Package 'flexCWM'

March 25, 2015

Type Package

Title Flexible Cluster-Weighted Modeling

Version 1.5

Date 2015-03-25

Author Mazza A., Punzo A., Ingrassia S.

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

Description Allows for maximum likelihood fitting of cluster-weighted models, a class of mixtures of regression models with random covariates.

License GPL-2

LazyLoad yes

Depends R (>= 3.0.0)

Imports parallel,numDeriv,mnormt,mclust,ellipse,mixture,Flury,adehabitat,MASS,statmod

NeedsCompilation no

Repository CRAN

Date/Publication 2015-03-25 16:07:42

R topics documented:

flexCWM-package *flexCWM - Flexible Cluster Weighted Modeling*

Description

Allows for maximum likelihood fitting of cluster-weighted models, a class of mixtures of regression models with random covariates.

Details

Package:	CWM
Type:	Package
Version:	1.4
Date:	2015-01-31
License:	GNU-2

Author(s)

Mazza A., Punzo A., Ingrassia S.

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

References

Ingrassia, S., Minotti, S. C., and Vittadini, G. (2012). Local Statistical Modeling via the Cluster-Weighted Approach with Elliptical Distributions. *Journal of Classification*, **29**(3), 363-401.

Ingrassia, S., Minotti, S. C., and Punzo, A. (2014). Model-based clustering via linear cluster-weighted models. *Computational Statistics and Data Analysis*, **71**, 159-182.

Ingrassia, S., Punzo, A., and Vittadini, G. (2015). The Generalized Linear Mixed Cluster-Weighted Model. *Journal of Classification*, **32**(forthcoming)

Punzo, A. (2014). Flexible Mixture Modeling with the Polynomial Gaussian Cluster-Weighted Model. *Statistical Modelling*, **14**(3), 257-291.

See Also

cwm

cwm *Fit for the CWM*

Description

Maximum likelihood fitting of the cluster-weighted model by the EM algorithm.

Usage

```
cwm(formulaY = NULL, familyY = gaussian, data, Xnorm = NULL, Xbin = NULL,
  Xpois = NULL, Xmult = NULL, modelXnorm = NULL, Xbtrials = NULL, k = 1:3,
  initialization = c("random.soft", "random.hard", "kmeans", "mclust", "manual"),
  start.z = NULL, seed = NULL, maxR = 1, iter.max = 1000, threshold = 1.0e-04,
  eps = 1e-100, parallel = FALSE)
```

Arguments

formulaY	an optional object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
familyY	a description of the error distribution and link function to be used for the conditional distribution of Y in each mixture component. This can be a character string naming a family function, a family function or the result of a call to a family function. The following family functions are supported: • binomial(link = "logit") • gaussian(link = "identity") • Gamma(link = "log") • inverse.gaussian(link = "1/mu^2") • poisson(link = "log") • student.t(link = "identity") Default value is gaussian(link = "identity").
data	an optional data.frame, list, or environment with the variables needed to use formulaY.
Xnorm, Xbin, Xpois, Xmult	an optional matrix containing variables to be used for marginalization having normal, binomial, Poisson and multinomial distributions.
modelXnorm	an optional vector of character strings indicating the parsimonious models to be fitted for variables in Xnorm. The default is c("E", "V") for a single continuous covariate, and c("EII", "VII", "EEI", "VEI", "EVI", "VVI", "EEE", "VEE", "EVE", "EEV", "V for multivariate continuous covariates (see mixture:gpcm for details).
Xbtrials	an optional vector containing the number of trials for each column in Xbin. If omitted, the maximum of each column in Xbin is used.
k	an optional vector containing the numbers of mixture components to be tried. Default value is 1:3.

`initialization` an optional character string. It sets the initialization strategy for the EM-algorithm. It can be:

- `"random.soft"`
- `"random.hard"`
- `"kmeans"`
- `"mclust"`
- `"manual"`

Default value is `"random.soft"`.

`start.z` matrix of soft or hard classification: it is used only if `initialization = "manual"`.

`seed` an optional scalar. It sets the seed for the random number generator, when random initializations are used; if `NULL`, current seed is not changed. Default value is `NULL`.

`maxR` number of initializations to be tried. Default value is 1.

`iter.max` an optional scalar. It sets the maximum number of iterations in the EM-algorithm. Default value is 200.

`threshold` an optional scalar. It sets the threshold for the Aitken acceleration procedure. Default value is 1.0e-04.

`eps` an optional scalar. It sets the smallest value for eigenvalues of covariance matrices for `Xnorm`. Default value is 1e-100.

`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()`.

Details

When `familyY = binomial`, the response variable must be a matrix with two columns, where the first column is the number of "successes" and the second column is the number of "failures". When several models have been estimated, methods `summary` and `print` consider the best model according to the information criterion in `criterion`, among the estimated models having a number of components among those in `k` an error distribution among those in `familyY` and a parsimonious model among those in `modelXnorm`.

