

Package ‘gamlss.mx’

November 22, 2009

Type Package

Title A GAMLSS add on package for fitting mixture distributions

Version 3.1-0

Date 2009-11-22

Author Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby
<r.rigby@londonmet.ac.uk>

Maintainer Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

LazyLoad yes

Depends R (>= 2.2.1), gamlss, nnet, MASS

Description The main purpose of this package is to allow fitting of mixture distributions with GAMLSS models.

License GPL-2 | GPL-3

URL <http://www.gamlss.com/>

Repository CRAN

Date/Publication 2009-11-22 16:17:24

R topics documented:

gamlss.mx-package	2
dMX	3
enzyme	4
gamlssMX	5
gamlssNP	6
MX.control	10
NP.control	11

Index	13
--------------	-----------

gamlss.mx-package *The GAMLSS add on package for mixture distributions*

Description

The main purpose of this package is to allow the user of the GAMLSS models to fit mixture distributions.

Details

Package: gamlss.mx
Type: Package
Version: 0.0
Date: 2005-08-3
License: GPL (version 2 or later)

This package has two main function the `gamlssMX()` which is loosely based on the package `flexmix` of R and the function `gamlssNP()` which is based on the `npmlreg` package of Jochen Einbeck, Ross Darnell and John Hinde (2006) which in turns is based on several GLIM4 macros originally written by Murray Aitkin and Brian Francis. It also contains the function `ggz()` which is written by Nick Sofroniou and the function `gauss.quad()` written by Gordon Smyth.

Author(s)

Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>> and Bob Rigby <<r.rigby@londonmet.ac.uk>>
Maintainer: Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Jochen Einbeck, Ross Darnell and John Hinde (2006) `npmlreg`: Nonparametric maximum likelihood estimation for random effect models, R package version 0.34
Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.
Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also <http://www.gamlss.com/>).

See Also

[gamlss,gamlss.family](#)

Examples

```
data(enzyme)
mmNO<-gamlssMXfits(n=10,enzyme$act~1, family=NO, K=2)
```

```

mmNO
fyNO<-dMX(y=seq(0,3,.01), mu=list(1.253, 0.1876), sigma=list(exp(-0.6665 ), exp(-2.573 )),
          pi=list(0.4079609, 0.5920391 ), family=list("NO","NO") )
hist(enzyme$act, freq=FALSE, ylim=c(0,3.5), xlim=c(0,3), br=21)
lines(seq(0,3,.01), fyNO, col="red")
# equivalent model using gamlssNP
mmNP <- gamlssNP(act~1, data=enzyme, random=~1, sigma.fo=~MASS, family=NO, K=2)

```

dMX

Evaluate the d (pdf) and p (cdf) functions from GAMLSS mixtures

Description

The functions `dMX` and `pMX` can be used to evaluate the pdf (p function) and the cdf (p function) respectively from a `gamlss.family` mixture.

Usage

```

dMX(y, mu = list(mu1 = 1, mu2 = 5), sigma = list(sigma1 = 1, sigma2 = 1),
     nu = list(nu1 = 1, nu2 = 1), tau = list(tau1 = 1, tau2 = 1),
     pi = list(pi1 = 0.2, pi2 = 0.8), family = list(fam1 = "NO", fam2 = "NO"),
     log = FALSE, ...)
pMX(q, mu = list(mu1 = 1, mu2 = 5), sigma = list(sigma1 = 1, sigma2 = 1),
     nu = list(nu1 = 1, nu2 = 1), tau = list(tau1 = 1, tau2 = 1),
     pi = list(pi1 = 0.2, pi2 = 0.8), family = list(fam1 = "NO", fam2 = "NO"),
     log = FALSE, ...)

```

Arguments

<code>y, q</code>	vector of quantiles
<code>mu</code>	a vector of mu's
<code>sigma</code>	a vector of sigma's
<code>nu</code>	a vector of nu's
<code>tau</code>	a vector of tau's
<code>pi</code>	a vector of pi's
<code>family</code>	a vector of GAMLSS family's
<code>log</code>	whether the log of the function or not
<code>...</code>	for extra arguments

Value

Returns values or pdf or cdf.

