Package ‘upclass’

February 20, 2015

Type Package
Title Updated Classification Methods using Unlabeled Data
Depends mclust
Suggests classifly, gclus
Version 2.0
Date 2013-11-26
Author Niamh Russell, Laura Cribbin, Thomas Brendan Murphy
Maintainer Niamh Russell <niamh.russell@gmail.com>
Description This package contains a collection of functions which implement data classification. It uses unlabeled data to obtain parameter estimates of models. The functions can be implemented over a number of models with the best model selected and displayed.
License GPL-2
Repository CRAN
NeedsCompilation no
Date/Publication 2014-09-03 15:30:38

R topics documented:

upclass-package .................................................. 2
Aitken .......................................................... 3
modelvec ......................................................... 4
noupclassify ................................................... 5
noupclassifymodel .............................................. 7
plot.upclassfit .................................................. 9
print.upclassfit ............................................... 10
summary.upclassfit ............................................. 11
upclassify ....................................................... 13
upclassifymodel ................................................ 15

Index 19
Description

This package contains a collection of functions which implement data classification. It creates updated classification rules by making use of unlabeled data when obtaining parameter estimates of models. The functions can be implemented over a number of models with the best model selected and displayed.

Details

Package: upclass
Type: Package
Version: 2.0
Date: 2013-11-26
License: GPL-2
LazyLoad: yes

The function upclassifymodel takes an updated approach to typical classification rules on unlabelled data. It obtains initial parameter estimates and membership probabilities using the labeled data only and then iterates through the EM algorithm using the complete data with continuous updating of estimates and probabilities. The example below shows graphically the goodness of fit of such a model using this updated approach and a typical classification method, upclassify. The function upclassify implements upclassifymodel over a desired list of models fitted to the data. The model that best fits the data is returned. For a complete list of function, use library(help="upclass").

Author(s)

Niamh Russell, Laura Cribbin, Thomas Brendan Murphy
Maintainer: Niamh Russell <niamh.russell.1@ucdconnect.ie>

References


Examples

data(iris)
X<- as.matrix(iris[,,-5])
cl<-as.matrix(iris[,5])

indtrain <- sort(sample(1:150, 30))
Xtrain <- X[indtrain,]
cltrain <- cl[indtrain]
Aitken Acceleration

Description

Calculates the Aitken acceleration estimate of the final converged maximized log-likelihood.

Usage

Aitken(ll)

Arguments

ll A vector of three consecutive log-likelihood values

Details

The final converged maximized log-likelihood can be used to determine convergence, i.e., for use in determining convergence of the EM algorithm. It can be used alongside other functions in the upclass package to decide whether or not the log-likelihood has converged.

Value

The return value as a list with the following components:

ll The most current estimate for the log-likelihood
linf An estimate of the final converged maximized log-likelihood
a The Aitken acceleration value where 0 <= a <= 1

References


See Also

upclassifymodel, noupclassifymodel.
Examples

li<-c(-261, -257.46,-256.4)
Aitken(ll)

modelvec  Univariate and Multivariate Model Names

Description

Model names to be used in the upclass package for univariate and multivariate data.

Usage

modelvec(d = 1)

Arguments

d  The dimension of the data. By default, d=1, and the data is considered to be univariate.

Value

if d=1, returned is a vector with the first two of the following components only; otherwise, they are omitted and the vector contains the remaining components:

"E"  Univariate, equal variance
"V"  Univariate, variable variance
"EII"  Multivariate, equal volume and spherical
"VII"  Multivariate, variable volume and spherical
"EEI"  Multivariate, equal volume, equal shape and axis aligned
"VEI"  Multivariate, variable volume, equal shape and axis aligned
"EVI"  Multivariate, equal volume, variable shape and axis aligned
"VVI"  Multivariate, variable volume, variable shape and axis aligned
"EEE"  Multivariate, equal volume, equal shape and equal orientation
"EEV"  Multivariate, equal volume, equal shape and variable orientation
"VEV"  Multivariate, variable volume, equal shape and variable orientation
"VVV"  Multivariate, variable volume, variable shape and variable orientation

References


noupclassify

See Also

upclassify, noupclassify.

Examples

modelvec(1) # Models available for univariate data.

data(iris)
modelvec(ncol(iris[, -5])) # Models available for multivariate data

Description

This function performs supervised classification over a range of different models and finds the model that best fits the data. In selecting the best model, the BIC values are compared.

Usage

noupclassify(Xtrain, cltrain, Xtest, cltest = NULL, modelscope = NULL, ...)