Value

This function returns a class `cwm` object, which is a list of values related to the model selected. It contains:

`call` an object of class `call`.

`formulaY` an object of class `formula` containing a symbolic description of the model fitted.

`familyY` the distribution used for the conditional distribution of Y in each mixture component.

`data` a `data.frame` with the variables needed to use `formulaY`.

`concomitant` a list containing `Xnorm`, `Xbin`, `Xpois`, `Xmult`.

`Xbtrials` number of trials used for `Xbin`.

`models` a list; each element is related to one of the models fitted. Each element is a list and contains:

- `posterior` posterior probabilities
- `iter` number of iterations performed in EM algorithm
- `k` number of (fitted) mixture components.
- `size` estimated size of the groups.
- `cluster` classification vector
- `loglik` final log-likelihood value
- `df` overall number of estimated parameters
- `prior` weights for the mixture components
- `IC` list containing values of the information criteria
- `converged` logical; `TRUE` if EM algorithm converged
- `GLModels` a list; each element is related to a mixture component and contains:
  - `model` a "`glm`" class object.
  - `sigma` estimated local scale parameters of the conditional distribution of Y, when `familyY` is `gaussian` or `student.t`
  - `t_df` estimated degrees of freedom of the t distribution, when `familyY` is `student.t`
  - `nuY` estimated shape parameter, when `familyY` is `Gamma`. The gamma distribution is parameterized according to McCullagh & Nelder (1989, p. 30)
- `concomitant` a list with estimated concomitant variables parameters for each mixture component
  - `normal.d, multinomial.d, poisson.d, binomial.d` marginal distribution of concomitant variables
  - `normal.mu` mixture component means for `Xnorm`
  - `normal.Sigma` mixture component covariance matrices for `Xnorm`
  - `normal.model` models fitted for `Xnorm`
  - `multinomial.probs` multinomial distribution probabilities for `Xmult`
  - `poisson.lambda` lambda parameters for `Xpois`
  - `binomial.p` binomial probabilities for `Xbin`

Author(s)

Mazza A., Punzo A., Ingrassia S.

References

Ingrassia, S., Minotti, S. C., and Vittadini, G. (2012). Local Statistical Modeling via the Cluster-Weighted Approach with Elliptical Distributions. *Journal of Classification*, **29**(3), 363-401.

Ingrassia, S., Minotti, S. C., and Punzo, A. (2014). Model-based clustering via linear cluster-weighted models. *Computational Statistics and Data Analysis*, **71**, 159-182.

Ingrassia, S., Punzo, A., and Vittadini, G. (2015). The Generalized Linear Mixed Cluster-Weighted Model. *Journal of Classification*, **32**(forthcoming)

McCullagh, P. and Nelder, J. (1989). Generalized Linear Models. Chapman & Hall, Boca Raton, 2nd edition

Punzo, A. (2014). Flexible Mixture Modeling with the Polynomial Gaussian Cluster-Weighted Model. *Statistical Modelling*, **14**(3), 257-291.

See Also

flexCWM-package

Examples

```
## an exemple with artificial data
data("ExCWM")
attach(ExCWM)
str(ExCWM)

# mixtures of binomial distributions
resXbin <- cwm(Xbin = Xbin, k = 1:2, initialization = "kmeans")
getParXbin(resXbin)

# Mixtures of Poisson distributions
resXpois <- cwm(Xpois = Xpois, k = 1:2, initialization = "kmeans")
getParXpois(resXpois)

# parsimonious mixtures of multivariate normal distributions
resXnorm <- cwm(Xnorm = cbind(Xnorm1,Xnorm2), k = 1:2, initialization = "kmeans")
getParXnorm(resXnorm)

## an exemple with real data
data("students")
attach(students)
str(students)
# CWM
fit2 <- cwm(WEIGHT ~ HEIGHT + HEIGHT.F , Xnorm = cbind(HEIGHT, HEIGHT.F),
  k = 2, initialization = "kmeans", modelXnorm = "EEE")
summary(fit2, concomitant = TRUE)
plot(fit2)
```

ExCWM *dataset ExCWM*

Description

An artificial data set, with 200 observations, generated by a CWM with 2 mixture components of different size, one binomial response variable, and four covariates with bivariate Gaussian, Poisson and Binomial distribution, respectively.

Usage

```
data(ExCWM)
```

Format

A dataset

See Also

flexCWM-package, cwm

Examples

```
data("ExCWM")
attach(ExCWM)
str(ExCWM)

# mixtures of binomial distributions
resXbin <- cwm(Xbin = Xbin, k = 1:2, initialization = "kmeans")
getParXbin(resXbin)

# Mixtures of Poisson distributions
resXpois <- cwm(Xpois = Xpois, k = 1:2, initialization = "kmeans")
getParXpois(resXpois)

# parsimonious mixtures of multivariate normal distributions
resXnorm <- cwm(Xnorm = cbind(Xnorm1,Xnorm2), k = 1:2, initialization = "kmeans")
getParXnorm(resXnorm)
```

Extractor functions *Extractors for* cwm *class objects.*

Description

These functions extract values from cwm class objects.

