the class-membership probability. If supplied, the number of observations of `X` must match the number of `Y`. If `NULL`, a model with no predictor variables is estimated.

`ctrlLCA` A list of control parameters for the EM algorithm used to fit the model.

Details

The function is a simple wrapper around the function `poLCA` in the homonymous package and returns less information about the estimated model. The selection of the number of latent classes is performed automatically by means of the Bayesian information criterion (BIC).

When included, covariates are used to predict the probability of class membership. In this case the model is termed as "latent class regression", or, alternatively "concomitant-variable latent class analysis". See `poLCA` for details.

Value

An object of class `'fitLCA'` providing the optimal latent class model selected by BIC.

The ouptut is a list containing:

`G` The best number of latent classes according to BIC.

`parameters` A list with the following components:

- `tau` The estimated mixing proportions.
- `theta` The estimated class conditional probabilities.

`coeff` Multinomial logit coefficient estimates on the covariates (when provided). `coeff` is a matrix with G-1 columns, and one row for each covariate. All logit coefficients are calculated for each class with respect to class 1, assumed as reference by default.

`loglik` Value of the maximized Log-likelihood.

`BIC` All BIC values computed for the range of values of *G* provided.

`bic` The optimal BIC value.

`npar` Number of estimated parameters.

`resDf` Number of residual degrees of freedom.

`z` A matrix whose `[i,g]` entry is the probability that observation `i` belongs to the gth class.

`class` Classification corresponding to the maximum a posteriori of matrix `z`.

`iter` Number of iterations.

References

Linzer, D. A. and Lewis, J. B. (2011). poLCA: An R package for polytomous variable latent class analysis. *Journal of Statistical Software* 42 1-29.

See Also

`poLCA`

Examples

```
data(gss82, package = "poLCA")
maxG(gss82, 1:7)      # not all latent class models can be fitted
fit <- fitLCA(gss82, G = 1:4)

## Not run:
# diminish tolerance and increase number of replicates
fit2 <- fitLCA(gss82, G = 1:4, ctrlLCA = controlLCA(tol = 1e-06, nrep = 10))

## End(Not run)

# the example with a single covariate as in ?poLCA
data(election, package = "poLCA")
elec <- election[, cbind("MORALG", "CARESG", "KNOWG", "LEADG", "DISHONG", "INTELG",
                         "MORALB", "CARESB", "KNOWB", "LEADB", "DISHONB", "INTELB")]
party <- election$PARTY
fit <- fitLCA(elec, G = 3, X = party)
pidmat <- cbind(1, 1:7)
exb <- exp(pidmat %*% fit$coeff)
matplot(1:7, ( cbind(1, exb)/(1 + rowSums(exb)) ),
        ylim = c(0,1), type = "l",
        main = "Party ID as a predictor of candidate affinity class",
        xlab = "Party ID: strong Democratic (1) to strong Republican (7)",
        ylab = "Probability of latent class membership",
        lwd = 2 , col = 1)
```

LCAvarsel *Variable selection for latent class analysis*

Description

Perform variable selection for latent class analysis for multivariate categorical data clustering. The function allows to find the set of variables with relevant clustering information and discard those that are redundant and/or not informative. Different searching methods can be used: stepwise backward or forward, swap-stepwise backward or forward, and stochastic evolutionary search via genetic algorithm. Concomitant covariates can be also included in the estimation of the latent class analysis model.

Usage

```
LCAvarsel(Y, G = 1:3, X = NULL,
          search = c("backward", "forward", "ga"),
          independence = FALSE,
          swap = FALSE,
          bicDiff = 0,
          ctrlLCA = controlLCA(),
          ctrlReg = controlReg(),
          ctrlGA = controlGA(),
          start = NULL,
          checkG = TRUE,
          parallel = FALSE,
          verbose = interactive())
```

Arguments

`Y` A dataframe with (response) categorical variables. The categorical variables used to fit the latent class analysis model are converted to `factor`. Missing values are not allowed and observations with `NA` entries are automatically removed.