Author(s)

Mikis Stasinopoulos

Examples

```
fyNO<-dMX(y=seq(0,3,.01), mu=list(1.253, 0.1876), sigma=list(exp(-0.6665 ), exp(-2.573 )),
          pi=list(0.4079609, 0.5920391 ), family=list("NO","NO" )
plot(fyNO~seq(0,3,.01), type="l")
FyNO<-pMX(q=seq(0,3,.01), mu=list(1.253, 0.1876), sigma=list(exp(-0.6665 ), exp(-2.573 )),
          pi=list(0.4079609, 0.5920391 ), family=list("NO","NO" )
plot(FyNO~seq(0,3,.01), type="l")
```

 enzyme

Data used in gamlss.mx

Description

enzyme : The data comprise independent measurement of enzyme activity in the blood of 245 individuals. The data were analysed by Bechker *at al.* (1993).

brains : the brain size, `brain`, and body weight, `body`, for 28 differnt animals.

Usage

```
data(enzyme)
data(brains)
```

Format

enzyme : data frame with 245 observations on the following variable `act`.

brains : data frame with 28 observations on the following variables. `body`, `brain`

`act` a numeric vector showing enzyme activity in the blood of 245 individuals.

`body` a numeric vector showing the body weight of 28 differnt animals

`brain` a numeric vector showing the brain size of 28 differnt animals

Examples

```
data(enzyme)
hist(enzyme$act)
data(brains)
brains$lbrain<-log(brains$brain)
brains$lbody<-log(brains$body)
with(brains, plot(lbrain~lbody))
```

gamlssMX

*Function to fit finite mixture of gamlss family distributions***Description**

The function `gamlssMX` is design for fitting a K fold non parametric mixture of gamlss family distributions.

Usage

```
gamlssMX(formula = formula(data), pi.formula = ~1,
          family = "NO", weights, K = 2, prob = NULL,
          data = sys.parent(), control = MX.control(),
          g.control = gamlss.control(trace = FALSE),
          zero.component = FALSE, ...)
gamlssMXfits(n = 5, formula = formula(data), pi.formula = ~1,
             family = "NO", weights, K = 2, prob = NULL,
             data = sys.parent(), control = MX.control(),
             g.control = gamlss.control(trace = FALSE),
             zero.component = FALSE, ... )
```

Arguments

<code>formula</code>	This argument it should be a formula (or a list of formulae of length K) for modelling the μ parameter of the model. Note that modelling the rest of the distributional parameters it can be done by using the usual <code>...</code> which passes the arguments to <code>gamlss()</code>
<code>pi.formula</code>	This should be a formula for modelling the prior probabilities as a function of explanatory variables. Note that no smoothing of other additive terms are allowed here only the usual linear terms. The modelling here is done using the <code>multinom()</code> function from package <code>nnet</code>
<code>family</code>	This should be a <code>gamlss.family</code> distribution (or a list of distributions). Note that if different distributions are used here their parameters should be comparable for ease of interpretation.
<code>weights</code>	prior weights if needed
<code>K</code>	the number of finite mixtures with default <code>K=2</code>
<code>prob</code>	prior probabilities if required for starting values
<code>data</code>	the data frame nedded for the fit. Note that this is compulsory if <code>pi.formula</code> is used.
<code>control</code>	This argument sets the control parameters for the EM iterations algorithm. The default setting are given in the <code>MX.control</code> function
<code>g.control</code>	This argument can be used to pass to <code>gamlss()</code> control parameters, as in <code>gamlss.control</code>
<code>n</code>	the number of fits required in <code>gamlssMXfits()</code>

```

zero.component
                whether zero component models exist, default is FALSE
...
                for extra arguments

```

Author(s)

Mikis Stasinopoulos and Bob Rigby

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape, (with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also <http://www.gamlss.com/>).

