Package 'npmlreg'

July 22, 2025

Type Package

Title Nonparametric Maximum Likelihood Estimation for Random Effect
Models

Version 0.46-5

Date 2018-08-31

Author Jochen Einbeck, Ross Darnell and John Hinde

Maintainer Jochen Einbeck <jochen.einbeck@durham.ac.uk>

Depends R (>= 2.3.1)

Imports statmod (>= 1.4.20)

Suggests MASS, nlme, lattice

LazyLoad yes

Description Nonparametric maximum likelihood estimation or Gaussian
quadrature for overdispersed generalized linear models and
variance component models.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2018-08-31 13:20:03 UTC

Contents

---

npmlreg-package	*Nonparametric maximum likelihood estimation for random effect models*

Description

Nonparametric maximum likelihood estimation or Gaussian quadrature for overdispersed generalized linear models and variance component models. The main functions are alldist and allvc.

Details

Package:	npmlreg
Type:	Package
License:	GPL version 2 or newer

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

For details on the GNU General Public License see http://www.gnu.org/copyleft/gpl.html or write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Acknowledgments

Author(s)

Jochen Einbeck, Ross Darnell and John Hinde.

Maintainer: Jochen Einbeck <jochen.einbeck@durham.ac.uk>

References

Aitkin, M., Francis, B. and Hinde, J. (2009). Statistical Modelling in R. Oxford Statistical Science Series, Oxford, UK.

Einbeck, J., and Hinde, J.: Nonparametric maximum likelihood estimation for random effect models in R. Vignette to R package **npmlreg**. Type `vignette("npmlreg-v")` to open it.

See Also

[glm](#)

---

alldist	*NPML estimation or Gaussian quadrature for overdispersed GLM's and variance component models*

Description

Fits a random effect model using Gaussian quadrature (Hinde, 1982) or nonparametric maximum likelihood (Aitkin, 1996a). The function `alldist` is designed to account for overdispersion, while `allvc` fits variance component models.

Usage

```
alldist(formula,
        random = ~1,
        family = gaussian(),
        data,
        k = 4,
        random.distribution = "np",
        tol = 0.5,
        offset,
        weights,
        pluginz,
        na.action,
        EMmaxit = 500,
        EMdev.change = 0.001,
        lambda = 0,
        damp = TRUE,
        damp.power = 1,
        spike.protect = 0,
        sdev,
        shape,
        plot.opt = 3,
        verbose = TRUE,
        ...)

allvc(formula,
```

```
        random = ~1,
        family = gaussian(),
        data,
        k = 4,
        random.distribution = "np",
        tol = 0.5,
        offset,
        weights,
        pluginz,
        na.action,
        EMmaxit = 500,
        EMdev.change = 0.001,
        lambda=0,
        damp = TRUE,
        damp.power = 1,
        spike.protect=0,
        sdev,
        shape,
        plot.opt = 3,
        verbose = TRUE,
        ...)
```

Arguments

formula	a formula defining the response and the fixed effects (e.g. y ~ x).
random	a formula defining the random model. In the case of alldist, set random = ~1 to model overdispersion, and for instance random = ~x to introcude a random coefficient x. In the case of allvc, set random=~1|PSU to model overdispersion on the upper level, where PSU is a **factor** for the primary sampling units, e.g. groups, clusters, classes, or individuals in longitudinal data, and define random coefficients accordingly.
family	conditional distribution of responses: "gaussian", "poisson", "binomial", "Gamma", or "inverse.gaussian" can be set. If "gaussian", "Gamma", or "inverse.gaussian", then equal component dispersion parameters are assumed, except if the optional parameter lambda is modified. The same link functions as for function [glm](#) are supported.
data	the data frame (mandatory, even if it is attached to the workspace!).
k	the number of mass points/integration points (supported are up to 600 mass points).
random.distribution	
	the mixing distribution, Gaussian Quadrature (gq) or NPML (np) can be set.
tol	the tol scalar (usually, $0 < \text{tol} \le 1$)
offset	an optional offset to be included in the model.
weights	optional prior weights for the data.
pluginz	optional numerical vector of length k specifying the starting mass points of the EM algorithm.

na.action	a function indicating what should happen when NA's occur, with possible arguments na.omit and na.fail. The default is set by the na.action setting in options().
EMmaxit	maximum number of EM iterations.
EMdev.change	stops EM algorithm when deviance change falls below this value.
lambda	only applicable for Gaussian, Gamma, and Inverse Gaussian mixtures. If set, standard deviations/ shape parameters are calculated smoothly across components via an Aitchison-Aitken kernel ([dkern](#)) with parameter lambda. The setting lambda= 0 is automatically mapped to lambda =1/k and corresponds to the case 'maximal smoothing' (i.e. equal component dispersion parameters), while lambda=1 means 'no smoothing' (unequal disp. param.)
damp	switches EM damping on or off.
damp.power	steers degree of damping applied on dispersion parameter according to formula 1-(1-tol)^(damp.power*iter+1), see Einbeck & Hinde (2006).
spike.protect	for Gaussian, Gamma, and Inverse Gaussian mixtures with unequal or smooth component standard deviations: protects algorithm to converge into likelihood spikes by stopping the EM algorithm if one of the component standard deviations (shape parameters, resp.), divided by the fitted mass points, falls below (exceeds, resp.) a certain threshold, which is 0.000001*spike.protect (10^6*spike.protect, resp.) Setting spike.protect=0 means disabling the spike protection. If set, then spike.protect=1 is recommended. Note that the displayed disparity may not be correct when convergence is not achieved. This can be checked with EMconverged.
sdev	optional; specifies standard deviation for normally distributed response. If unspecified, it will be estimated from the data.
shape	optional; specifies shape parameter for Gamma-and IG-distributed response. For Gamma, setting shape=1 gives an exponential distribution. If unspecified, it will be estimated from the data.
plot.opt	if equal to zero, then no graphical output is given. For plot.opt=1 the development of the disparity $-2\log L$ over iteration number is plotted, for plot.opt=2 the EM trajectories are plotted, and for plot.opt=3 both plots are shown.
verbose	if set to FALSE, no printed output is given during function execution. Useful for tolfind.
...	generic options for the glm function. Not all options may be supported under any circumstances.