`G` An integer vector specifying the numbers of latent classes for which the BIC is to be calculated.

`X` A vector or dataframe of concomitant covariates to be used to predict the class membership probabilities. If supplied, the number of observations of `X` must match the number of `Y` and missing values are automatically removed. If `NULL`, a model with no predictor variables is estimated. Note that the variable selection procedure does NOT perform selection of the concomitant covariates.

`search` A character vector indicating the type of search: `"backward"` starts from a model with all the available variables and at each step of the algorithm removes/adds a variable until no further change to the clustering set; `"forward"` starts from a minimum identifiable model and at each step of the algorithm adds/removes a variable until no further change to the clustering set; `"ga"` performs a stochastic search via a genetic algorithm.

`independence` A logical value indicating if, at each step of the selection algorithm, the proposed/non-clustering variables must be assumed independent from the current set of clustering variables.

`swap` A logical value indicating wheter or not a swap-stepwise search must be performed. If `TRUE`, a swap move is executed after each add and removal step. Only used when `search` is set to `"backward"` or `"forward"`.

`bicDiff` A numerical value indicating the minimum absolute BIC difference between clustering model and no clustering model used to accept the inclusion/removal of a variable into/from the set of clustering variables in the stepwise and swap-stepwise search algorithms.

`ctrlLCA` A list of control parameters for estimation of the latent class analysis model via EM algorithm; see also `controlLCA`.

`ctrlReg` A list of control parameters for the multinomial logistic regression step used to model the conditional distribution of the proposed/non-clustering variables. Only used when `independence = FALSE`; see also `controlReg`.

`ctrlGA` A list of control parameters for the genetic algorithm employed for the variable selection procedure when `search = "ga"`; see also `controlGA`.

`start` A character vector or a numeric binary matrix of initial clustering variables. When `search` is set to `"backward"` or `"forward"`, if supplied, it must be a character vector of variable names to be used as the initial clustering set. When `search = "ga"`, if provided, it must be a binary matrix of solutions indicating the set(s) of clustering variables included in the initial population of the genetic algorithm.

`checkG` A logical argument indicating if the identifiability of the latent class analysis model has to be checked for the values of `G` given in input. When `TRUE` (by default) only identifiable models according to the criterion described in `maxG` are estimated. If `FALSE`, also non identifiable models are estimated during the variable selection procedure; in this last case, *use it at your own risk!*

`parallel` A logical argument indicating if parallel computation should be used. If `TRUE`, all the available cores are used. The argument could also be set to a numeric integer value specifying the number of cores to be employed.

`verbose` A logical argument specifying wether the iterations of the variable selection procedure need to be shown or not. By default is `TRUE` if the session is interactive, `FALSE` otherwise.

Details

This function implements variable selection methods for latent class analysis for model-based clustering of multivariate categorical data. The general framework is based on a model-selection approach where the usefulness for clustering of a variable is assessed by comparing different models: a model where the variable contains relevant clustering information versus a model where it does not and it is redundant or not informative.

The model selection task corresponds to a combinatorial optimization problem and to conduct the search over the models space the following methods are available:

- *Stepwise backward/forward*. Enabled when `search = "backward"`. The algorithm starts from a model with all the variables included in the clustering set, then at each step a variable is removed/added until there is no further modification to the set of selected variables. At the start of the variable selection procedure, two consecutive removal steps are performed if `start = NULL`.
- *Stepwise forward/backward*. Enabled when `search = "forward"`. The algorithm starts from the minimum subset of variables that allows a latent class analysis model to be identified, then the variables are added/removed in turn to/from the set of clustering variables until no further change to the set of selected ones. The initial set of clustering variables is chosen by default using the strategy described in Dean and Raftery (2010); however, argument `start` can be used to provide an alternative set of initial clustering variables.
- *Swap-stepwise backward/forward*. Enabled when `search = "backward"` and `swap = TRUE`. In this case, an additional swap move is performed after each removal and addition step.
- *Swap-stepwise forward/backward*. Enabled when `search = "forward"` and `swap = TRUE`. In this case, an extra swap move is performed after each addition and removal step.
- *Stochastic evolutionary search*. Enabled when `search = "ga"`. A genetic algorithm with binary encoding is employed to search for the optimal set of clustering variables. The algorithm stops when the maximum number of iterations specified by `maxiter` has been reached or there are no further improvement in the fitness function after `run` iterations; see `controlGA`.

