Package 'ContaminatedMixt'

May 7, 2026

Type Package

Title Clustering and Classification with the Contaminated Normal

Version 1.3.8

Date 2023-05-05

Author Antonio Punzo, Angelo Mazza, Paul D. McNicholas

Maintainer Angelo Mazza <a.mazza@unict.it>

Description Fits mixtures of multivariate contaminated normal distributions (with eigen-decomposed scale matrices) via the expectation conditional-maximization algorithm under a clustering or classification paradigm Methods are described in Antonio Punzo, Angelo Mazza, and Paul D McNicholas (2018) <doi:10.18637/jss.v085.i10>.

License GPL-2

LazyLoad yes

Depends R (>= 2.15.0)

Imports mixture, mnormt, mclust, caret, mvtnorm, grDevices

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-05-05 16:20:02 UTC

Contents

`ContaminatedMixt-package`
ContaminatedMixt - Parsimonious Mixtures of Contaminated Normal Distributions

---

Description

This package allows to fit, according to the expectation-conditional maximization algorithm, the 14 parsimonious mixtures of multivariate contaminated normal distributions, with eigen-decomposed scale matrices, introduced by Punzo and McNicholas (2016). Model-based clustering and classification scenarios are implemented. Likelihood-based model selection criteria can be adopted to select the parsimonious model and the number of groups.

Details

Package:	ContaminatedMixt
Type:	Package
Version:	1.1
Date:	2016-09-02
License:	GNU-2

Author(s)

Antonio Punzo, Angelo Mazza, Paul D. McNicholas

Maintainer: Angelo Mazza <a.mazza@unict.it>

References

Punzo A., Mazza A. and McNicholas P. D. (2018). **ContaminatedMixt**: An R Package for Fitting Parsimonious Mixtures of Multivariate Contaminated Normal Distributions. *Journal of Statistical Software*, **85**(10), 1–25.

Punzo A. and McNicholas P. D. (2016). Parsimonious mixtures of multivariate contaminated normal distributions. *Biometrical Journal*, **58**(6), 1506–1537.

See Also

`CNmixt`

agree *Agreement Between Partitions*

Description

Evaluates the agreement of a given partition with respect to the partition arising from the mixture of multivariate contaminated normal distributions. If the mixture has been fitted for classification purposes, the agreement will be based on the unlabeled observations only.

Usage

```
agree(object, givgroup, criterion = "BIC")
```

Arguments

`object`	An object of class `ContaminatedMixt`
`givgroup`	vector, of the same dimension of the number of observations used to fit the model in `object`, representing a given partition
`criterion`	an optional character string with the information criterion to consider; supported values are: `"AIC"`, `"AICc"`, `"AICu"`, `"AIC3"`, `"AWE"`, `"BIC"`, `"CAIC"`, `"ICL"`. Default value is `"BIC"`.

Value

A contingency table.

Author(s)

Antonio Punzo, Angelo Mazza, Paul D. McNicholas

References

Punzo A., Mazza A. and McNicholas P. D. (2018). **ContaminatedMixt**: An R Package for Fitting Parsimonious Mixtures of Multivariate Contaminated Normal Distributions. *Journal of Statistical Software*, **85**(10), 1–25.

Punzo A. and McNicholas P. D. (2016). Parsimonious mixtures of multivariate contaminated normal distributions. *Biometrical Journal*, **58**(6), 1506–1537.

See Also

`ContaminatedMixt-package`, `CNmixt`

CNmixt *Fitting for the Parsimonious Mixtures of Contaminated Normal Distributions*

Description

Fits, by using the expectation conditional-maximization (ECM) algorithm, parsimonious mixtures of multivariate contaminated normal distributions (with eigen-decomposed scale matrices) to the given data within a clustering paradigm (default) or classification paradigm. Can be run in parallel. Likelihood-based model selection criteria are used to select the parsimonious model and the number of groups.