Details

The nonparametric maximum likelihood (NPML) approach was introduced in Aitkin (1996) as a tool to fit overdispersed generalized linear models. The idea is to approximate the unknown and unspecified distribution of the random effect by a discrete mixture of exponential family densities, leading to a simple expression of the marginal likelihood which can then be maximized using a standard EM algorithm.

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). `allvc` is restricted to 2-level models.

The number of components k of the finite mixture has to be specified beforehand. When option `'gq'` is set, then Gauss-Hermite masses and mass points are used, assuming implicitly a normally distributed random effect. When option `'np'` is chosen, the EM algorithm uses the Gauss-Hermite masses and mass points as starting points. The position of the starting points can be concentrated or extended by setting `tol` smaller or larger than one, respectively.

Fitting random coefficient models (Aitkin, Francis & Hinde, 2009, pp. 496, p. 514) is possible by specifying the random term explicitly. Note that the setting random= ~ x gives a model with a random slope and a random intercept, and that only one random coefficient can be specified. The option `random.distribution` is restricted to np in this case, i.e. Gaussian Quadrature is not supported for random coefficient models (see also Aitkin, Francis & Hinde (2005), page 475 bottom).

As for `glm`, there are three different ways of specifying a binomial model, namley through

- a two-column matrix before the '~' symbol, specifying the counts of successes and non-successes.
- a vector of proportions of successes before the '~' symbol, and the associated number of trials provided in the `weights` argument.
- a two-level factor before the '~' symbol (only for Bernoulli-response).

The weights have to be understood as frequency weights, i.e. setting all weights in `alldist` equal to 2 will duplicate each data point and hence double the disparity and deviance.

The Inverse Gaussian (IG) response distribution is parametrized as usual through the mean and a scaling parameter. We refer to the latter, which is the inverse of the dispersion parameter in exponential family formulation, as shape. The canonical "1/mu^2" link is supported, but it is quite tenuous since the linear predictor is likely to become negative after adding the random effect. The `log` link behaves more reliably for this distribution.

For $k \geq 54$, mass points with negligible mass (i.e. < 1e-50) are omitted. The maximum number of 'effective' mass points is then 198.

Value

The function alldist produces an object of class glmmNPML (if `random.distributon` is set to 'np') or glmmGQ ('gq'). Both objects contain the following 29 components:

coefficients   a named vector of coefficients (including the mass points). In case of Gaussian quadrature, the coefficient given at z corresponds to the standard deviation of the mixing distribution.

residuals   the difference between the true response and the empirical Bayes predictions.

fitted.values   the empirical Bayes predictions (Aitkin, 1996b) on the scale of the responses.

family   the 'family' object used.

linear.predictors

        the extended linear predictors $\hat{\eta}_{ik}$.

disparity	the disparity (-2logL) of the fitted mixture regression model.
deviance	the deviance of the fitted mixture regression model.
null.deviance	the deviance for the null model (just containing an intercept), comparable with deviance.
df.residual	the residual degrees of freedom of the fitted model (including the random part).
df.null	the residual degrees of freedom for the null model.
y	the (extended) response vector.
call	the matched call.
formula	the formula supplied.
random	the random term of the model formula.
data	the data argument.
model	the (extended) design matrix.
weights	the case weights initially supplied.
offset	the offset initially supplied.
mass.points	the fitted mass points.
masses	the mixture probabilities corresponding to the mass points.
sdev	a list of the two elements sdev$sdev and sdev$sdevk. The former is the estimated standard deviation of the Gaussian mixture components (estimated over all mixture components), and the latter gives the unequal or smooth component-specific standard deviations. All values are equal if lambda=0.
shape	a list of the two elements shape$shape and shape$shapek, to be interpreted in analogy to sdev.
rsdev	estimated random effect standard deviation.
post.prob	a matrix of posteriori probabilities.
post.int	a vector of 'posteriori intercepts' (as in Sofroniou et al. (2006)).
ebp	the empirical Bayes Predictions on the scale of the linear predictor. For compatibility with older versions.
EMiter	gives the number of iterations of the EM algorithm.
EMconverged	logical value indicating if the EM algorithm converged.
lastglm	the fitted glm object from the last EM iteration.
Misc	contains additional information relevant for the summary and plot functions, in particular the disparity trend and the EM trajectories.

If a binomial model is specified by giving a two-column response, the weights returned by weights are the total numbers of cases (factored by the supplied case weights) and the component y of the result is the proportion of successes.

