Package 'inarmix'

February 20, 2015

Type Package

Title Mixture models for longitudinal count data.

Version 0.4

Date 2014-02-20

Author Nicholas Henderson and Paul Rathouz

Maintainer Nicholas Henderson <nhenders@stat.wisc.edu>

Description Fits mixtures models for longitudinal data. Appropriate when the data are counts and when the correlation structure is assumed to be AR(1).

License GPL-2

Depends Matrix

Imports Rcpp (>= 0.10.3)

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-03-05 18:21:57

R topics documented:

diagnose *Diagnostics for the model fit.*

Description

Provides diagnostics of the mixture model fit.

Usage

```
diagnose(results)
```

Arguments

results	An object of class "inarmix"

Value

An object of class "diagnose.inarmix" which is a list containing the following components

em.converged	An indication of whether or not the "EM algorithm" converged.
niter	The number of iterations for required for the EM algorithm to converge.
nclasses	The number of latent classes used in the model fit.
loglikhood	The values of the log-likelihood function at each EM iteration.
ConvMat	This is a matrix containing several pieces of information. The columns with GEE in the title indicate if the algorithms for updating the parameters within an EM iteration have converged. The column with l[i] just contains the log-likelihood value. The column with the title "||Psi^2||" gives the norm of the "global estimating equation" - this should be near zero if the algorithm converged.

Author(s)

Nicholas Henderson

Examples

```
XX <- cbind(rep(1,9),c(0:8)/4)
colnames(XX) <- c("const","time")
coefs <- rbind(c(-.2,0),c(1.2,.3))
alpha <- c(.2,.2)
scale <- c(2,2)
mix.prop <- c(.8,.2)

testdat <- GenerateMixData(200,coefs,alpha,scale,mix.prop,XX)
testfit <- inarmix(y~time,nclasses=2,id=subject,data=testdat,maxiter=3)

diagnose(testfit)
```

GenerateMixData *Generates simulated data according to a specified INAR mixture model.*

Description

This function generates data according to a user-specified INAR mixture model. It returns a data frame which can be used immediately by inarmix.

Usage

```
GenerateMixData(m, coefs, autocorr, scale=NULL, mix, design.mat,
                return.labels=F,poisson=F)
```

Arguments

m	a numeric scalar (the number of subjects)
coefs	a numeric vector
autocorr	a numeric scalar
scale	a numerica scalar
mix	a numeric vector
design.mat	a numeric matrix (the design matrix)
return.labels	an indicator of whether or not to return the class labels
poisson	an indicator of whether or not the data should have marginal Poisson distributions.

Value

A data frame which contains the response and a subject identifier. The other columns contain the data entered from the design.mat argument.

Author(s)

Nicholas Henderson

Examples

```
XX <- cbind(rep(1,9),c(0:8)/4)
colnames(XX) <- c("const","time")
coefs <- rbind(c(-.2,0),c(1.2,.3))
autocorr <- c(.2,.2)
scale <- c(1.5,1.5)
mix.prop <- c(.8,.2)

testdat <- GenerateMixData(1000,coefs,autocorr,scale,mix.prop,XX)
```

`inarmix` *Finite mixture model for longitudinal count data.*

Description

Estimates model parameters of a finite mixture model. Appropriate for longitudinal data when the responses are counts and when the correlation structure is assumed to be AR(1).

Usage

```
inarmix(formula,nclasses=1,id,data,initvals=NULL,maxiter=200,stoptol=1e-5,
        num.restarts=1,time=NULL,dropthresh=.01)
```

Arguments

`formula`	a formula expression used to specify the regression model
`nclasses`	The number of components in the finite mixture model.
`id`	the name of the variable which identifies the individual subjects.
`data`	a data frame
`initvals`	initial estimates of the parameters (optional). This should be a list of the form list(coef=,autocorr=,scale=,mix.prop=)
`maxiter`	The maximum number of EM iterations to be performed (optional).
`stoptol`	tolerance level which determines convergence. The default is 1e-7.
`num.restarts`	The number of runs. Each run has a random starting value for the parameters.
`time`	the name of the variable which indicates time. When left blank, the data are assumed to have the correct time-ordering.
`dropthresh`	The threshold at which one class is dropped from the model. If the estimated proportion for a class drops below this level, it is removed from the fitting procedure.

Value

An object of class `"inarmix"` which is a list containing at least the following components

`coefficients`	A matrix of estimated regression coefficients. Each row contains the coefficients for one class.
`mix.prop`	The estimated class-membership probabilities.
`post.probs`	A nclasses x num.subjects matrix. The posterior probabilities of class-membership for each subject and class.
`loglikhood`	The final value of the log-likelihood.
`niter`	The number of iterations required for convergence.
`cov.mat`	The variance covariance matrix of the parameter estimates.
`call`	the matched call
`nclasses`	The number of classes in the final model fit.

Author(s)

Nicholas Henderson

Examples

```
set.seed(4297)

#############################################################
#### Simulate data from a two class model

XX <- cbind(rep(1,9),c(0:8)/4)
colnames(XX) <- c("const","time")
beta <- rbind(c(-.2,0),c(1.2,.3))
### this means that for group 1: (beta_{0},beta_{1}) = (-.2,0)
### and for group 2: (beta_{0},beta_{1}) = (1.2,.3)
autocorr <- c(.2,.2)
scale <- c(2,2)
mix.prop <- c(.8,.2)  ## proportion in group 1 is .8

testdat <- GenerateMixData(500,beta,autocorr,scale,mix.prop,XX)
testdat[1:5,]


#########################################################################
####  Fit a linear curve with two classes (with a maximum of 4 iterations)

twoclassfit <- inarmix(y~time,nclasses=2,id=subject,data=testdat,maxiter=4)
summary(twoclassfit)

diagnose(twoclassfit)

#############################################################
#####  Fit the same model with specified starting values.

inpars <- list()
inpars$coef <- rbind(c(-.5,.1),c(.5,0))
inpars$autocorr <- rep(.3,2)
inpars$scale <- rep(2,2)
inpars$mix.prop <- c(.6,.4)

twoclassfit2 <- inarmix(y~time,nclasses=2,id=subject,data=testdat,initvals=inpars,
                        maxiter=4)
summary(twoclassfit2)


#############################################################
### Try fitting a one class model with the same data
oneclassfit <- inarmix(y~time,nclasses=1,id=subject,data=testdat)
summary(oneclassfit)


#########################################################################
```

```
####  Fit a two class model with multiple starts
## Not run:
testfit_multi <- inarmix(y~time,nclasses=2,id=subject,data=testdat,num.restarts=3)
summary(testfit_multi)

#### Look at final log-likelihood values for each restart
testfit_multi$reploglik

#### Look at final parameter estimates for each restart
testfit_multi$finalvals

################################################################################
###########  Simulate data from a four class model

XX <- cbind(rep(1,9),seq(0,2,by=.25))
colnames(XX) <- c("const","time")
beta <- rbind(c(-.4,-.1),c(1.4,-.6),c(0,.7),c(1.4,0))
autocorr <- rep(.2,4)
scale <- rep(1.5,4)
mix.prop <- c(.5,.25,.15,.1)

testdat4 <- GenerateMixData(1000,beta,autocorr,scale,mix.prop,XX)

### Fit a four class model

testfit_four <- inarmix(y~time,nclasses=4,id=subject,data=testdat4,maxiter=5)
summary(testfit_four)

## End(Not run)
```

Index