Package ‘mcglm’

March 30, 2016

Type Package

Title Fit multivariate covariance generalized linear models

Version 0.2.0

Date 2016-01-07

Description This package fit multivariate covariance generalized linear models (McGLM).

Depends R (>= 3.2.1)

Suggests testthat, plyr, lattice, latticeExtra, knitr, rmarkdown, networkD3, MASS, mvtnorm, tweedie, devtools

Imports Matrix, assertthat

License GPL-3 | file LICENSE

LazyData TRUE

URL `https://github.com/wbonat/mcglm`

BugReports `https://github.com/wbonat/mcglm/issues`

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Author Wagner Hugo Bonat [aut, cre],
Walmes Marques Zeviani [ctb],
Fernando de Pol Mayer [ctb]

Maintainer Wagner Hugo Bonat <`wbonat@ufpr.br`>

R topics documented:

ahs *Australian health survey*

Description

The Australian health survey was used by Bonat and Jorgensen (2015) as an example of multivariate count regression model. The data consists of five count response variables concerning health system access measures and nine covariates concerning social conditions in Australian for 1987-88.

- `sex` - Factor with levels `male` and `female`.
- `age` - Respondent's age in years divided by 100.
- `income` - Respondent's annual income in Australian dollars divided by 1000.
- `levyplus` - Coded factor. If respondent is covered by private health insurance fund for private patients in public hospital with doctor of choice (1) or otherwise (0).
- `freepoor` - Coded factor. If respondent is covered by government because low income, recent immigrant, unemployed (1) or otherwise (0).
- `freerepa` - Coded factor. If respondent is covered free by government because of old-age or disability pension, or because invalid veteran or family of deceased veteran (1) or otherwise (0).
- `illnes` - Number of illnesses in past 2 weeks, with 5 or illnesses coded as 5.
- `actdays` - Number of days of reduced activity in the past two weeks due to illness or injury.
- `hscore` - Respondent's general health questionnaire score using Goldberg's method. High score indicates poor health.
- `chcond` - Factor with three levels. If respondent has chronic condition(s) and is limited in activity (`limited`), or if the respondent has chronic condition(s) but is not limited in activity (`nonlimited`) or otherwise (`otherwise`, reference level).
- `Ndoc` - Number of consultations with a doctor or specialist (response variable).
- `Nndoc` - Number of consultations with health professionals (response variable).
- `Nadm` - Number of admissions to a hospital, psychiatric hospital, nursing or convalescence home in the past 12 months (response variable).
- `Nhosp` - Number of nights in a hospital during the most recent admission.
- `Nmed` - Total number of prescribed and non prescribed medications used in the past two days.

Usage

```
data(ahs)
```

Format

a `data.frame` with 5190 records and 17 variables.

Source

Deb, P. and Trivedi, P. K. (1997). Demand for medical care by the elderly: A finite mixture approach, Journal of Applied Econometrics 12(3):313–336.

Examples

```
library(lattice)
library(latticeExtra)

data(ahs, package="mcglm")
str(ahs)

xt <- xtabs(~age+sex, data=ahs)
mosaicplot(xt)

xt <- xtabs(~age+chcond, data=ahs)
mosaicplot(xt)

useOuterStrips(
    combineLimits(
        xyplot(Ndoc+Nndoc+Nadm+Nhosp+Nmed~age|sex,
               outer=TRUE, data=ahs,
               jitter.x=TRUE, amount=0.01,
               type=c("p", "a"),
               scales=list(y=list(relation="free")),
               ylab="Number or occurences",
               xlab="Age (years/100)")
    )
)

useOuterStrips(
    combineLimits(
        xyplot(Ndoc+Nndoc+Nadm+Nhosp+Nmed~income|sex,
               outer=TRUE, data=ahs,
               jitter.x=TRUE, amount=0.01,
               type=c("p", "a"),
               scales=list(y=list(relation="free")),
               ylab="Number or occurences",
               xlab="Age (years/100)")
    )
)
```

anova.mcglm *ANOVA method for McGLMs.*

Description

ANOVA method for object of class McGLMS.

Usage

```
## S3 method for class 'mcglm'
anova(object, ...)
```

Arguments

`object`	an object of class `mcglm`, usually, a result of a call to `mcglm()`.
`...`	additional arguments affecting the summary produced. Note that there is no extra options for mcglm object class.