As a by-product, alldist produces a plot showing the disparity in dependence of 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 for GQ.

Note

In contrast to the GLIM 4 version, this R implementation uses for Gaussian (as well Gamma and IG) mixtures by default a damping procedure in the first cycles of the EM algorithm (Einbeck & Hinde, 2006), which stabilizes the algorithm and makes it less sensitive to the optimal choice of tol. If tol is very small (i.e. less than 0.1), it can be useful to set damp.power to values larger than 1 in order to accelerate convergence. Do not use damp.power=0, as this would mean permanent damping during EM. Using the option pluginz, one can to some extent circumvent the necessity to specify tol by giving the starting points explicitly. However, when using pluginz for normal, Gamma- or IG- distributed response, damping will be strictly necessary to ensure that the imposed starting points don't get blurred immediately due to initial fluctuations, implying that tol still plays a role in this case.

Author(s)

Originally translated from the GLIM 4 functions alldist and allvc (Aitkin & Francis, 1995) to R by Ross Darnell (2002). Modified, extended, and prepared for publication by Jochen Einbeck & John Hinde (2006).

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 nonparametric maximum likelihood estimation in general random effect models. Statistical Modelling: Proceedings of the 11th IWSM 1996, 87-94.

Aitkin, M. (1999). A general maximum likelihood analysis of variance components in generalized linear models. Biometrics 55, 117-128.

Aitkin, M., Francis, B. and Hinde, J. (2009). Statistical Modelling in R. Oxford Statistical Science Series, Oxford, UK.

Einbeck, J. & Hinde, J. (2006). A note on NPML estimation for exponential family regression models with unspecified dispersion parameter. Austrian Journal of Statistics 35, 233-243.

Goldstein, H. (2003). Multilevel Statistical Models (3rd edition). Arnold, London, UK.

Hinde, J. (1982). Compound Poisson regression models. Lecture Notes in Statistics 14, 109-121.

Sofroniou, N., Einbeck, J., and Hinde, J. (2006). Analyzing Irish suicide rates with mixture models. Proceedings of the 21st International Workshop on Statistical Modelling in Galway, Ireland, 2006.

See Also

glm, summary.glmmNPML, predict.glmmNPML family.glmmNPML, plot.glmmNPML.

Examples

```
# The first three examples (galaxy data, toxoplasmosis data , fabric faults)
# are based on GLIM examples in Aitkin et al. (2005), and the forth example using
```

```
# the Hospital-Stay-Data (Rosner, 2000) is taken from Einbeck & Hinde (2006).
# The fifth data example using the Oxford boys is again inspired by Aitkin et al. (2005).
# The sixth example on Irish suicide rates is taken from Sofroniou et al. (2006).


# The galaxy data
  data(galaxies, package="MASS")
  gal<-as.data.frame(galaxies)
  galaxy.np6 <- alldist(galaxies/1000~1, random=~1, random.distribution="np",
      data=gal, k=6)
  galaxy.np8u <- alldist(galaxies/1000~1, random=~1, random.distribution="np",
      data=gal, k=8, lambda=0.99)
  round(galaxy.np8u$sdev$sdevk, digits=3)
  # [1]  0.912 0.435 0.220 0.675 1.214 0.264 0.413 0.297

# The toxoplasmosis data
  data(rainfall)
  rainfall$x<-rainfall$Rain/1000
  rainfall$x2<- rainfall$x^2; rainfall$x3<- rainfall$x^3
  toxo.np3<- alldist(cbind(Cases,Total-Cases) ~ x+x2+x3, random=~1,
      random.distribution="np", family=binomial(link=logit), data=rainfall, k=3)
  toxo.np3x<- alldist(cbind(Cases,Total-Cases) ~ x, random=~x,
      random.distribution="np", family=binomial(link=logit), data=rainfall, k=3)
  # is the same as
  toxo.np3x<- alldist(Cases/Total ~ x, random = ~x, weights=Total,
      family=binomial(link=logit), data=rainfall, k=3)
  # or
  toxo.np3x<-update(toxo.np3, .~.-x2-x3, random = ~x)

# The fabric faults data
  data(fabric)
  coefficients(alldist(y ~ x, random=~1, family=poisson(link=log),
      random.distribution="gq", data= fabric, k=3, verbose=FALSE))
  # (Intercept)          x          z
  #  -3.3088663   0.8488060   0.3574909

# The Pennsylvanian hospital stay data
  data(hosp)
  fitnp3<-  alldist(duration~age+temp1, data=hosp, k=3, family=Gamma(link=log),
      tol=0.5)
  fitnp3$shape$shape
  # [1] 50.76636
  fitnp3<-  alldist(duration~age+temp1, data=hosp, k=3, family=Gamma(link=log),
      tol=0.5, lambda=0.9)
  fitnp3$shape$shapek
  # [1]  49.03101  42.79522 126.64077

# The Oxford boys data
  data(Oxboys, package="nlme")
  Oxboys$boy <- gl(26,9)
  allvc(height~age, random=~1|boy, data=Oxboys, random.distribution='gq', k=20)
  allvc(height~age, random=~1|boy, data=Oxboys,random.distribution='np',k=8)
  # with random coefficients:
```

```
  allvc(height~age,random=~age|boy, data=Oxboys, random.distribution='np', k=8)

# Irish suicide data
  data(irlsuicide)
  # Crude rate model:
  crude<- allvc(death~sex* age, random=~1|ID, offset=log(pop),
      k=3, data=irlsuicide, family=poisson)
  crude$disparity
  # [1] 654.021
  # Relative risk model:
  relrisk<- allvc(death~1, random=~1|ID, offset=log(expected),
      k=3, data=irlsuicide, family=poisson)
  relrisk$disparity
  # [1] 656.4955
```

---

dkern   *Aitchison-Aitken kernel*

Description

Discrete kernel for categorical data with k unordered categories.