Usage

```
CNmixt(X, G, contamination = NULL, model = NULL,
   initialization = "mixt", alphafix = NULL, alphamin = 0.5,
   seed = NULL, start.z = NULL, start.v = NULL, start = 0,
   label = NULL, AICcond = FALSE, iter.max = 1000,
   threshold = 1.0e-10, parallel = FALSE, eps = 1e-100,verbose = TRUE)
CNmixtCV(X, G, contamination = NULL, model = NULL,
   initialization = "mixt", k = 10,alphafix = NULL,
   alphamin = 0.5, seed = NULL, start.z = NULL, start.v = NULL,
   start = 0, label = NULL, iter.max = 1000, threshold = 1.0e-10,
   parallel = FALSE, eps = 1e-100, verbose = TRUE)
```

Arguments

X	a dim=c(n,p) matrix such that the n rows correspond to observations and the p columns correspond to variables.
G	a vector containing the numbers of groups to be tried.
contamination	an optional boolean indicating if the model(s) to be fitted have to be contaminated or not. If NULL, then both types of models are fitted.
model	a vector indicating the model(s) to be fitted. In the multivariate case ($p > 1$), possible values are: "EII", "VII", "EEI", "VEI", "EVI", "VVI", "EEE", "VEE", "EVE", "EEV", "VVE", "VEV", "EVV", "VVV". If NULL, then all 14 models are fitted. In the univariate case ($p = 1$), possible values are "E" and "V".
initialization	initialization strategy for the ECM algorithm. It can be: • "mixt" (default): the initial ($n \times G$) matrix with posterior probabilities of groups membership arises from a preliminary run of mixtures of multivariate normal distributions as fitted by the gpcm() function of the **mixture** package (see mixture:gpcm for details). • "kmeans": the initial ($n \times G$) hard classification matrix arises from a preliminary run of the k-means algorithm; • "random.post": the initial ($n \times G$) matrix with posterior probabilities of groups membership is randomly generated;

- "random.clas": the initial ($n \times G$) classification matrix is randomly generated;
- "manual": the user must specify either the initial ($n \times G$) classification matrix or the initial ($n \times G$) matrix with posterior probabilities of groups membership, via the argument start.z and, optionally, the initial ($n \times G$) matrix of posterior probabilities to be a good observation in each group, via the argument start.v.

alphafix — a vector of length G with the proportion of good observations in each group. If length(alphafix) != G, then the first element is replicated G times. Default value is NULL.

alphamin — a vector of length G with the minimum proportion of good observations in each group. If length(alphamin) != G, then the first element is replicated G times. Default value is 0.5.

seed — the seed for the random number generator, when random initializations are used; if NULL, current seed is not changed. Default value is NULL.

start.z — initial $n \times G$ matrix of either soft or hard classification. Default value is NULL.

start.v — initial $n \times G$ matrix of posterior probabilities to be a good observation in each group. Default value is a $n \times G$ matrix of ones.

start — when initialization = "mixt", initialization used for the gpcm() function of the **mixture** package (see mixture:gpcm for details).

label — a vector of integers of length equal to the number of rows of X. It indicates the known group of membership of each observation. Use 0 when membership is not known. Use NULL when membership is unknown for all observations.

AICcond — When TRUE, the AICcond criterion, an estimate of the predictive ability of a generative model for classification, is computed (Vandewalle et al., 2013).

iter.max — maximum number of iterations in the ECM algorithm. Default value is 1000.

threshold — threshold for Aitken's acceleration procedure. Default value is 1.0e-03.

parallel — When TRUE, the package parallel is used for parallel computation. When several models are estimated, computational time is reduced. The number of cores to use may be set with the global option cl.cores; default value is detected using detectCores().

eps — an optional scalar. It sets the smallest value for the eigenvalues of the component scale matrices. Default value is 1e-100.

k — number of equal sized subsamples used in k-fold cross-validation.

verbose — write text to the console

Details

The multivariate data contained in X are either clustered or classified using parsimonious mixtures of multivariate contaminated normal distributions with some or all of the 14 parsimonious models described in Punzo and McNicholas (2016). Model specification (via the model argument) follows the nomenclature popularized in other packages such as **mixture** and **mclust**. Such a nomenclature refers to the decomposition and constraints on the scale matrix (see Banfield and Raftery, 1993, Celeux and Govaert, 1995 and Punzo and McNicholas, 2016 for details):

$$\Sigma_g = \lambda_g \Gamma_g \Delta_g \Gamma_g'.$$

The nomenclature describes (in order) the volume (λ_g), shape (Δ_g), and orientation (Γ_g), in terms of "V"ariable, "E"qual, or the "I"dentity matrix. As an example, the string "VEI" would refer to the model where $\Sigma_g = \lambda_g \Delta$. Note that for $G = 1$, several models are equivalent (for example, "EEE" and "VVV"). Thus, for $G = 1$ only one model from each set of equivalent models will be run.