In the swapping step, a non-clustering variable is switched with a clustering one. The couple of variables to be swapped is selected according to their evidence of being or not being useful for clustering. This step can prevent the algorithm from getting trapped into a local sub-optimum when many correlated variables are present; however, it increases the computational cost of the variable selection procedure.

By default, at each step the variable selection procedure considers only latent class analysis models for which the identifiability condition described in `maxG` holds. When performing stepwise or swap-stepwise selection, for some combinations of clustering variables and number of classes, it could happen that a step of the variable selection procedure could not be performed because no latent class model is identifiable on any of the possible clustering sets. In such case, the step is not performed and a NA is returned. In the case of evolutionary search, non identifiable models are automatically discarded. When `checkG = FALSE`, also non identifiable models are estimated and considered during the variable selection process. Note that in this case the final output could be unreliable.

The stochastic evolutionary search implemented via the genetic algorithm allows for a better exploration of the model space. During the search, multiple sets of clustering variables are considered at the same time; then, for each set, a latent class analysis model is estimated on the clustering variables and a regression/independence model is estimated on the non-clustering ones. Different sets are generated by various genetic operators and the fittest individuals are selected. The fitness function is defined as the BIC of the joint distribution of both clustering and non-clustering variables, where clustering variables are modeled via a latent class analysis model and non-clustering variables are modeled via multinomial logistic regression or simple independent multinomial distributions in the case `independence = TRUE`. The nature of the genetic algorithm leads to a more exhaustive search, however with a larger computational cost than standard stepwise selection methods. The use of the `parallel` option allows for the estimation of multiple models in parallel and can speed up the computations.

If provided, the vector/matrix of concomitant covariates given in input in `X` is included in the latent class analysis model for the clustering variables at each step of the variable selection process. Thus, formally, a "latent class regression" model is estimated on the clustering variables (see `fitLCA`). Note that these covariates are only used to predict the class membership probabilities and no selection is performed on them.

Value

An object of class `'LCAvarsel'` containing the following components:

`variables` A character vector containing the set of selected relevant clustering variables.

`model` An object of class `'fitLCA'` corresponding to the latent class analysis model fitted on the selected variables. See `fitLCA`.

`info` A dataframe or a matrix containing information about the iterations of the variable selection procedure. If `search` is `"backward"` or `"forward"`, `info` is a dataframe with a row for each step of the algorithm and provides information regarding the type of step (Remove/Add), the name of the proposed variable, the BIC difference between the clustering model and the no clustering model for the proposed variable and the decision (Accepted/Rejected). When `search = "ga"`, `info` is a matrix containing summary statistics of the fitness function for the last 10 iterations of the genetic algorithm.

`search` A character string indicating the type of search used to perform the variable selection.

`swap` A logical value indicating if the swap move was used in the selection procedure. If `search = "ga"`, the value is `NULL`.

`independence` A logical value indicating if the proposed/non-clustering variables have been assumed independent from the current set of clustering variables during the search.

GA An object of class 'ga-class' with information about the genetic algorithm. Only present when search = "ga". See ga-class.

na A numeric vector which contains the row indices of the observations removed because of missing values. Only present when the provided data matrix X contains NAs.

References

Fop, M., and Smart, K. M. and Murphy, T. B. (2017). Variable selection for latent class analysis with application to low back pain diagnosis. *Annals of Applied Statistics*, 11(4), 2085-2115.

Dean, N. and Raftery, A. E. (2010). Latent class analysis variable selection. *Annals of the Institute of Statistical Mathematics*, 62:11-35.