Usage

```
dkern(x, y, k, lambda)
```

Arguments

x	categorical data vector
y	postive integer defining a fixed category
k	positive integer giving the number of categories
lambda	smoothing parameter

Details

This kernel was introduced in Aitchison & Aitken (1976); see also Titterington (1980).

The setting lambda =1/k corresponds to the extreme case 'maximal smoothing', while lambda = 1 means 'no smoothing'. Statistically sensible settings are only $1/k \le$ lambda ≤ 1.

Author(s)

Jochen Einbeck (2006)

References

Aitchison, J. and Aitken, C.G.G. (1976). Multivariate binary discrimination by kernel method. Biometrika 63, 413-420.

Titterington, D. M. (1980). A comparative study of kernel-based density estimates for categorical data. Technometrics, 22, 259-268.

Examples

```
k<-6;
dkern(1:k,4,k,0.99)
# Kernel centered at the 4th component with a very small amount of smoothing.


## The function is currently defined as
function(x,y,k,lambda){
ifelse(y==x, lambda, (1-lambda)/(k-1))
  }
```

---

fabric   *The Fabric Data*

Description

The data are 32 observations on faults in rolls of fabric

Usage

```
data(fabric)
```

Format

A data frame with 32 observations on the following 3 variables.

leng the length of the roll : a numeric vector

y the number of faults in the roll of fabric : a discrete vector

x the log of the length of the roll : a numeric vector

Details

The data are 32 observations on faults in rolls of fabric taken from Hinde (1982) who used the EM algorithm to fit a Poisson-normal model. The response variable is the number of faults in the roll of fabric and the explanatory variable is the log of the length of the roll.

Note

This data set and help file is an identical copy of the fabric data in package **gamlss.data**.

Source

John Hinde.

References

Hinde, J. (1982) Compound Poisson regression models: in *GLIM* 82, *Proceedings of the International Conference on Generalized Linear Models*, ed. Gilchrist, R., 109–121, Springer: New York.

Examples

```
data(fabric)
attach(fabric)
plot(x,y)
detach(fabric)
```

---

family.glmmNPML   *Methods for objects of class glmmNPML or glmmGQ*

Description

Methods for the generic `family` and `model.matrix` functions

Usage

```
## S3 method for class 'glmmNPML'
family(object, ...)
## S3 method for class 'glmmGQ'
family(object, ...)
## S3 method for class 'glmmNPML'
model.matrix(object, ...)
## S3 method for class 'glmmGQ'
model.matrix(object, ...)
```

Arguments

object   object of class `glmmNPML` or `glmmGQ`.

...   further arguments, ensuring compability with generic functions.

Note

The generic R functions `update()`, `coefficients()`, `coef()`, `fitted()`, `fitted.values()`, and `df.residual()` can also be applied straightforwardly on all objects of class `glmmNPML` or `glmmGQ`. They are not listed above as they use the generic default functions and are not separately implemented.

Explicit implementations exist for `predict`, `summary`, `print`, and `plot`, and these functions are explained in the corresponding help files.

Author(s)

Jochen Einbeck and John Hinde (2007)

See Also

[`summary.glmmNPML`](), [`predict.glmmNPML`](), [`family`](), [`model.matrix`](), [`update`](), [`coefficients`](), [`alldist`]().

---

gqz   *Gauss-Hermite integration points*

Description

Calculate Gaussian Quadrature points for the Normal distribution using the abscissas and weights for Hermite integration.

Usage

```
gqz(numnodes=20, minweight=0.000001)
```

Arguments

numnodes	theoretical number of quadrature points.
minweight	locations with weights that are less than this value will be omitted.

Details

The conversion of the locations and weights is given in Lindsey (1992, page 169:3) and Skrondal & Rabe-Hesketh (2004, page 165:1). The argument numnodes is the theoretical number of quadrature points, locations with weights that are less than the argument `minweight` will be omitted. The default value of `minweight=0.000001` returns 14 masspoints for the default `numnodes=20` as in Aitkin, Francis & Hinde (2005).

Value

A list with two vectors:

location	locations of mass points
weight	masses

Author(s)

Nick Sofroniou (2005)

References

Aitkin, M., Francis, B. and Hinde, J. (2005). Statistical Modelling in GLIM 4. Second Edition, Oxford Statistical Science Series, Oxford, UK.

Lindsey, J. K. (1992). The Analysis of Stochastic Processes using GLIM. Berlin: Springer-Verlag.

Skrondal, A. and Rabe-Hesketh, S. (2004). Generalized latent variable modelling. Boca Raton: Chapman and Hall/CRC.