The algorithms detailed in Celeux and Govaert (1995) are considered in the first CM-step of the ECM algorithm to update Σ_g for all the models apart from "EVE" and "VVE". For "EVE" and "VVE", majorization-minimization (MM) algorithms (Hunter and Lange, 2000) and accelerated line search algorithms on the Stiefel manifold (Absil, Mahony and Sepulchre, 2009 and Browne and McNicholas, 2014), which are especially preferable in higher dimensions (Browne and McNicholas, 2014), are used to update Σ_g; the same approach is also adopted in the **mixture** package for those models.

Starting values are very important to the successful operation of these algorithms and so care must be taken in the interpretation of results. All the initializations considered here provide initial quantities for the first CM-step of the ECM algorithm. The predictive ability of a model for classification may be estimated using the cross-validated error rate, returned by CNmixtCV or trough the the AIC-cond criterion (Vandewalle et al., 2013).

Value

CNmixt returns an object of class ContaminatedMixt. CNmixtCV returns a list with the cross-validated error rate estimated for each model.

Author(s)

Antonio Punzo, Angelo Mazza, Paul D. McNicholas

References

Absil P. A., Mahony R. and Sepulchre R. (2009). Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton, NJ.

Banfield J. D. and Raftery A. E. (1993). Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics*, **49**(3), 803–821.

Browne R. P. and McNicholas P. D. (2013). Estimating Common Principal Components in High Dimensions. *Advances in Data Analysis and Classification*, **8**(2), 217–226.

Browne, R. P. and McNicholas P. D. (2014). Orthogonal Stiefel manifold optimization for eigen-decomposed covariance parameter estimation in mixture models. *Statistics and Computing*, **24**(2), 203–210.

Browne R. P. and McNicholas P. D. (2015). **mixture**: Mixture Models for Clustering and Classification. R package version 1.4.

Celeux G. and Govaert G. (1995). Gaussian Parsimonious Clustering Models. *Pattern Recognition*. **28**(5), 781–793.

Hunter D. R. and Lange K. (2000). Rejoinder to Discussion of "Optimization Transfer Using Surrogate Objective Functions". *Journal of Computational and Graphical Statistics*, **9**(1), 52–59.

Punzo A., Mazza A. and McNicholas P. D. (2018). **ContaminatedMixt**: An R Package for Fitting Parsimonious Mixtures of Multivariate Contaminated Normal Distributions. *Journal of Statistical Software*, **85**(10), 1–25.

Punzo A. and McNicholas P. D. (2016). Parsimonious mixtures of multivariate contaminated normal distributions. *Biometrical Journal*, **58**(6), 1506–1537.

Vandewalle V., Biernacki C., Celeux G. and Govaert G. (2013). A predictive deviance criterion for selecting a generative model in semi-supervised classification. *Computational Statistics and Data Analysis*, **64**, 220–236.