Usage

```
getBestModel(object, criterion = "BIC", k = NULL, modelXnorm = NULL, familyY = NULL)
getPosterior(object, ...)
getSize(object, ...)
getCluster(object, ...)
getParGLM(object, ...)
getParConcomitant(object, name = NULL, ...)
getPar(object, ...)
```

```
getParPrior(object, ...)
getParXnorm(object, ...)
getParXbin(object, ...)
getParXpois(object, ...)
getParXmult(object, ...)
getIC(object,criteria)
whichBest(object, criteria = NULL, k = NULL, modelXnorm = NULL, familyY = NULL)

## S3 method for class 'cwm'
summary(object, criterion = "BIC",  concomitant = FALSE,
  digits = getOption("digits")-2, ...)
## S3 method for class 'cwm'
print(x, ...)
```

Arguments

object, x	a class cwm object.
criterion	a string with the information criterion to consider; supported values are:"AIC", "AICc", "AICu", "AIC3 Default value is "BIC".
criteria	a vector of strings with the names of information criteria to consider. If NULL all the supported infromation criteria are considered.
k	an optional vector containing the numbers of mixture components to consider. If not specified, all the estimated models are considered.
modelXnorm	an optional vector of character strings indicating the parsimonious models to consider for Xnorm. If not specified, all the estimated models are considered.
familyY	an optional vector of character strings indicating the conditional distribution of Y in each mixture component to consider. If not specified, all the estimated models are considered.
name	an optional vector of strings specifing the names of distribution families of concomitant variables; if NULL, parameters estimated for all concomitant variables are returned.
concomitant	When TRUE, concomitant variables parameters are displayed. Default is FALSE.
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 k an error distribution among those in familyY and a parsimonious model among those in modelXnorm. getIC provides values for the information criteria in criteria.

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

Examples

```
#res <- cwm(Y=Y,Xcont=X,k=1:4,seed=1)
#summary(res)
#plot(res)
```

---

plot.cwm *Plot for CWMs*

Description

Plot method for cwm class objects.

Usage

```
 ## S3 method for class 'cwm'
plot(x, regr = TRUE, ctype = c("Xnorm","Xbin","Xpois",
                  "Xmult"), which = NULL, criterion = "BIC", k = NULL,
                  modelXnorm = NULL, familyY = NULL, ...)
```

Arguments

x	An object of class cwm.
regr	boolean, allows for bivariate regression plot.
ctype	a vector with concomitant variables types to plot.
which	a vector with columns number to plot, or "all" for all the columns
criterion	a string with the information criterion to consider; supported values are:"AIC", "AICc", "AICu", "AIC3 Default value is "BIC".
k	an optional vector containing the numbers of mixture components to consider. If not specified, all the estimated models are considered.
modelXnorm	an optional vector of character strings indicating the parsimonious models to consider for Xnorm. If not specified, all the estimated models are considered.
familyY	an optional vector of character strings indicating the conditional distribution of Y in each mixture component to consider. If not specified, all the estimated models are considered.
...	further arguments for plot.

Examples

```
data("students")
attach(students)
str(students)
fit2 <- cwm(WEIGHT ~ HEIGHT + HEIGHT.F , Xnorm = cbind(HEIGHT, HEIGHT.F), k = 2,
  initialization = "kmeans", modelXnorm = "EEE")
summary(fit2, concomitant = TRUE)
plot(fit2)
```

students *dataset students*

Description

A dataframe with data from a survey of 270 students attending a statistics course at the Department of Economics and Business of the University of Catania in the academic year 2011/2012. It contains the following variables:

- GENDER gender of the respondent;
- HEIGHT height of the respondent, measured in centimeters;
- WEIGHT weight of the respondent, measured in kilograms;
- HEIGHT.F height of respondent's father, measured in centimeters.

Usage

```
data(students)
```

Format

A dataset

Source

http://www.economia.unict.it/punzo/

References

Ingrassia, S., Minotti, S. C., and Punzo, A. (2014). Model-based clustering via linear cluster-weighted models. *Computational Statistics and Data Analysis*, **71**, 159-182.

See Also

flexCWM-package, cwm

Examples

```
data("students")
attach(students)
str(students)
fit2 <- cwm(WEIGHT ~ HEIGHT + HEIGHT.F , Xnorm = cbind(HEIGHT, HEIGHT.F), k = 2,
  initialization = "kmeans", modelXnorm = "EEE")
summary(fit2, concomitant = TRUE)
plot(fit2)
```

Index