See Also

[alldist](), [allvc]()

Examples

```
gqz(20, minweight=1e-14)
  # gives k=20 GH integration points. These are used in alldist
  # and allvc as fixed mass point locations when working with
  # option random.distribution='gq', and serve as EM starting points
  # otherwise.
```

---

hosp   *The Pennsylvanian Hospital Stay Data*

Description

The data, 25 observations, are a subset from a larger data set collected on persons discharged from a selected Pennsylvania hospital as part of a retrospective chart review of antibiotic use in hospitals (Towensend et al., 1979, Rosner, 2000).

Usage

```
data(hosp)
```

Format

A data frame with 25 observations on the following 9 variables. All variables are given as numerical vectors.

id patient ID.

duration the total number of days patients spent in hospital.

age age of patient in whole years.

sex gender: 1=M, 2=F.

temp1 first temperature following admission.

wbc1 first WBC count ($\times 10^3$) following admission. [WBC= white blood cells].

antib received antibiotic: 1=yes, 2=no.

bact received bacterial culture: 1=yes, 2=no.

serv service: 1 =med., 2=surg.

Warnings

Don't confuse with the Barcelona 'Hospital stay data' aep in package **gamlss**.

Source

B. Rosner, Harvard University.

References

Rosner, B. (2000). Fundamentals of Biostatistics. Thomson Learning, Duxbury, CA, USA.

Townsend, T.R., Shapiro, M., Rosner, B., & Kass, E. H. (1979). Use of antimicrobial drugs in general hospitals. I. Description of population and definition of methods. Journal of Infectious Diseases 139 , 688-697.

Examples

```
data(hosp)
glm1<- glm(duration~age+temp1+wbc1, data=hosp, family=Gamma(link=log))
```

---

irlsuicide   *Irish Suicide Data*

Description

Suicide Rates in the Republic of Ireland 1989-1998.

Usage

```
data(irlsuicide)
```

Format

A data frame with 104 observations on the following 8 variables.

Region a factor with levels Cork , Dublin , EHB - Dub., Galway, Lim., Mid HB, MWHB-Lim., NEHB, NWHB, SEHB-Wat., SHB-Cork, Waterf., WHB-Gal..

ID a factor with levels 1 2 3 4 5 6 7 8 9 10 11 12 13 corresponding to Regions.

pop a numeric vector giving the population sizes (estimated for 1994).

death a numeric vector giving the total number of deaths.

sex a factor for gender with levels 0 (female) and 1 (male).

age a factor for age with levels 1 (0-29), 2 (30-39), 3 (40-59), 4 (60+ years).

smr a numeric vector with standardized mortality ratios (SMRs)

expected a numeric vector with 'expected' number of cases obtained from a reference population (Ahlbom, 1993).

Details

The data set is examined in Sofroniou et al. (2006), using a variance component model with regions as upper level.

Source

Institute of Public Health in Ireland (2005). All Ireland Mortality Database. Retrieved August 8, 2005.

References

Ahlbom, A., (1993). Biostatistics for Epidemiologists. Boca Raton: Lewis Publishers.

Sofroniou, N., Einbeck, J., and Hinde, J. (2006). Analyzing Irish Suicide Rates with Mixture Models. Proceedings of the 21st Workshop on Statistical Modelling in Galway, Ireland, 2006.

Examples

```
data(irlsuicide)
library(lattice)
trellis.device(color=FALSE)
plot2age<-rep(gl(4,2),13)
xyplot(irlsuicide$death/irlsuicide$pop~plot2age|irlsuicide$Region,
    pch=(1+(irlsuicide$sex==1)),xlab="age",ylab="Crude rates")
```

---

missouri   *Missouri lung cancer data*

Description

Lung cancer mortality in the 84 largest Missouri cities, for males aged 45-54, 1972-1981. Data presented in Tsutakawa (1985).

Usage

```
data(missouri)
```

Format

A data frame with 84 observations on the following 2 variables.

Size population of the city.

Deaths number of lung cancer deaths.

Details

The data set was analyzed using a Poisson model with normal random effect in Tsutakawa (1985), and using a binomial logit model with unspecified random effect distribution in Aitkin (1996b). Aitkin fitted this model with GLIM4.

Source

Tsutakawa, R. (1985).

References

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.

Tsutakawa, R. (1985). Estimation of Cancer Mortalilty Rates: A Bayesian Analysis of Small Frequencies. Biometrics 41, 69-79.

Examples

```
data(missouri)
alldist(Deaths~1, offset=log(Size), random=~1, k=2,
    family=poisson(link='log'), data=missouri)
```

| plot.glmmNPML | *Plot Diagnostics for objects of class glmmNPML or glmmGQ* |

Description

The functions `alldist` and `allvc` produce objects of type `glmmGQ`, if Gaussian quadrature (Hinde, 1982, `random.distribution="gq"`) was applied for computation, and objects of class `glmmNPML`, if parameter estimation was carried out by nonparametric maximum likelihood (Aitkin, 1996a, `random.distribution="np"`). The functions presented here give some useful diagnostic plotting functionalities to analyze these objects.