Value

A `data.frame` with Chi-square statistic to test the null hypothesis of a parameter, or a set of parameters, be zero. The Wald test based on the observed covariance matrix of the parameters is used.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

coef.mcglm *Extract model coefficients for mcglm class*

Description

`coef.mcglm` is a function which extracts model coefficients from objects of `mcglm` class.

Usage

```
## S3 method for class 'mcglm'
coef(object, std.error = FALSE, response = c(NA,
  1:length(object$beta_names)), type = c("beta", "tau", "power",
  "correlation"), ...)
```

Arguments

`object`	An object of `mcglm` class.
`std.error`	Logical. If `TRUE` returns the standard errors of the estimates. Default is `FALSE`.
`response`	A numeric vector specyfing for which response variables the coefficients should be returned.
`type`	A string vector (can be 1 element length) specyfing which coefficients should be returned. Options are `"beta"`, `"tau"`, `"power"`, `"tau"` and `"correlation"`.
`...`	additional arguments affecting the summary produced. Note that there is no extra options for mcglm object class.

Value

A `data.frame` with parameters names, estimates, response number and parameters type.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

`confint.mcglm` *Confidence Intervals for mcglm*

Description

Computes confidence intervals for parameters in a fitted `mcglm` model.

Usage

```
## S3 method for class 'mcglm'
confint(object, parm, level = 0.95, ...)
```

Arguments

`object`	a fitted `mcglm` object.
`parm`	a specification of which parameters are to be given confidence intervals, either a vector of number or a vector of strings. If missing, all parameters are considered.
`level`	the nominal confidence level.
`...`	additional arguments affecting the confidence interval produced. Note that there is no extra options for `mcglm` object class.

Value

A `data.frame` with confidence intervals, parameters names, response number and parameters type.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

covprod *Cross variability matrix*

Description

Compute the cross-covariance matrix between covariance and regression parameters. Equation (11) of Bonat and Jorgensen (2015).

Usage

```
covprod(A, res, W)
```

Arguments

`A`	A matrix.
`res`	A vector of residuals.
`W`	A matrix of weights.

Author(s)

Wagner Hugo Bonat

ESS *Extract generalized error sum of squares (ESS) for multivariate covariance generalized linear models.*

Description

Extract the generalized error sum of squares for a fitted McGLM.

Usage

```
ESS(object, verbose = TRUE)
```

Arguments

`object`	an object or a list of objects representing a model of `mcglm` class.
`verbose`	Logical

Value

Returns the value of the Gaussian pseudo-loglikelihood.

Author(s)

Wagner Hugo Bonat

fitted.mcglm *Extract Model Fitted Values of McGLM*

Description

Extract fitted values for objects of `mcglm` class.

Usage

```
## S3 method for class 'mcglm'
fitted(object, ...)
```

Arguments

`object`	An object of `mcglm` class.
`...`	additional arguments affecting the summary produced. Note that there is no extra options for `mcglm` object class.

Value

Depending on the number of response variable, the function `fitted.mcglm` returns a vector (univariate models) or a matrix (multivariate models) of fitted values.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

fit_mcglm *Chaser and reciprocal likelihood algorithms.*

Description

This function implements the two manin algorithms used to fitting McGLMs. The chaser and the reciprocal likelihood algorithms. In general the chaser algorithm is faster than the reciprocal likelihood and should be preferred.

Usage

```
fit_mcglm(list_initial, list_link, list_variance, list_covariance, list_X,
  list_Z, list_offset, list_Ntrial, list_power_fixed, list_sparse, y_vec,
  correct = TRUE, max_iter, tol = 0.001, method = "rc", tunning = 0,
  verbose)
```

Arguments

`list_initial` A list of initial values for regression and covariance parameters.

`list_link` A list of link function names.

`list_variance` A list of variance function names.

`list_covariance` A list of covariance function names.

`list_X` A list of design matrices.

`list_Z` A list of knowm matrices to used on the matrix linear predictor.

`list_offset` A list of offset values.

`list_Ntrial` A list of number of trials, useful only when analysis binomial data.

`list_power_fixed` A list of logicals indicating if the power parameters should be estimated or not.

`list_sparse` A list of logicals indicating if the matrices should be set up as sparse matrices. This argument is useful only when using exponential-matrix covariance link function. In the other cases the algorithm detects automatically if the matrix should be sparse or not.

