

Package ‘JointGLM’

April 17, 2009

Version 1.0-3

Date 2006-11-02

Title Joint Modelling of Mean and Dispersion (outdated, see package JointModeling)

Author Mathieu Ribatet <ribatet@lyon.cemagref.fr> and Bertrand Iooss

Maintainer Bertrand Iooss <bertrand.iooss@cea.fr>

Depends R (>= 1.8.0), mgcv

Description Some functions usefull to perform joint modelling of Mean and Dispersion through two interlinked GLM’s. This package is outdated. We recommend the use of the JointModeling package which generalizes this one.

License GPL (>= 2)

Repository CRAN

Date/Publication 2006-11-02 18:07:03

R topics documented:

eql	2
fitjoint	3
Joint Modelling	4
plot.joint	5
print.joint	6
summary.joint	7
Index	8

eql *Compute the Extended Quasi-Likelihood and the Extended Quasi-Deviance*

Description

Compute the Extended Quasi-Likelihood and the Extended Quasi-Deviance for a Joint Model.

Usage

```
eql(mod.mean, mod.disp, df.adj = TRUE)
```

Arguments

mod.mean, mod.disp	Two fitted GLMs/GAMs corresponding to the mean and the dispersion components respectively.
df.adj	Logical. If <i>TRUE</i> - the default, adjustment on degree of freedom is applied for the EQD.

Details

The Extended Quasi-Likelihood (EQL) generalizes the Quasi-Likelihood as it behaves as a likelihood function on both mean and dispersion parameters. The Extended Quasi-Deviance is defined as

$$\sum_i \frac{d_i}{\phi_i} + \sum_i \log(2\pi\phi_i V(y_i))$$

where d_i is the deviance contribution of the i -th observation, ϕ_i is the fitted values of dispersion parameter for the i -th observation, V is the variance function of the mean GLM and y_i is the i -th observation.

Value

This function returns the Extended Quasi-Likelihood and the Extended Quasi-Deviance for a joint model.

Author(s)

Mathieu Ribatet and Bertrand Iooss

References

McCullagh P. and Nelder J. A. (1987). *Generalized Linear Models Second Edition*. London: Chapman and Hall. ISBN 0-412-31760-5.

fitjoint

Joint Modelling Fitting

Description

Fit two interlinked GLMs or GAMS to model dependently the Mean and the Dispersion.

Usage

```
fitjoint(model, form.mean, form.disp, data, family.mean =
gaussian, family.disp = Gamma(link='log'), eps = 10^(-6), iter.max =
100, maxit = 50, reml = TRUE)
```

Arguments

model	A character string. Must be one of "glm" and "gam" for GLMs and GAMS models respectively.
form.mean, form.disp	R <i>formula</i> put in a character type way which specifies the linear model for the mean and the dispersion respectively.
data	A <i>data.frame</i> for the data. The first column corresponds to the response observations, while others to the explicative variables.
family.mean, family.disp	The <i>family</i> for the mean and dispersion. See family .
eps	Optional numeric. The precision for the convergence test.
iter.max, maxit	Numerics. The maximum number of iteration in the Joint Modelling algorithm and glm.fit function respectively.
reml	Logical. If <i>TRUE</i> - the default, the Restricted Maximum Likelihood Estimation is used.

Details

The estimation procedure of the Joint Model is based on the *see-saw* algorithm. This procedure is explained in *McCullagh and Nelder, 1987*. The pair of models is fitting thanks to the Extended Quasi-Deviance criterion. We start by setting the dispersion parameter for the mean to 1 and fitting the model for the mean. We then form d the response variable for dispersion and fit the dispersion GLM/GAM. From the fitted values of the dispersion GLM/GAM, we form prior weights given by the inverse of these fitted values and refit the model for the mean. The alternation of fits continues until stability is reached. We say that stability is reached when the EQD variation - in percent - between two iteration is lower than an ϵ .

Value

This function returns an object of class `c("joint", mod)` where `mod` is either `"glm"` or `"gam"`. In particular this object is a list with four components. The first two are *glm/gam objects* corresponding to the final estimate of the mean and dispersion GLMs/GAMs component. A component *iterations* corresponding to the number of iterations in the Joint Modelling fitting algorithm. And a component *eql* corresponding to the Extended Quasi-Likelihood for this Joint Model Object.

Author(s)

Mathieu Ribatet and Bertrand Iooss

References

Lee, Y and Nelder, J. A. (2003). *Robust Design via Generalized Linear Models*. Journal of Quality Technology, 35:2-12. McCullagh P. and Nelder J. A. (1987). *Generalized Linear Models Second Edition*. London: Chapman and Hall. ISBN 0-412-31760-5.

Examples

```
X <- rnorm(500, 5)
mu <- 2*X + 1
sigma2 <- exp(-X)
Y <- rnorm(500, mu, sqrt(sigma2))
data.frm <- data.frame(Y=Y, X=X)
ajust <- fitjoint("glm", 'Y~X', 'd~X', data = data.frm)
ajust
```

Joint Modelling *Hat values for the Joint model*

Description

Computes hat values for the joint model.

Usage

```
hat.glm(glm)
```

Arguments

`glm` A object of the class *glm*.