Usage

```
## S3 method for class 'glmmNPML'
plot(x, plot.opt = 15, noformat=FALSE, ...)
## S3 method for class 'glmmGQ'
plot(x, plot.opt = 3, noformat=FALSE, ...)
```

Arguments

x   a fitted object of class `glmmNPML` or `glmmGQ`.

plot.opt   an integer with values $0 \leq$ `plot.opt` ≤ 15.

| noformat | if TRUE, then any formatting of the plots is omitted (useful if the user wants to include the plots into a panel of several other plots, possibly generated by other functions). |
| ... | further arguments which will mostly not have any effect (and are included only to ensure compatibility with the generic plot()- function.) |

Details

See the help pages to alldist and the vignette (Einbeck & Hinde, 2007). It is sufficient to write plot instead of plot.glmmNPML or plot.glmmGQ, since the generic plot function provided in R automatically selects the right model class.

Value

For class glmmNPML: Depending on the choice of plot.opt, a subset of the following four plots:

1	Disparity trend.
2	EM Trajectories.
3	Empirical Bayes Predictions against observed response.
4	Individual posterior probabilities.

The number given in plot.opt is transformed into a binary number indicating which plots are to be selected. The first digit (from the right!) refers to plot 1, the second one to plot 2, and so on. For example, plot.opt=4 gives the binary number 0100 and hence selects just plot 3.

For class glmmGQ: Depending on the choice of plot.opt, a subset of plots 1 and 3. Again, the number is transformed into binary coding, yielding only the disparity trend for plot.opt=1, only the EBP's for plot.opt=2, and both plots for plot.opt=3.

Author(s)

Jochen Einbeck and John Hinde (2007)

References

Aitkin, M. (1996a). A general maximum likelihood analysis of overdispersion in generalized linear models. Statistics and Computing 6, 251-262.

Einbeck, J., and Hinde, J.: Nonparametric maximum likelihood estimation for random effect models in R. Vignette to R package **npmlreg**. Type vignette("npmlreg-v") to open it.

Hinde, J. (1982). Compound Poisson regression models. Lecture Notes in Statistics 14, 109-121.

See Also

[alldist](), [allvc]()

Examples

```
data(galaxies, package="MASS")
gal<-as.data.frame(galaxies)
galaxy.np4u <- alldist(galaxies/1000~1,random=~1,k=4,tol=0.5,data=gal,lambda=1)
predict(galaxy.np4u, type="response") # EBP on scale of responses

plot(galaxy.np4u,  plot.opt=4) # plots only EBP vs.  response
plot(galaxy.np4u,  plot.opt=3) # gives same output as given by default when executing alldist
plot(galaxy.np4u)              # gives all four plots.
```

---

post	*Posterior probabilities/intercepts, and mass point classifications*

Description

Takes an object of class glmmNPML or glmmGQ and displays the posterior probabilites w_{ik} as well as the posterior intercepts (Sofroniou et. al, 2006). Further it classfies the observations to mass points according to their posterior probability. The level on which the information in all three cases is displayed can be chosen by the user via the level argument (″upper″ or ″lower″). The actual information in both cases is identical, the latter is just an expanded version of the former. In case of simple overdispersion models, the level argument is not relevant.

Usage

```
post(object, level="upper")
```

Arguments

object	an object of class glmmNPML or glmmGQ.
level	″upper″ or ″lower″.

Value

A list of the following four items:

prob	posterior probabilities (identical to object$post.prob in case of ″lower″ and for one-level models).
int	posterior intercepts (identical to object$post.int in case of ″lower″ and for one-level models).
classif	a numerical vector containing the class numbers (the order of the classes corresponds to the order of the mass points given in the output of alldist or allvc).
level	either ″lower″, ″upper″, or ″none″ (for one-level models).

Author(s)

Jochen Einbeck and John Hinde (2006)

References

Sofroniou, N., Einbeck, J., and Hinde, J. (2006). Analyzing Irish suicide rates with mixture models. Proceedings of the 21st International Workshop on Statistical Modelling in Galway, Ireland, 2006.

See Also

[alldist](#), [allvc](#)

Examples

```
data(galaxies, package="MASS")
gal <- as.data.frame(galaxies)
post(alldist(galaxies/1000~1, random=~1, data=gal, k=5))$classif
   # classifies the 82 galaxies to one of the five mass points
```

---

predict.glmmNPML   *Prediction from objects of class glmmNPML or glmmGQ*

Description

The functions `alldist` and `allvc` produce objects of type `glmmGQ`, if Gaussian quadrature (Hinde, 1982, `random.distribution="gq"`) was applied for computation, and objects of class `glmmNPML`, if parameter estimation was carried out by nonparametric maximum likelihood (Aitkin, 1996a, `random.distribution="np"`). The functions presented here give predictions from those objects.

Usage

```
## S3 method for class 'glmmNPML'
predict(object, newdata, type = "link", ...)
## S3 method for class 'glmmGQ'
predict(object, newdata, type = "link", ...)
```

Arguments

object	a fitted object of class `glmmNPML` or `glmmGQ`.
newdata	a data frame with covariates from which prediction is desired. If omitted, empirical Bayes predictions for the original data will be given.
type	if set to `link`, the prediction is given on the linear predictor scale. If set to `response`, prediction is given on the scale of the responses.
...	further arguments which will mostly not have any effect (and are included only to ensure compatibility with the generic `predict()`- function.)

Details

The predicted values are obtained by

- Empirical Bayes (Aitkin, 1996b), if newdata has not been specified. That is, the prediction on the linear predictor scale is given by $\sum \eta_{ik} w_{ik}$, whereby η_{ik} are the fitted linear predictors, w_{ik} are the weights in the final iteration of the EM algorithm (corresponding to the posterior probability for observation i to come from component k), and the sum is taken over the number of components k for fixed i.