`y_vec` A vector of the response variables stacked.

`correct` A logical indicating if the algorithm will use the correction term or not.

`max_iter` Maximum number of iterations.

`tol` A numeric spcyfing the tolerance.

`method` A string specyfing the method used to fit the models (`chaser` or `rc`).

`tunning` A numeric value in general near zero for the rc method and near 1 for the chaser method. This argument control the step-length.

`verbose` A logical if TRUE print the values of the covariance parameters used on each iteration.

Value

A list with estimated regression and covariance parameters.

`GOSHO` *NA*

Description

Compute the Gosho's information criterion for multivariate covariance generalized linear models. WARNINGS: This function is limited to models with ONE response variable.

Usage

```
GOSHO(object, id, verbose = TRUE)
```

Arguments

`object` An object of mcglm class.

`id` a vector which identifies the clusters. The length and order of id should be the same as the number of observations. Data are assumed to be sorted so that observations on a cluster are contiguous rows for all entities in the formula.

`verbose` Logical.

Value

A matrix. Note that the function assumes that the data are in the correct order.

Author(s)

Wagner Hugo Bonat

`Hunting` *Hunting in Pico Basile, Bioko Island, Equatorial Guinea.*

Description

Case study analysed in Bonat et. al. (2016) concernings on data of animals hunted in the village of Basile Fang, Bioko Norte Province, Bioko Island, Equatorial Guinea. Monthly number of blue duikers and other small animals shot or snared was collected for a random sample of 52 commercial hunters from August 2010 to September 2013. For each animal caught, the species, sex, method of capture and altitude were documented. The data set has 1216 observations.

- `Alt` - Factor five levels indicating the Altitude where the animal was caught.
- `Sex` - Factor two levels `Female` and `Male`.
- `Method` - Factor two levels `Escopeta` and `Trampa`.
- `OT` - Monthly number of other small animals hunted.
- `BD` - Monthly number of blue duikers hunted.
- `Offset` - Monthly number of hunter days.
- `Hunter` - Hunter index.
- `Month` - Month index.
- `MonthCalendar` - Month using calendar numbers (1-January, ..., 12-December).
- `Year` - Year calendar (2010–2013).

Usage

```
data(Hunting)
```

Format

a `data.frame` with 1216 records and 10 variables.

Source

Bonat, et. al. (2016). Modelling the covariance structure in marginal multivariate count models: Hunting in Bioko island. The annals of Applied Statatistics, to appear.

Examples

```
library(mcglm)
library(Matrix)
data(Hunting, package="mcglm")
formu <- OT ~ Method*Alt + Sex + Alt*poly(Month, 4)
Z0 <- Diagonal(dim(Hunting)[1],1)
fit <- mcglm(linear_pred = c(formu), matrix_pred = list(list(Z0)),
             link = c("log"), variance = c("poisson_tweedie"),
             power_fixed = c(FALSE),
             offset = list(log(Hunting$Offset)), data = Hunting)
summary(fit)
```

`mcglm` *Fitting Multivariate Covariance Generalized Linear Models (McGLM)*

Description

`mcglm` is used to fit multivariate covariance generalized linear models. The models are specified by a set of lists giving a symbolic description of the linear predictor. The user can choose between a list of link, variance and covariance functions. The models are fitted using an estimating function approach, combining quasi-score functions for regression parameters and Pearson estimating function for covariance parameters. For details see Bonat and Jorgensen (2015).

Fits a multivariate covariance generalized linear models (McGLMs) to data. McGLM is a general framework for non-normal multivariate data analysis, designed to handle multivariate response variables, along with a wide range of temporal and spatial correlation structures defined in terms of a covariance link function combined with a matrix linear predictor involving known matrices. The models take non-normality into account in the conventional way by means of a variance function, and the mean structure is modelled by means of a link function and a linear predictor. The models are fitted using an efficient Newton scoring algorithm based on quasi-likelihood and Pearson estimating functions, using only second-moment assumptions. This provides a unified approach to a wide variety of different types of response variables and covariance structures, including multivariate extensions of repeated measures, time series, longitudinal, spatial and spatio-temporal structures.

Usage

```
mcglm(linear_pred, matrix_pred, link, variance, covariance, offset, Ntrial,
  power_fixed, data, control_initial = "automatic", contrasts = NULL,
  control_algorithm = list())
```

Arguments

`linear_pred` A list of formula see `formula` for details.