Details

The expression of the generalized form for the *hat matrix* is given by

$$H = W^{1/2} X (X^T W X)^{-1} X^T W^{1/2}$$

where W is diagonal matrix representing the weights, X a matrix corresponding to explicative variables realisations.

Value

This function returns the *hat values* i.e. the diagonal of the *hat matrix*.

Author(s)

Mathieu Ribatet and Bertrand Iooss

 plot.joint

 Model Checking

Description

Graphicals representation to help in model checking.

Usage

```
## S3 method for class 'joint':
plot(x, comp = "mean", var = NULL, res = 'standard',
     which = 1:5, ask = nb.fig < length(which) && dev.interactive(), smooth =
     TRUE, ...)
obs.vs.model(x.joint, plot.disp = FALSE, ...)
rstand.vs.linpred(x, smooth = TRUE, ...)
res.vs.explvar(x, var, res = 'standard', smooth = TRUE, ...)
absres.vs.fitted(x, res = 'standard', smooth = TRUE, ...)
adjvar.vs.linpred(x, smooth = TRUE, ...)
qqglm(x, ...)
```

Arguments

<code>x, x.joint</code>	A fitted GLM/GAM object or a fitted joint model.
<code>comp</code>	The component for the joint model. Must be one of "mean" or "disp".
<code>var</code>	Character. The name of the explanatory variable selected.
<code>res</code>	Character. Should be <i>standard</i> , <i>student</i> or <i>brut</i> . The residual type considered.
<code>which</code>	Numeric vector. Which plot must be produced.
<code>ask</code>	Logical. If TRUE, user is asked before a new plot is produced.
<code>smooth</code>	Logical. If TRUE - the default, a smoothing curve is plotted thanks to the lowess function.
<code>...</code>	Optional parameters to be passed to plot , abline and qqnorm functions.
<code>plot.disp</code>	Logical. Should the +/- standard deviation error bar be plotted around the point selected by the identify function.

Details

The function `plot.joint` is a special function to compute all the checking plots. Checking plots implemented are :

1. Observations functions of Fitted values
2. Standardized Residuals functions of Linear predictor
3. Residuals functions of an explanatory variable
4. Absolute Residuals functions of Fitted values
5. Adjusted Dependent Variable functions of Linear predictor
6. *QQ-plot* of the residuals

Several conclusions can be established thanks to these plots. One can refer to *McCullagh and Nelder, 1987* for interpretation.

Value

Return a (several) graphic window(s).

Author(s)

Mathieu Ribatet and Bertrand Iooss

References

McCullagh P. and Nelder J. A. (1987). *Generalized Linear Models Second Edition*. London: Chapman and Hall. ISBN 0-412-31760-5.

Examples

```
X <- rnorm(500, 5)
mu <- 2*X + 1
sigma2 <- exp(-X)
Y <- rnorm(500, mu, sqrt(sigma2))
data.frm <- data.frame(Y=Y, X=X)
ajust <- fitjoint("glm", 'Y~X', 'd~X', data = data.frm)
layout(matrix(c(1,1,2,2,3,3,4,4,0,5,5,0),3,byrow=TRUE))
plot(ajust, var = 'X')
```

print.joint

Printing joint object

Description

Printing joint object

Usage

```
## S3 method for class 'joint':  
print(x, ...)
```

Arguments

x An object of class "joint". Most often the return of the `fitjoint` function.
... Other option to be passed to the `plot` function.

Author(s)

Mathieu Ribatet and Bertrand Iooss

See Also

`fitjoint`

summary.joint *Summary for objects of class joint*

Description

Summary for joint objects

Usage

```
## S3 method for class 'joint':  
summary(object, ...)
```

Arguments

object An object of class "joint". Most often this object is returned by the `fitjoint` function.
... Other options to be passed to the `summary` function.

Value

Several useful informations about the fitted "joint" object.

Author(s)

Mathieu Ribatet and Bertrand Iooss

See Also

`fitjoint`, `\joint{summary}`

Index

- *Topic **hplot**
 - `plot.joint`, 5
- *Topic **htest**
 - `eql`, 1
 - `fitjoint`, 2
 - Joint Modelling, 4
- *Topic **print**
 - `print.joint`, 6
 - `summary.joint`, 7
- `abline`, 5
- `absres.vs.fitted(plot.joint)`, 5
- `adjvar.vs.linpred(plot.joint)`, 5
- `eql`, 1
- `family`, 3
- `fitjoint`, 2, 6, 7
- `glm.fit`, 3
- `hat.glm(Joint Modelling)`, 4
- `identify`, 5
- Joint Modelling, 4
- `joint.gam(fitjoint)`, 2
- `joint.glm(fitjoint)`, 2
- `lowess`, 5
- `obs.vs.model(plot.joint)`, 5
- `plot`, 5, 6
- `plot.joint`, 5, 5
- `print.joint`, 6
- `qqglm(plot.joint)`, 5
- `qqnorm`, 5
- `res.vs.explvar(plot.joint)`, 5
- `rstand.vs.linpred(plot.joint)`, 5
- `summary`, 7
- `summary.joint`, 7