- the marginal model, if object is of class glmmNPML and newdata has been specified. That is, computation is identical as above, but with w_{ik} replaced by the masses π_k of the fitted model.

- the analytical expression for the marginal mean of the responses, if object is of class glmmGQ and newdata has been specified. See Aitkin et al. (2009), p. 481, for the formula. This method is only supported for the logarithmic link function, as otherwise no analytical expression for the marginal mean of the responses exists.

It is sufficient to call predict instead of predict.glmmNPML or predict.glmmGQ, since the generic predict function provided in R automatically selects the right model class.

Value

A vector of predicted values.

Note

The results of the generic fitted() method correspond to predict(object, type="response"). Note that, as we are working with random effects, fitted values are never really 'fitted' but rather 'predicted'.

Author(s)

Jochen Einbeck and John Hinde (2007).

References

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. (2009). Statistical Modelling in R. Oxford Statistical Science Series, Oxford, UK.

Hinde, J. (1982). Compound Poisson regression models. Lecture Notes in Statistics 14, 109-121.

See Also

alldist, allvc, predict

Examples

```
 # Toxoplasmosis data:
    data(rainfall)
    rainfall$x<-rainfall$Rain/1000
    toxo.0.3x<- alldist(cbind(Cases,Total-Cases)~1, random=~x,
         data=rainfall, k=3, family=binomial(link=logit))
    toxo.1.3x<- alldist(cbind(Cases,Total-Cases)~x, random=~x,
         data=rainfall, k=3, family=binomial(link=logit))
    predict(toxo.0.3x, type="response", newdata=data.frame(x=2))
    # [1] 0.4608
    predict(toxo.1.3x, type="response", newdata=data.frame(x=2))
    # [1] 0.4608
    # gives the same result, as both models are equivalent and only differ
    # by a  parameter transformation.

 # Fabric faults data:
    data(fabric)
    names(fabric)
    # [1] "leng" "y"     "x"
    faults.g2<- alldist(y ~ x, family=poisson(link=log), random=~1,
        data= fabric,k=2, random.distribution="gq")
    predict(faults.g2, type="response",newdata=fabric[1:6,])
    # [1]  8.715805 10.354556 13.341242  5.856821 11.407828 13.938013
    # is not the same as
    predict(faults.g2, type="response")[1:6]
    # [1]  6.557786  7.046213 17.020242  7.288989 13.992591  9.533823
    # since in the first case prediction is done using the analytical
    # mean of the marginal distribution, and in the second case  using the
    # individual posterior probabilities in an  empirical Bayes approach.
```

---

rainfall   *Rainfall data*

Description

Toxoplasmosis data.
The rainfall data frame has 34 rows and 3 columns.

Usage

```
data(rainfall)
```

Format

This data frame contains the following columns:

Rain mm of rain
Cases cases of toxoplasmosis
Total total

Source

Luca Scrucca; originally in R package forward

References

Atkinson, A.C. and Riani, M. (2000), *Robust Diagnostic Regression Analysis*, First Edition. New York: Springer, Table A.22

---

summary.glmmNPML   *Summarizing finite mixture regression fits*

Description

These functions are the summary and print methods for objects of type glmmNPML and glmmGQ.

Usage

```
## S3 method for class 'glmmNPML'
summary(object, digits = max(3, getOption("digits") - 3), ...)
## S3 method for class 'glmmGQ'
summary(object, digits = max(3, getOption("digits") - 3), ...)
## S3 method for class 'glmmNPML'
print(x, digits=max(3,getOption('digits')-3), ...)
## S3 method for class 'glmmGQ'
print(x, digits=max(3,getOption('digits')-3),  ...)
```

Arguments

object	a fitted object of class glmmNPML or glmmGQ.
x	a fitted object of class glmmNPML or glmmGQ.
digits	number of digits; applied on various displayed quantities.
...	further arguments, which will mostly be ignored.

Details

The summary...- and print... -functions invoke the generic UseMethod(...) function and detect the right model class automatically. In other words, it is enough to write summary(...) or print(...).

Value

Prints regression output or summary on screen.

Objects returned by summary.glmmNPML or summary.glmmGQ are essentially identical to objects of class glmmNPML or glmmGQ. However, their $coef component contains the parameter standard errors and t values (taken from the GLM fitted in the last EM iteration), and they have two additional components $dispersion and $lastglmsumm providing the estimated dispersion parameter and a summary of the glm fitted in the last EM iteration.

Note

Please note that the provided parameter standard errors tend to be underestimated as the uncertainty due to the EM algorithm is not incorporated into them. According to Aitkin et al (2009), Section 7.5, page 440, more accurate standard errors can be obtained by dividing the (absolute value of the) parameter estimate through the square root of the change in disparity when omitting/not omitting the variable from the model.

Author(s)

originally from Ross Darnell (2002), modified and prepared for publication by Jochen Einbeck and John Hinde (2007).

References

Aitkin, M., Francis, B. and Hinde, J. (2009). Statistical Modelling in R. Oxford Statistical Science Series, Oxford, UK.