See Also

[gamlss](#), [gamlss.family](#)

Examples

```

library(MASS)
data(geyser)
# fitting 2 finite normal mixtures
m1<-gamlssMX(waiting~1,data=geyser,family=NO, K=2)
#fitting 2 finite gamma mixtures
m2<-gamlssMX(waiting~1,data=geyser,family=GA, K=2)
# fitting a model for pi
# first create a data frame
geyser1<-matrix(0,ncol=2, nrow=298)
geyser1[,1] <-geyser$waiting[-1]
geyser1[,2] <-geyser$duration[-299]
colnames(geyser1)<- c("waiting", "duration")
geyser1 <-data.frame(geyser1)
# get the best of 5 fits
m3<-gamlssMXfits(n=5, waiting~1, pi.formula=~duration, data=geyser1,family=NO, K=2)
m3

```

gamlssNP

A function to fit finite mixtures using the gamlss family of distributions

Description

This function will fit a finite (or normal) mixture distribution where the kernel distribution can belong to any `gamlss` family of distributions using the EM algorithm. The function is based on functions `alldist()` and `allvc` of the `npmlreg` package of Jochen Einbeck, John Hinde and Ross Darnell.

Usage

```
gamlssNP(formula, random = ~1, family = NO(), data = NULL, K = 4,
          mixture = c("np", "gq"),
          tol = 0.5, weights, pluginz, control = NP.control(...),
          g.control = gamlss.control(trace = FALSE), ...)
```

Arguments

formula	a formula defining the response and the fixed effects for the mu parameters
random	a formula defining the random part of the model
family	a gamlss family object
data	the data frame which for this function is mandatory even if the data are attached
K	the number of mass points/integration points (supported values are 1:10,20)
mixture	the mixing distribution, "np" for non-parametric or "gq" for Gaussian Quadrature
tol	the tolerance scalar usually between zero and one
weights	prior weights
pluginz	optional
control	this sets the control parameters for the EM iterations algorithm. The default setting is the NP.control function
g.control	the gamlss control function, gamlss.control, passed to the gamlss fit
...	for extra arguments

Details

The function `gamlssNP()` is a modification of the R functions `alldist()` and `allvc` created by Jochen Einbeck and John Hinde. Both functions were originally created by Ross Darnell (2002). Here the two functions are merged to one `gamlssNP` and allows finite mixture from gamlss family of distributions.

The following are comments from the original Einbeck and Hinde documentation.

"The nonparametric maximum likelihood (NPML) approach was introduced in Aitkin (1996) as a tool to fit overdispersed generalized linear models. Aitkin (1999) extended this method to generalized linear models with shared random effects arising through variance component or repeated measures structure. Applications are two-stage sample designs, when firstly the primary sampling units (the upper-level units, e.g. classes) and then the secondary sampling units (lower-level units, e.g. students) are selected, or longitudinal data. Models of this type have also been referred to as multi-level models (Goldstein, 2003). This R function is restricted to 2-level models. The idea of NPML is to approximate the unknown and unspecified distribution of the random effect by a discrete mixture of k exponential family densities, leading to a simple expression of the marginal likelihood, which can then be maximized using a standard EM algorithm. When option 'gq' is set, then Gauss-Hermite masses and mass points are used and considered as fixed, otherwise they serve as starting points for the EM algorithm. The position of the starting points can be concentrated or extended by setting `tol` smaller or larger than one, respectively. Variance component models with

random coefficients (Aitkin, Hinde & Francis, 2005, p. 491) are also possible, in this case the option `random.distribution` is restricted to the setting `'np'`. The weights have to be understood as frequency weights, i.e. setting all weights equal to 2 will duplicate each data point and hence double the disparity and deviance. Warning: There might be some options and circumstances which had not been tested and where the weights do not work." Note that in keeping with the `gamlss` notation disparity is called global deviance.

Value

The function `gamlssNP` produces an object of class `"gamlssNP"`. This object contains several components.