Arguments

Xtrain A numeric matrix of data where rows correspond to observations and columns correspond to variables. The group membership of each observation is known - labeled data.

cltrain A numeric vector with distinct entries representing a classification of the corresponding observations in Xtrain.

Xtest A numeric matrix of data where rows correspond to observations and columns correspond to variables. The group membership of each observation may not be known - unlabeled data.

cltest A numeric vector with distinct entries representing a classification of the corresponding observations in Xtest. By default, these are not supplied and the function sets out to obtain them.

modelscope A character string indicating the desired models to be tested. With default NULL, all available models are tested. The models available for univariate and multivariate data are described in modelvec.

... Arguments passed to or from other methods.
Value

An object of class "upclassfit" providing a list of output components for each model in modelscope, with the Best model (according to BIC) first. The details of the output components are as follows:

call  How to call the function and the order of its arguments.
Ntrain The number of observations in the training set.
Ntest The number of observations in the test set.
d The dimension of the data.
G The number of groups in the training set.
modelName The model considered in this run of the algorithm.
parameters A list of the model parameters estimated by Mclust.
pro The proportion of the data to be found in each group.
mean Mean vectors for each group.
variance The variance and covariances produced by Mclust.
train A list of information about the training data. This will not have changed from before the run.
  z A matrix containing the estimated probabilities that each observation in the training data belongs to each group.
c1 A vector containing the labels of the training data.
misclass The number of misclassifications of the training data.
rate The misclassification rate expressed as a percentage.
Brier The Brier score expressed as a percentage.
tab The misclassification table for the training data.
test A list of information about the test data.
  z A matrix containing the estimated probabilities that each observation in the training data belongs to each group.
c1 A vector containing the new labels of the training data.
misclass The number of misclassifications of the training data, provided the correct labels have been supplied.
rate The misclassification rate expressed as a percentage, provided the correct labels have been supplied.
Brier The Brier score expressed as a percentage.
tab The misclassification table for the training data, provided the correct labels have been supplied.
ll The log-likelihood of the data.
bic The Bayes information criterion for the specified model.

Author(s)

Niamh Russell
References


C. Fraley and A.E. Raftery (2002). Model-based clustering, discriminant analysis, and density

C. Fraley and A.E. Raftery (2006) MCLUST Version 3 for R: Normal Mixture Modeling and
Model-Based Clustering, Technical Report no. 504, Department of Statistics, University of Wash-
ington

See Also

upclassify, noupclassifymodel, modelvec

Examples

data(iris)
x<- as.matrix(iris[, -5])
cl<-as.matrix(iris[, 5])

indtrain <- sort(sample(1:150, 30))
xtrain <- x[indtrain,]
cltrain <- cl[indtrain]

indtest <- setdiff(1:150, indtrain)
xtest <- x[indtest,]
cltest <- cl[indtest]

fitnoupmodels <- noupclassify(xtrain, cltrain, xtest, cltest) #testing every model.
fitnoupmodels$Best$modelName

noupclassifymodel Classification Method using labeled data only

Description

noupclassifymodel implements the EM algorithm to classify unlabeled data using parameter es-
timates derived from labeled data only. It is a background function not designed to be used directly.

Usage

noupclassifymodel(Xtrain, cltrain, Xtest, cltest = NULL, modelName = "EEE", ...)
Arguments

**Xtrain**
A numeric matrix of observations where rows correspond to observations and columns correspond to variables. The group membership of each observation is known - labeled data.

**cltrain**
A numeric vector with distinct entries representing a classification for the corresponding observations in Xtrain.

**Xtest**
A numeric matrix of observations where rows correspond to observations and columns correspond to variables. The group membership of each observation may not be known - unlabeled data.

**cltest**
A numeric vector with distinct entries representing a classification of the corresponding observations in Xtest. By default, these are not supplied and the function sets out to determine these.

**modelName**
A character string indicating the model, with default "EEB". The models available for univariate and multivariate data are described in modelvec.

... Arguments passed to or from other methods.

Value

The return value is a list with the following components:

**call**
The function call from noupclassifymodel or upclassifymodel

**ntrain**
The number of observations in the training data.

**ntest**
The number of observations in the test data.

**d**
The dimension of the data.

**g**
The number of groups in the data.

**modelName**
A character string identifying the model (same as the input argument)

**parameters**

- **pro** A vector whose $k$th component is the mixing proportion for the $k$th component of the mixture model
- **mean** The mean for each component. If there is more than one component, this is a matrix whose $k$th column is the mean of the $k$th component of the mixture model.
- **variance** A list of variance parameters for the model. The components of this list depend on the model specification.