Scrucca, L. (2017). On some extensions to GA package: Hybrid optimisation, parallelisation and islands evolution. *The R Journal*, 9(1), 187-206.

Scrucca, L. (2013). GA: A package for genetic algorithms in R. *Journal of Statistical Software*, 53(4), 1-3.

See Also

fitLCA, maxG

Examples

```
## Not run:
# few simple examples
data(carcinoma, package = "poLCA")
sel1 <- LCAvarsel(carcinoma)                 # Fop et al. (2017) method with no swap step
sel2 <- LCAvarsel(carcinoma, swap = TRUE)     # Fop et al. (2017) method with swap step
sel3 <- LCAvarsel(carcinoma, search = "forward",
                  independence = TRUE)        # Dean and Raftery(2010) method
sel4 <- LCAvarsel(carcinoma, search = "ga")   # stochastic evolutionary search

# an example with a concomitant covariate
data(election, package = "poLCA")
elec <- election[, cbind("MORALG", "CARESG", "KNOWG", "LEADG", "DISHONG", "INTELG",
                         "MORALB", "CARESB", "KNOWB", "LEADB", "DISHONB", "INTELB")]
party <- election$PARTY
fit <- fitLCA(elec, G = 3, X = party)
sel <- LCAvarsel(elec, G = 3, X = party, parallel = TRUE)
pidmat <- cbind(1, 1:7)
exb1 <- exp(pidmat %*% fit$coeff)
exb2 <- exp(pidmat %*% sel$model$coeff)
par(mfrow = c(1,2))
matplot(1:7, ( cbind(1, exb1)/(1 + rowSums(exb1)) ),
        ylim = c(0,1), type = "l",
        main = "Party ID as a predictor of candidate affinity class",
        xlab = "Party ID: strong Democratic (1) to strong Republican (7)",
        ylab = "Probability of latent class membership",
        lwd = 2 , col = 1)
matplot(1:7, ( cbind(1, exb2)/(1 + rowSums(exb2)) ),
        ylim = c(0,1), type = "l",
        main = "Party ID as a predictor of candidate affinity class",
        xlab = "Party ID: strong Democratic (1) to strong Republican (7)",
        ylab = "Probability of latent class membership",
        lwd = 2 , col = 1)
# compare
compareCluster(fit$class, sel$model$class)

## End(Not run)
```

maxG *Maximum number of latent classes*

Description

Finds the number of latent classes that are allowed to be fitted on a dataset in order for the latent class analysis model to be identifiable.

Usage

```
maxG(Y, Gvec)
```

Arguments

Y	A categorical data matrix.
Gvec	A numeric vector denoting the range of number of latent classes to be fitted.

Details

In practice, different latent class analysis models are fitted by attributing different values to G, usually ranging from 1 to G_{max}. However, for a set of variables, not all the models corresponding to increasing values of G are identifiable. Indeed, a necessary (but not sufficient) condition for a latent class analysis model to be identifiable is:

$$\prod_{j=1}^{M} C_j > G\left(\sum_{j=1}^{M} C_j - M + 1\right)$$

where C_j denotes the number of categories of variable j, $j = 1, ..., M$, and M is the number of variables in the data `Y`. Another condition requires the number of observed distinct configurations of the variables in the data to be greater than the number of parameters of the model. The function returns the subset of values of vector `Gvec` such that both the above conditions are satisfied.

Value

A numeric vector containing the subset of number of latent classes that are allowed to be fitted on the data in order for the model to be identifiable. If no model is identifiable for the range of values provided, the function returns `NULL` and throws a warning.

References

Bartholomew, D. and Knott, M. and Moustaki, I. (2011). Latent Variable Models and Factor Analysis: A Unified Approach. *Wiley*.

Goodman, L. A. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*. 61, 215-231.

Examples

```
data(carcinoma, package = "poLCA")
maxG(carcinoma, 1:4)
maxG(carcinoma, 2:3)
maxG(carcinoma, 5)     # the model is not identifiable
```

Index