<code>family</code>	the name of the <code>gamlss</code> family
<code>type</code>	the type of distribution which in this case is "Mixture"
<code>parameters</code>	the parameters for the kernel <code>gamlss</code> family distribution
<code>call</code>	the call of the <code>gamlssNP</code> function
<code>y</code>	the response variable
<code>bd</code>	the binomial denominator, only for BI and BB models
<code>control</code>	the NP.control settings
<code>weights</code>	the vector of weights of the expanded fit
<code>G.deviance</code>	the global deviance
<code>N</code>	the number of observations in the fit
<code>rqres</code>	a function to calculate the normalized (randomized) quantile residuals of the object (here is the <code>gamlss</code> object rather than <code>gamlssNP</code> and it should change??)
<code>iter</code>	the number of external iterations in the last <code>gamlss</code> fitting (?? do we need this?)
<code>type</code>	the type of the distribution or the response variable here set to "Mixture"
<code>method</code>	which algorithm is used for the <code>gamlss</code> fit, <code>RS()</code> , <code>CG()</code> or <code>mixed()</code>
<code>contrasts</code>	the type of contrasts used in the fit
<code>converged</code>	whether the <code>gamlss</code> fit has converged
<code>residuals</code>	the normalized (randomized) quantile residuals of the model
<code>mu.fv</code>	the fitted values of the extended μ model, also <code>sigma.fv</code> , <code>nu.fv</code> , <code>tau.fv</code> for the other parameters if present
<code>mu.lp</code>	the linear predictor of the extended μ model, also <code>sigma.lp</code> , <code>nu.lp</code> , <code>tau.lp</code> for the other parameters if present
<code>mu.wv</code>	the working variable of the extended μ model, also <code>sigma.wv</code> , <code>nu.wv</code> , <code>tau.wv</code> for the other parameters if present
<code>mu.wt</code>	the working weights of the μ model, also <code>sigma.wt</code> , <code>nu.wt</code> , <code>tau.wt</code> for the other parameters if present
<code>mu.link</code>	the link function for the μ model, also <code>sigma.link</code> , <code>nu.link</code> , <code>tau.link</code> for the other parameters if present
<code>mu.terms</code>	the terms for the μ model, also <code>sigma.terms</code> , <code>nu.terms</code> , <code>tau.terms</code> for the other parameters if present

<code>mu.x</code>	the design matrix for the mu, also <code>sigma.x</code> , <code>nu.x</code> , <code>tau.x</code> for the other parameters if present
<code>mu.qr</code>	the QR decomposition of the mu model, also <code>sigma.qr</code> , <code>nu.qr</code> , <code>tau.qr</code> for the other parameters if present
<code>mu.coefficients</code>	the linear coefficients of the mu model, also <code>sigma.coefficients</code> , <code>nu.coefficients</code> , <code>tau.coefficients</code> for the other parameters if present
<code>mu.formula</code>	the formula for the mu model, also <code>sigma.formula</code> , <code>nu.formula</code> , <code>tau.formula</code> for the other parameters if present
<code>mu.df</code>	the mu degrees of freedom also <code>sigma.df</code> , <code>nu.df</code> , <code>tau.df</code> for the other parameters if present
<code>mu.nl.df</code>	the non linear degrees of freedom, also <code>sigma.nl.df</code> , <code>nu.nl.df</code> , <code>tau.nl.df</code> for the other parameters if present
<code>df.fit</code>	the total degrees of freedom use by the model
<code>df.residual</code>	the residual degrees of freedom left after the model is fitted
<code>data</code>	the original data set
<code>EMiter</code>	the number of EM iterations
<code>EMconverged</code>	whether the EM has converged
<code>allresiduals</code>	the residuas for the long fit
<code>mass.points</code>	the estimates mass point (if "np" mixture is used)
<code>K</code>	the number of mass points used
<code>post.prob</code>	contains a matrix of posteriori probabilities,
<code>prob</code>	the estimated mixture probalilities
<code>aic</code>	the Akaike information criterion
<code>sbc</code>	the Bayesian information criterion
<code>formula</code>	the formula used in the expanded fit
<code>random</code>	the random effect formula
<code>pweights</code>	prior weights
<code>ebp</code>	the Empirical Bayes Predictions (Aitkin, 1996b) on the scale of the linear predictor

Note that in case of Gaussian quadrature, the coefficient given at 'z' in coefficients corresponds to the standard deviation of the mixing distribution.