`matrix_pred` A list of known matrices to be used on the matrix linear predictor. Details can be obtained on `mc_matrix_linear_predictor`.

`link` A list of link functions names, see `mc_link_function` for details.

`variance` A list of variance functions names, see `mc_variance_function` for details.

`covariance` A list of covariance link functions names, current options are: identity, inverse and exponential-matrix (expm).

`offset` A list with values of offset values if any.

`Ntrial` A list with values of the number of trials on Bernoulli experiments. It is useful only for binomialP and binomialPQ variance functions.

`power_fixed` A list of logicals indicating if the values of the power parameter should be estimated or not.

`data` A dta frame.

`control_initial` A list of initial values for the fitting algorithm. See details below.

`contrasts` Extra arguments to passed to `model.matrix`.

`control_algorithm` A list of arguments to be passed for the fitting algorithm. See `fit_mcglm` for details.

Value

mcglm returns an object of class 'mcglm'.

Author(s)

Wagner Hugo Bonat

`mc_bias_corrected_std` *Bias-corrected standard error for regression parameters*

Description

Compute bias-corrected standard error for regression parameters in the context of clustered observations. It is also robust and has improved finite sample properties.

Usage

```
mc_bias_corrected_std(object, id)
```

Arguments

`object`	An object of mcglm class.
`id`	a vector which identifies the clusters. The length and order of id should be the same as the number of observations. Data are assumed to be sorted so that observations on a cluster are contiguous rows for all entities in the formula.

Value

A matrix. Note that the function assumes that the data are in the correct order.

Author(s)

Wagner Hugo Bonat

`mc_build_bdiag` *Build a block-diagonal matrix of zeros.*

Description

Build a block-diagonal matrix of zeros. Such functions is used when computing the derivatives of the Cholesky decomposition of C.

Usage

```
mc_build_bdiag(n_resp, n_obs)
```

Arguments

`n_resp`	A numeric specifyng the number of response variables.
`n_obs`	A numeric specifying the number of observations in the data set.

Details

It is an internal function.

Value

A list of zero matrices.

Author(s)

Wagner Hugo Bonat

`mc_build_C` *Build the joint covariance matrix*

Description

This function builds the joint variance-covariance matrix using the Generalized Kronecker product and its derivatives with respect to rho, power and tau parameters.

Usage

```
mc_build_C(list_mu, list_Ntrial, rho, list_tau, list_power, list_Z, list_sparse,
  list_variance, list_covariance, list_power_fixed, compute_C = FALSE,
  compute_derivative_beta = FALSE, compute_derivative_cov = TRUE)
```

Arguments

`list_mu` A list with values of the mean.

`list_Ntrial` A list with the number of trials. Usefull only for binomial responses.

`rho` Vector of correlation parameters.

`list_tau` A list with values for the tau parameters.

`list_power` A list with values for the power parameters.

`list_Z` A list of matrix to be used in the matrix linear predictor.

`list_sparse` A list with Logical.

`list_variance` A list specifying the variance function to be used for each response variable.

`list_covariance` A list specifying the covariance function to be used for each response variable.

`list_power_fixed` A list of Logical specifying if the power parameters are fixed or not.

`compute_C` Logical. Compute or not the C matrix.

`compute_derivative_beta` Logical. Compute or not the derivative of C with respect to regression parameters.

`compute_derivative_cov` Logical. Compute or not the derivative of C with respect the covariance parameters.

Value

A list with the inverse of the C matrix and the derivatives of the C matrix with respect to rho, power and tau parameters.

Author(s)

Wagner Hugo Bonat

`mc_build_omega` *Build omega matrix*

Description

This function build Ω matrix according the covariance link function.

Usage

```
mc_build_omega(tau, Z, covariance_link, sparse = FALSE)
```

Arguments

`tau`	A vector
`Z`	A list of matrices.
`covariance_link`	String specifing the covariance link function: identity, inverse, expm.
`sparse`	Logical force to use sparse matrix representation 'dsCMatrix'.

Value

A list with the Ω matrix its inverse and derivatives with respect to τ.

Author(s)

Wagner Hugo Bonat

`mc_build_sigma` *Build variance-covariance matrix*

Description

This function builds a variance-covariance matrix, based on the variance function and omega matrix.