See Also

ContaminatedMixt-package

Examples

```
## Note that the example is extremely simplified
## in order to reduce computation time

# Artificial data from an EEI Gaussian mixture with G = 2 components

library("mnormt")
p <- 2
set.seed(12345)
X1 <- rmnorm(n = 200, mean = rep(2, p), varcov = diag(c(5, 0.5)))
X2 <- rmnorm(n = 200, mean = rep(-2, p), varcov = diag(c(5, 0.5)))
noise <- matrix(runif(n = 40, min = -20, max = 20), nrow = 20, ncol = 2)
X <- rbind(X1, X2, noise)

group <- rep(c(1, 2, 3), times = c(200, 200, 20))
plot(X, col = group, pch = c(3, 4, 16)[group], asp = 1, xlab = expression(X[1]),
ylab = expression(X[2]))

# ---------------------- #
# Model-based clustering #
# ---------------------- #

res1 <- CNmixt(X, model = c("EEI", "VVV"), G = 2, parallel = FALSE)

summary(res1)

agree(res1, givgroup = group)

plot(res1, contours = TRUE, asp = 1, xlab = expression(X[1]), ylab = expression(X[2]))

# -------------------------- #
# Model-based classification #
# -------------------------- #

indlab <- sample(1:400, 20)
lab <- rep(0,nrow(X))
lab[indlab] <- group[indlab]
res2 <- CNmixt(X, G = 2, model = "EEI", label = lab)


agree(res2, givgroup = group)
```

CNpredict *Cluster Prediction*

Description

Cluster prediction for multivariate observations based on uncontaminated/contaminated normal mixture models

Usage

```
CNpredict(newdata, prior, mu, invSigma, eta=NULL, alpha=NULL)
## S3 method for class 'ContaminatedMixt'
predict(object, newdata, ...)
```

Arguments

newdata	a dim=c(n,p) matrix representing the coordinates of n new data point(s)
object	an object of class ContaminatedMixt resulting from a call to CNmixt. When several models have been estimated, getBestModel is used to select one of them
...	Options to be passed to getBestModel
prior	a vector with length=G, where G is the number of components of the mixture model. Its kth component is the mixing proportion for the kth component
mu	a dim=c(p,G) matrix with mean values for each component of the mixture model
invSigma	an array with dim=c(p,p,G) whose element invSigma[,,k] is the inverse covariance matrix for the kth component of the mixture model.
alpha	a vector of length=G with the proportions of good observations; it must be a number between 0 and 1. Use NULL for uncontaminated models
eta	a vector of length=G with the degree of contamination; it should be a number greater than 1. Use NULL for uncontaminated models

Value

a vector with group membership

Author(s)

Antonio Punzo, Angelo Mazza, Paul D. McNicholas

References

Punzo A., Mazza A. and McNicholas P. D. (2018). **ContaminatedMixt**: An R Package for Fitting Parsimonious Mixtures of Multivariate Contaminated Normal Distributions. *Journal of Statistical Software*, **85**(10), 1–25.

Punzo A. and McNicholas P. D. (2016). Parsimonious mixtures of multivariate contaminated normal distributions. *Biometrical Journal*, **58**(6), 1506–1537.

See Also

ContaminatedMixt-package

Examples

```
point <- c(0,0,0)
mu <- c(1,-2,3)
Sigma <- diag(3)
alpha <- 0.8
eta <- 5
f <- dCN(point, mu, Sigma, alpha, eta)
x <- rCN(10, mu, Sigma, alpha, eta)
```

---

dCN *Multivariate Contaminated Normal Distribution*

Description

Probability density function and random number generation for the multivariate contaminated normal distribution.

Usage

```
dCN(x, mu = rep(0,p), Sigma, alpha = 0.99, eta = 1.01)
rCN(n, mu = rep(0,p), Sigma, alpha = 0.99, eta = 1.01)
```

Arguments

x	either a vector of length p or a matrix with p columns, being p = ncol(Sigma), representing the coordinates of the point(s) where the density must be evaluated
mu	either a vector of length p, representing the mean value, or (except for rCN) a matrix whose rows represent different mean vectors; if it is a matrix, its dimensions must match those of x
Sigma	a symmetric positive-definite matrix representing the scale matrix of the distribution; a vector of length 1 is also allowed (in this case, p = 1 is set)
alpha	proportion of good observations; it must be a number between 0 and 1
eta	degree of contamination; it should be a number greater than 1
n	the number of random vectors to be generated

Value

dCN returns a vector of density values; rCN returns a matrix of n rows of random vectors

Author(s)

Antonio Punzo, Angelo Mazza, Paul D. McNicholas

References

Punzo A., Mazza A. and McNicholas P. D. (2018). **ContaminatedMixt**: An R Package for Fitting Parsimonious Mixtures of Multivariate Contaminated Normal Distributions. *Journal of Statistical Software*, **85**(10), 1–25.