As a by-product, `gamlssNP` produces a plot showing the global deviance against the iteration number. Further, a plot with the EM trajectories is given. The x-axis corresponds to the iteration number, and the y-axis to the value of the mass points at a particular iteration. This plot is not produced when mixture is set to "gq"

Author(s)

Mikis Stasinopoulos based on function created by Jochen Einbeck John Hinde and Ross Darnell

References

- Aitkin, M. and Francis, B. (1995). Fitting overdispersed generalized linear models by nonparametric maximum likelihood. *GLIM Newsletter* 25 , 37-45.
- Aitkin, M. (1996a). A general maximum likelihood analysis of overdispersion in generalized linear models. *Statistics and Computing* 6 , 251-262.
- Aitkin, M. (1996b). Empirical Bayes shrinkage using posterior random effect means from non-parametric maximum likelihood estimation in general random effect models. *Statistical Modelling: Proceedings of the 11th IWSM 1996* , 87-94.
- Aitkin, M., Francis, B. and Hinde, J. (2005) *Statistical Modelling in GLIM 4*. Second Edition, Oxford Statistical Science Series, Oxford, UK.
- Einbeck, J. & Hinde, J. (2005). A note on NPML estimation for exponential family regression models with unspecified dispersion parameter. Technical Report IRL-GLWY-2005-04, National University of Ireland, Galway.
- Einbeck, J. Darnell R. and Hinde J. (2006) npmlreg: Nonparametric maximum likelihood estimation for random effect models, R package version 0.34
- Hinde, J. (1982). Compound Poisson regression models. *Lecture Notes in Statistics* 14 ,109-121.
- Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape, (with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.
- Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also <http://www.gamlss.com/>).

See Also

[gamlss](#), [gamlss.family](#)

Examples

```
data(enzyme)
# equivalent model using gamlssNP
mmNP1 <- gamlssNP(act~1, data=enzyme, random=~1, family=NO, K=2)
mmNP2 <- gamlssNP(act~1, data=enzyme, random=~1, sigma.fo=~MASS, family=NO, K=2)
AIC(mmNP1, mmNP2)
```

MX.control

The control function for gamlssMX

Description

The function sets controls for the `gamlssMX` function.

Usage

```
MX.control(cc = 1e-04, n.cyc = 200, trace = FALSE,
           seed = NULL, plot = TRUE, sample = NULL, ...)
```

Arguments

<code>cc</code>	convergent criterion for the EM
<code>n.cyc</code>	number of cycles for EM
<code>trace</code>	whether to print the EM iterations
<code>seed</code>	a number for setting the seeds for starting values
<code>plot</code>	whether to plot the sequence of global deviance up to convergence
<code>sample</code>	how large the sample to be in the starting values
<code>...</code>	for extra arguments

Value

Returns a list

Author(s)

Mikis Stasinopoulos and Bob Rigby

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape, (with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also <http://www.gamlss.com/>).

See Also

[gamlss](#), [gamlssMX](#), [gamlssMXfits](#)

NP.control

Control function for gamlssNP

Description

This is a control function for `gamlssNP` function.

Usage

```
NP.control(EMcc = 0.001, EMn.cyc = 200, damp = TRUE,  
           trace = TRUE, plot.opt = 3, ...)
```

Arguments

<code>EMcc</code>	convergence criterion for the EM
<code>EMn.cyc</code>	number of cycles for the EM
<code>damp</code>	Not in used
<code>trace</code>	whether to print the EM iterations
<code>plot.opt</code>	plotting the
<code>...</code>	for extra arguments

Value

Returns a list.

Author(s)

Mikis Stasinopoulos

References

Einbeck, J. Darnell R. and Hinde J. (2006) npmlreg: Nonparametric maximum likelihood estimation for random effect models, R package version 0.34

See Also

[gamlss](#), [gamlssNP](#)

Index

*Topic **datasets**

enzyme, [4](#)

*Topic **distribution**

dMX, [3](#)

gamlss.mx-package, [1](#)

*Topic **package**

gamlss.mx-package, [1](#)

*Topic **regression**

dMX, [3](#)

gamlss.mx-package, [1](#)

gamlssMX, [4](#)

gamlssNP, [6](#)

MX.control, [10](#)

NP.control, [11](#)

brains (*enzyme*), [4](#)

dMX, [3](#)

enzyme, [4](#)

gamlss, [2](#), [6](#), [10–12](#)

gamlss.family, [2](#), [6](#), [10](#)

gamlss.mx (*gamlss.mx-package*), [1](#)

gamlss.mx-package, [1](#)

gamlssMX, [4](#), [11](#)

gamlssMXfits, [11](#)

gamlssMXfits (*gamlssMX*), [4](#)

gamlssNP, [6](#), [12](#)

MX.control, [10](#)

NP.control, [11](#)

pMX (*dMX*), [3](#)