**train/test**

- **z** A matrix whose $(i,k)$th entry is the conditional probability of the $i$th observation belonging to the $k$th component of the mixture.
- **cl** A numeric vector with distinct entries representing a classification of the corresponding observations in Xtrain/Xtest.
- **rate** The number of misclassified observations.
- **Brierscore** The Brier score measuring the accuracy of the probabilities ($z$)s obtained
- **tab** A table of actual and predicted group classifications.
- **ll** The log-likelihood for the data in the mixture model.
- **bic** The Bayesian Information Criterion for the data.
References


See Also

`modelvec, noupclassify`

Examples

```r
# This function is not designed to be used on its own,
# but to be called by `code(noupclassify)
data(wine, package = "gclus")
X <- as.matrix(wine[, -1])
cl <- unclass(wine[, 1])
indtrain <- sort(sample(1:178, 120))
indtest <- setdiff(1:178, indtrain)

fitnoup <- noupclassifymodel(X[indtrain,],
cl[indtrain], X[indtest,], cl[indtest])
```

---

**plot.upclassfit**  
*Plot for upclassfit*

Description

plot method for class "upclassfit".

Usage

```r
## S3 method for class 'upclassfit'
plot(x, ...)
```

Arguments

- `x` an upclassfit object.
- `...` further graphical parameters.

Details

It produces the a posteriori probabilities of each observation belonging to each group, for the best model selected by either `upclassify` or `noupclassify`. Suspect classifications may be easier to identify as they will appear more to the centre between the upper and lower margins of the plot.
Author(s)
Niamh Russell

See Also
upclassify, noupclassify

Examples

data(wine, package = "gclus")
X <- as.matrix(wine[, -1])
c1 <- unclass(wine[, 1])
indtrain <- sort(sample(1:178, 120))
indtest <- setdiff(1:178, indtrain)

fitup <- upclassify(X[indtrain[,], c1[indtrain[,], X[indtest[,], c1[indtest[,]
plot(fitup)

print.upclassfit Printing Classification Method

Description
print method for class "upclassfit".

Usage
## S3 method for class 'upclassfit'
print(x, ...)

Arguments
x an object of class "upclassfit", the result of a call to upclassify or noupclassify.

... further arguments passed to or from other methods.

Details
print.upclassfit gives a more concise output than summary.upclassfit. Any of the members
of the list can be called using the names listed in the helptext for upclassify or noupclassify.

Value
modelName A character string identifying the model (same as the input argument).
misclass The number of misclassified observations
rate The percentage of misclassified observations
**summary.upclassfit**

**Author(s)**

Niamh Russell

**References**


**See Also**

`summary.upclassfit`, `upclassify`, `noupclassify`

**Examples**

```r
data(wine, package = "gclus")
X <- as.matrix(wine[, -1])
cl <- as.matrix(wine[, 1])
indtrain <- sort(sample(1:178, 120))
indtest <- setdiff(1:178, indtrain)

fitup <- upclassify(X[indtrain,], cl[indtrain], X[indtest,], cl[indtest])
print(fitup)
```

---

**Summary**

**Summary**

**Description**

`summary` method for class "upclassfit"

**Usage**

```r
## S3 method for class 'upclassfit'
summary(object, ...)
```

**Arguments**

- `object` an object of class "upclassfit", the result of a call to `upclassify` or `noupclassify`.
- `...` further arguments passed to or from other methods.
Details

`summary.upclassfit` gives a fuller output than `print.upclassfit`. Any of the members of the list can be called using the names listed in the help text for `upclassify` or `noupclassify`.

Value

- **Model Name**: A character string identifying the model (same as the input argument).
- **Log Likelihood**: The log-likelihood for the data in the mixture model.
- **Dimension**: The dimension of the data.
- **Ntrain**: The number of observations in the training data.
- **Ntest**: The number of observations in the test data.
- **bic**: The Bayesian Information Criterion for the best model.
- **misclass**: The number of misclassified observations (displayed only if labels are provided for the test data).
- **rate**: The percentage of misclassified observations (displayed only if labels are provided for the test data).

Author(s)

Niamh Russell

References


See Also

`upclassify`, `noupclassify`

Examples

data(wine, package = "gclus")
X <- as.matrix(wine[, -1])
cl <- as.matrix(wine[, 1])
indtrain <- sort(sample(1:178, 120))
indtest <- setdiff(1:178, indtrain)

fitup <- upclassify(X[indtrain,], cl[indtrain], X[indtest,], cl[indtest])
summary(fitup)
Description

This function performs `upclassifymodel` over a range of different models and finds the model that best fits the data by comparing the BIC values.