See Also

[alldist](), [allvc](), [summary](), [print](), [family.glmmNPML]()

---

tolfind	*Grid search over tol for NPML estimation of (generalized) random effect models*

Description

Performs a grid search to select the parameter `tol`, which is a tuning parameter for starting point selection of the EM algorithm for NPML estimation (see e.g. Aitkin, Hinde & Francis, 2009, p. 437)

Usage

```
tolfind(formula,
        random = ~1,
        family = gaussian(),
        data,
        k = 4,
        random.distribution="np",
        offset,
        weights,
        na.action,
        EMmaxit = 500,
        EMdev.change = 0.001,
        lambda = 0,
        damp = TRUE,
        damp.power = 1,
```

```
spike.protect = 1,
sdev,
shape,
plot.opt = 1,
steps = 15,
find.in.range = c(0.05, 0.8),
verbose = FALSE,
noformat = FALSE,
...)
```

Arguments

formula	a formula defining the response and the fixed effects (e.g. y ~ x).
random	a formula defining the random model. Set random=~1 to model overdispersion.
family	conditional distribution of responses: "gaussian", "poisson", "binomial", "Gamma", or "inverse.gaussian" can be set.
data	the data frame (mandatory, even if it is attached to the workspace!).
k	the number of mass points/integration points (supported are up to 600 mass points).
random.distribution	
	the mixing distribution, Gaussian Quadrature (gq) or NPML (np) can be set.
offset	an optional offset to be included in the model.
weights	optional prior weights for the data.
na.action	a function indicating what should happen when NA's occur, with possible arguments na.omit and na.fail. The default is set by the na.action setting in options() .
EMmaxit	maximum number of EM iterations.
EMdev.change	stops EM algorithm when deviance change falls below this value.
lambda	see the help file for alldist.
damp	switches EM damping on or off.
damp.power	steers degree of damping.
spike.protect	see the help file for alldist. For unequal or smooth component dispersion parameters, the setting spike.protect=1 is strongly recommended.
sdev	optional fixed standard deviation for normal mixture.
shape	optional fixed shape parameter for Gamma and IG mixtures.
plot.opt	For plot.opt=1 the EM trajectories are plotted, for plot.opt=2 the development of the disparity $-2 \log L$ over iteration number is plotted, for plot.opt=3 both plots are shown, and for plot.opt=0 none of them.
steps	number of grid points for the search of tol.
find.in.range	range for the search of tol.
verbose	If set to FALSE, no printed output is given during execution of alldist or allvc.
noformat	If TRUE, then any formatting of the plots is omitted.
...	further arguments which will be ignored.

Details

The EM algorithm for NPML estimation (Aitkin, 1996) uses the Gauss-Hermite masses and mass points as starting points. The position of the starting points can be concentrated or extended by setting tol smaller or larger than 1, respectively. The tuning parameter tol is, as in GLIM4, responsible for this scaling. A careful selection of tol may be necessary for some data sets. The reason is that NPML has a tendency to get stuck in local maxima, as the log-likelihhod function is not concave for fixed k (Boehning, 1999).

For Gaussian, Gamma, and IG mixtures this R implementation uses by default a damping procedure in the first cycles of the EM algorithm (Einbeck & Hinde, 2006), which stabilizes the algorithm and makes it less sensitive to the optimal choice of tol. Application of tolfind to check that the optimal solution has not been overlooked may nevertheless be advisable.

tolfind works for alldist and allvc. The tolfind function is mainly designed for NPML (random.distribution="np"). It can also be applied to Gaussian Quadrature (random.distribution="gq"), though tol is of little importance for this and primarily influences the speed of convergence.

Value

A list of 5 items:

MinDisparity	the minimal disparity achieved (for which EM converged).
Mintol	the tol value at which this disparity is achieved.
AllDisparities	a vector containing all disparities calculated on the grid.
Alltol	all corresponding tol values making up the grid.
AllEMconverged	a vector of Booleans indicating if EM converged for the particular tol values.

Author(s)

Jochen Einbeck & John Hinde (2006).

References

Aitkin, M. (1996). A general maximum likelihood analysis of overdispersion in generalized linear models. Statistics and Computing 6 , 251-262.

Aitkin, M., Francis, B. and Hinde, J. (2009). Statistical Modelling in R. Oxford Statistical Science Series, Oxford, UK.

Boehning, D. (1999). Computer-Assisted Analysis of Mixtures and Applications. Meta-Analysis, Disease Mapping and others. Chapman & Hall / CRC, Boca Raton, FL, USA.

Einbeck, J. & Hinde, J. (2006). A note on NPML estimation for exponential family regression models with unspecified dispersion parameter. Austrian Journal of Statistics 35, 233-243.

See Also

[alldist](), [allvc]()

Examples

```
data(galaxies, package="MASS")
gal<-as.data.frame(galaxies)
tolfind(galaxies/1000~1, random=~1, k=5, data=gal, lambda=1, damp=TRUE,
    find.in.range=c(0,1), steps=10)
# Minimal Disparity: 380.1444 at tol= 0.5
```

---

weightslogl.calc.w   *Internal npmlreg functions*

Description

These are not to be called by the user.

Usage

```
weightslogl.calc.w(p, fjk, weights)
expand(x, k)
expand.vc(x, ni)
binomial.expand(Y, k, w)
```

Arguments

p	...
fjk	...
weights	...
x	...
k	...
ni	...
Y	...
w	...

Author(s)

Ross Darnell and Jochen Einbeck.

Index