Punzo A. and McNicholas P. D. (2016). Parsimonious mixtures of multivariate contaminated normal distributions. *Biometrical Journal*, **58**(6), 1506–1537.

See Also

ContaminatedMixt-package

Examples

```
point <- c(0,0,0)
mu <- c(1,-2,3)
Sigma <- diag(3)
alpha <- 0.8
eta <- 5
f <- dCN(point, mu, Sigma, alpha, eta)
x <- rCN(10, mu, Sigma, alpha, eta)
```

Extractor functions *Extractors for* ContaminatedMixt *Class Objects.*

Description

These functions extract values from ContaminatedMixt class objects.

Usage

```
getBestModel(object, criterion = "BIC", G = NULL, model = NULL,
            contamination = NULL)
getPosterior(object, ...)
getSize(object, ...)
getCluster(object, ...)
getPar(object, ...)
getCV(object)
getIC(object,criteria)
getDetection(object,...)
whichBest(object, criteria = NULL, G = NULL, model = NULL,
  contamination = NULL)
whichBestCV (object, G=NULL, model=NULL, contamination=NULL)
## S3 method for class 'ContaminatedMixt'
summary(object, criterion = "BIC",
  digits = getOption("digits")-2, ...)
## S3 method for class 'ContaminatedMixt'
print(x, ...)
```

Arguments

object, x	a class ContaminatedMixt object or a a class ContaminatedMixt object for getCV and whichBestCV.
criterion	a string with the information criterion to consider; supported values are: "AIC", "AICc", "AICcond", "AICu", "AIC3", "AWE", "BIC", "CAIC", "ICL". Default value is "BIC".
criteria	a vector of strings with the names of information criteria to consider. If NULL, all the supported information criteria are considered.
G	an optional vector containing the numbers of groups to consider. If not specified, all the estimated models are considered.
model	an optional vector of character strings indicating the parsimonious models to consider. If not specified, all the estimated models are considered.
contamination	an optional boolean indicating if the model(s) to be considered have to be contaminated or not. If NULL, then both types of models are considered.
digits	integer used for number formatting.
...	additional arguments to be passed to getBestModel (or to whichBest for the print method).

Details

When several models have been estimated, these functions consider the best model according to the information criterion in criterion, among the estimated models having a number of components among those in G and a parsimonious model among those in model. whichBestCV considers the best model according to the cross-validated error rates computed by CNmixtCV. getIC provides values for the information criteria in criteria.

The getBestModel method returns a ContaminatedMixt object containing the best model only, selected as described above.

---

m.step *M-step of the EM algorithm for Parsimonious Normal Mixtures*

Description

Carries out the M-step for EM algorithm

Usage

```
m.step(X, modelname, z, mtol=1e-10, mmax=10)
```

Arguments

X	a matrix such that n rows correspond to observations and p columns correspond to variables.
modelname	A three letter sequence indicating the covariance structure. Possible values are: "EII", "VII", "EEI", "VEI", "EVI", "VVI", "EEE", "VEE", "EVE", "EEV", "VVE", "VEV", "EVV", "VVV".
z	A matrix of weights such that n rows correspond to observations and G columns correspond to groups.
mtol	The convergence criteria for the M-step if an iterative procedure is necessary.
mmax	The maximum number of iterations for an iterative procedure.

Value

A list of the model parameters with the mu, Sigma, invSigma and px for each group.

Author(s)

Antonio Punzo, Angelo Mazza, Paul D. McNicholas

References

Punzo A., Mazza A. and McNicholas P. D. (2018). **ContaminatedMixt**: An R Package for Fitting Parsimonious Mixtures of Multivariate Contaminated Normal Distributions. *Journal of Statistical Software*, **85**(10), 1–25.

Punzo A. and McNicholas P. D. (2016). Parsimonious mixtures of multivariate contaminated normal distributions. *Biometrical Journal*, **58**(6), 1506–1537.