Usage

```r
upclassify(Xtrain, cltrain, Xtest, cltest = NULL,
modelscope = NULL, tol = 10^-5, iterlim = 1000,
Aitken = TRUE, ...)
```

Arguments

- **Xtrain**: A numeric matrix of observations where rows correspond to observations and columns correspond to variables. The group membership of each observation is known - labeled data.
- **cltrain**: A numeric vector with distinct entries representing a classification of the corresponding observations in `Xtrain`.
- **Xtest**: A numeric matrix of observations where rows correspond to observations and columns correspond to variables. The group membership of each observation may not be known - unlabeled data.
- **cltest**: A numeric vector with distinct entries representing a classification of the corresponding observations in `Xtest`. By default, these are not supplied and the function sets out to obtain these.
- **modelscope**: A character string indicating the desired models to be tested. With default NULL, all available models are tested. The models available for univariate and multivariate data are described in `modelvec`.
- **tol**: A non-negative number, with default 10^-5, which is a measure of how strictly convergence is defined.
- **iterlim**: A non-negative integer, with default 1000, which is the desired limit on the maximum number of iterations.
- **Aitken**: A logical value with default TRUE which tests for convergence using Aitken acceleration. If value is set to FALSE, convergence is tested by comparing `tol` to the change in log-likelihood between two consecutive iterations. For further information on Aitken acceleration, see `Aitken`.
- **...**: Arguments passed to or from other methods
Value

An object of class "upclassfit" providing a list of output components for each model in `modelscope`, with the Best model (according to BIC) first.

The details of the output components are as follows

call How to call the function and the order of its arguments.
Ntrain The number of observations in the training set.
Ntest The number of observations in the test set.
d The dimension of the data.
G The number of groups in the training set.
iter The number of iterations taken.
converged Whether or not the algorithm has converged. If `converged` is FALSE, then `iter` will be the maximum no of iterations.
modelName The model considered in this run of the algorithm.
parameters A list of the final model parameters estimated by the algorithm.
pro The proportion of the data to be found in each group.
mean Mean vectors for each group.
variance The variance and covariances produced by Mclust.
train A list of information about the training data. This will not have changed from before the run.
z A matrix containing the estimated probabilities that each observation in the training data belongs to each group.
c1 A vector containing the labels of the training data.
misclass The number of misclassifications of the training data.
rate The misclassification rate expressed as a percentage.
brier The Brier score expressed as a percentage.
tab The misclassification table for the training data.

test A list of information about the test data.
z A matrix containing the estimated probabilities that each observation in the training data belongs to each group.
c1 A vector containing the new labels of the training data.
misclass The number of misclassifications of the training data, provided the correct labels have been supplied.
rate The misclassification rate expressed as a percentage, provided the correct labels have been supplied.
brier The Brier score expressed as a percentage.
tab The misclassification table for the training data, provided the correct labels have been supplied.

ll The log-likelihood of the data.
bic The Bayes information criterion for the specified model.
**upclassifymodel**

**Author(s)**

Niamh Russell

**References**


**See Also**

`upclassifymodel`, `modelvec`, `Aitken`

**Examples**

```r
data(iris)
X <- as.matrix(iris[, -5])
cl <- unclass(iris[, 5])

indtrain <- sort(sample(1:150, 110))
Xtrain <- X[indtrain,]
cltrain <- cl[indtrain]

indtest <- setdiff(1:150, indtrain)
Xtest <- X[indtest,]
cltest <- cl[indtest]
modelscope <- c("EII", "VII", "VEI", "EVI")

fitupmodels <- upclassify(Xtrain, cltrain, Xtest, cltest, modelscope)
fitupmodels$Best$modelName # What is the best model?
```

**Description**

This function implements the EM algorithm by iterating over the E-step and M-step. The initial values are obtained from the labeled data then both steps are further iterated over the complete data, labeled and unlabeled data combined.
Usage

upclassifymodel(Xtrain, cltrain, Xtest, cltest = NULL,
modelName = "EEE", tol = 10^-5, iterlim = 1000,
Aitken = TRUE, ...)

Arguments

Xtrain  A numeric matrix of observations where rows correspond to observations and columns correspond to variables. The group membership of each observation is known - labeled data.

cltrain  A numeric vector with distinct entries representing a classification of the corresponding observations in Xtrain.

Xtest  A numeric matrix of observations where rows correspond to observations and columns correspond to variables. The group membership of each observation may not be known - unlabeled data.

cltest  A numeric vector with distinct entries representing a classification of the corresponding observations in Xtest. By default, these are not supplied and the function sets out to obtain them.

modelName  A character string indicating the model, with default "EEE". The models available for selection are described in modelvec

tol  A positive number, with default $10^{-5}$, which is a measure of how strictly convergence is defined.

iterlim  A positive integer, with default 1000, which is the desired limit on the maximum number of iterations.

Aitken  A logical value with default TRUE which tests for convergence using Aitken acceleration. If value is set to FALSE, convergence is tested by comparing tol to the change in log-likelihood between two consecutive iterations. For further information on Aitken acceleration, see Aitken

...  Arguments passed to or from other methods.

Details

This is an updated approach to typical classification methods. Initially, the M-step is performed on the labeled (training) data to obtain parameter estimates for the model. These are used in an E-step to obtain group memberships for the unlabeled (test) data. The training data labels and new probability estimates for test data labels are combined to form the complete data. From here, the M-step and E-step are iterated over the complete data, with continuous updating until convergence has been reached. This has been shown to result in lower misclassification rates, particularly in cases where only a small proportion of the total data is labeled.

Value

The return value is a list with the following components:

call  The function call from upclassifymodel.

Ntrain  The number of observations in the training data.
Ntest  The number of observations in the test data.
d  The dimension of the data.
G  The number of groups in the data
iter  The number of iterations required to reach convergence. If convergence was not
       obtained, this is equal to iterlim.
converged  A logical value where TRUE indicates convergence was reached and FALSE means
            iter reached iterlim without obtaining convergence.
modelName  A character string identifying the model (same as the input argument).
parameters  A vector whose $k$th component is the mixing proportion for the $k$th component of
            the mixture model. If the model includes a Poisson term for noise, there should
            be one more mixing proportion than the number of Gaussian components.
mean  The mean for each component. If there is more than one component, this is
       a matrix whose $k$th column is the mean of the $k$th component of the mixture
       model.
variance  A list of variance parameters for the model. The components of this list depend
          on the model specification.
train/test  A matrix whose $[i,k]$th entry is the conditional probability of the $i$th observation
            belonging to the $k$th component of the mixture.
c1  A numeric vector with distinct entries representing a classification of the corre-
            sponding observations in Xtrain/Xtest.
rate  The number of misclassified observations.
Brierscore  The Brier score measuring the accuracy of the probabilities ($z$s) obtained.
tab  A table of actual and predicted group classifications.
ll  The log-likelihood for the data in the mixture model.
bic  The Bayesian Information Criterion for the model.

Author(s)
Niamh Russell

References
C. Fraley and A.E. Raftery (2002). Model based clustering, discriminant analysis, and density

Model-Based Clustering, Technical Report no. 504, Department of Statistics, University of Wash-
ington.

Dean, N., Murphy, T.B. and Downey, G (2006). Using unlabelled data to update classification rules
with applications in food authenticity studies. *Journal of the royal Statistical Society: Series C* 55
(1), 1-14.

See Also
upclassify, Aitken, modelvec
Examples

# This function is not designed to be used on its own,
# but to be called by \code{upclassify}
data(wine, package = "gclus")
X <- as.matrix(wine[, -1])
cl <- unclass(wine[, 1])
indtrain <- sort(sample(1:178, 120))
indtest <- setdiff(1:178, indtrain)

fitup <- upclassifymodel(X[indtrain,], cl[indtrain], X[indtest,], cl[indtest])
Index

*Topic Classification
noupclassify, 5
noupclassifymodel, 7
plot.upclassfit, 9
print.upclassfit, 10
summary.upclassfit, 11
upclassify, 13
upclassifymodel, 15

*Topic convergence
Aitken, 3

*Topic data
modelvec, 4

Aitken, 3, 13, 15–17

graphical parameters, 9
modelvec, 4, 5, 7–9, 13, 15–17

noupclassify, 5, 9–12
noupclassifymodel, 3, 7, 7

plot(plot.upclassfit), 9
plot.upclassfit, 9
print(print.upclassfit), 10
print.upclassfit, 10, 12

summary(summary.upclassfit), 11
summary.upclassfit, 10, 11, 11

upclass(upclass-package), 2
upclass-package, 2
upclassify, 5, 7, 10–12, 13, 17
upclassifymodel, 3, 13, 15, 15