See Also

ContaminatedMixt-package

Examples

```
point <- c(0,0,0)
mu <- c(1,-2,3)
Sigma <- diag(3)
alpha <- 0.8
eta <- 5
f <- dCN(point, mu, Sigma, alpha, eta)
x <- rCN(10, mu, Sigma, alpha, eta)
```

pairs.ContaminatedMixt *Scatterplot Matrix for ContaminatedMixt Objects*

Description

A matrix of scatterplots, for objects of class ContaminatedMixt, is produced.

Usage

```
## S3 method for class 'ContaminatedMixt'
pairs(x, criterion = "BIC", ...)
```

Arguments

x	an object of class ContaminatedMixt
criterion	an optional character string with the information criterion to consider; supported values are: "AIC", "AICc", "AICu", "AIC3", "AWE", "BIC", "CAIC", "ICL". Default value is "BIC".
...	Options to be passed to pairs.

Author(s)

Antonio Punzo, Angelo Mazza, Paul D. McNicholas

See Also

ContaminatedMixt

plot.ContaminatedMixt *Scatterplot for ContaminatedMixt Objects*

Description

Scatterplot, with optionally superimposed contours, for objects of class ContaminatedMixt.

Usage

```
## S3 method for class 'ContaminatedMixt'
plot(x, criterion = "BIC", contours = FALSE, xmarg = 1, ymarg = 2,
                    res = 200, levels = seq(.0001,1,by=0.01), ...)
```

Arguments

x	an object of class ContaminatedMixt
criterion	a string with the information criterion to consider; supported values are: "AIC", "AICc", "AICu", "AIC3", "AWE", "BIC", "CAIC", "ICL". Default value is "BIC".
contours	if TRUE, the contours of the mixture density are superimposed on the plot. Default is FALSE.
xmarg	scalar argument giving the position of the variable to be used on the x-axis.
ymarg	scalar argument giving the position of the variable to be used on the y-axis.
res	scalar argument giving the resolution for the calculation grid required for the contour plot. Default is 200, which results in a 200×200 grid.
levels	Numeric vector giving the levels at which contours should be drawn. Default is to draw a contour in 0.01 steps, starting from the contour of height .0001. This may result in more/less contours than desired depending on the resulting density.
...	Options to be passed to plot.

Author(s)

Antonio Punzo, Angelo Mazza, Paul D. McNicholas

See Also

CNmixt

---

wine *Wine Data Set*

Description

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wine: Barolo, Grignolino, Barbera. The data set is used to evaluate the ability of the CNmixt() function in clustering the data assuming unknown their cultivars.

Usage

```
data(wine)
```

Format

This data frame contains 178 rows, each corresponding to a different cultivar of wine produced in Piedmont (Italy), and 14 columns. The first column is the type of wine (Type), a factor variable with the following levels: Barolo, Grignolino, Barbera. The variables measured on the three types of wines are the following: Alcohol, Malic acid, Ash, Alcalinity, Magnesium, Phenols, Flavanoids, Nonflavanoids, Proanthocyanins, Color intensity, Hue, OD280.OD315Dilution, Proline. All variables but the label class are continuous.

Details

The original data set comprises 27 variables. Here a subset of 14 variables only has been included.

Source

This dataset is from the UCI machine learning repository and it is available at http://archive.ics.uci.edu/ml/datasets/Wine.

References

Forina M., Lanteri S. Armanino C., Casolino C., Casale M., Oliveri, P. (2008). V-PARVUS. *An Extendible Pachage of programs for esplorative data analysis, classification and regression analysis*. Dip. Chimica e Tecnologie Farmaceutiche ed Alimentari, Universita' di Genova.

See Also

ContaminatedMixt-package, CNmixt

Examples

```
data("wine")

group <- wine[, 1]
pairs(wine[, -1], cex = 0.6, pch = c(2, 3, 1)[group], col = c(3, 4, 2)[group], gap = 0,
cex.labels = 0.6)

res3 <- CNmixt(wine[, -1], G = 3, model = "EEE", initialization = "random.post",
  seed = 5, parallel = FALSE)
agree(res3, givgroup = group)
pairs(res3, cex = 0.6, gap = 0, cex.labels = 0.6)
```

Index