Usage

```
mc_build_sigma(mu, Ntrial = 1, tau, power, Z, sparse, variance, covariance,
  power_fixed, compute_derivative_beta = FALSE)
```

Arguments

`mu` A numeric vector. In general the output from `mc_link_function`.

`Ntrial` A numeric vector, or NULL or a numeric specifing the number of trials in the binomial experiment. It is usefull only when using variance = binomialP or binomialPQ. In the other cases it will be ignored.

`tau` A numeric vector.

`power` A numeric or numeric vector. It should be one number for all variance functions except binomialPQ, in that case the argument specifies both p and q.

`Z` A list of matrices.

`sparse` Logical.

`variance` String specifing the variance function: constant, tweedie, poisson_tweedie, binomialP or binomialPQ.

`covariance` String specifing the covariance function: identity, inverse or expm.

`power_fixed` Logical if the power parameter is fixed at initial value (TRUE). In the case power_fixed = FALSE the power parameter will be estimated.

`compute_derivative_beta`
Logical. Compute or not the derivative with respect to regression parameters.

Value

A list with the Cholesky decomposition of Σ, Σ^{-1} and the derivative of Σ with respect to the power and tau parameters.

Author(s)

Wagner Hugo Bonat

See Also

`mc_link_function`, `mc_variance_function`, `mc_build_omega`.

---

mc_build_sigma_between

Build the correlation matrix between response variables

Description

This function builds the correlation matrix between response variable, its inverse and derivatives.

Usage

```
mc_build_sigma_between(rho, n_resp, inverse = FALSE)

mc_derivative_sigma_between(n_resp)
```

Arguments

rho	A numeric vector.
n_resp	A numeric.
inverse	Logical.

Value

A list with sigmab and its derivatives with respect to rho.

Author(s)

Wagner Hugo Bonat

mc_core_pearson *Core of the Pearson estimating function.*

Description

Core of the Pearson estimating function.

Usage

```
mc_core_pearson(product, inv_C, res)
```

Arguments

product	A matrix.
inv_C	A matrix.
res	A vector of residuals.

Details

It is an internal function.

Value

A vector

Author(s)

Wagner Hugo Bonat

`mc_correction` *Pearson correction term*

Description

Compute the correction term associated with the Pearson estimating function.

Usage

```
mc_correction(D_C, inv_J_beta, D, inv_C)
```

Arguments

`D_C` A list of matrices.

`inv_J_beta` A matrix. In general it is computed based on the output of the `[mcglm]{mc_quasi_score}`.

`D` A matrix. In general it is the output of the mc_link_function.

`inv_C` A matrix. In general the output of the mc_build_C.

Details

It is an internal function useful inside the fitting algorithm.

Value

A vector with the correction terms to be used on the Pearson estimating function.

Author(s)

Wagner Hugo Bonat

`mc_cross_sensitivity` *Cross-sensitivity*

Description

Compute the cross-sensitivity matrix between regression and covariance parameters. Equation 10 of Bonat and Jorgensen (2015).

Usage

```
mc_cross_sensitivity(Product_cov, Product_beta,
  n_beta_effective = length(Product_beta))
```

Arguments

`Product_cov` A list of matrices.

`Product_beta` A list of matrices.

`n_beta_effective` Numeric. Effective number of regression parameters.

Value

The cross-sensitivity matrix. Equation (10) of Bonat and Jorgensen (2015).

Author(s)

Wagner Hugo Bonat

`mc_cross_variability` *Compute the cross-variability matrix*

Description

Compute the cross-variability matrix between covariance and regression parameters.

Usage

```
mc_cross_variability(Product_cov, inv_C, res, D)
```

Arguments

`Product_cov` A list of matrices.

`inv_C` A matrix.

`res` A vector.

`D` A matrix.

Value

The cross-variability matrix between regression and covariance parameters.

Author(s)

Wagner Hugo Bonat

mc_derivative_cholesky *Derivatives of the Cholesky decomposition*

Description

This function compute the derivative of the Cholesky decomposition.

Usage

```
mc_derivative_cholesky(derivada, inv_chol_Sigma, chol_Sigma)
```

Arguments

derivada	A matrix.
inv_chol_Sigma	A matrix.
chol_Sigma	A matrix.

Details

It is an internal function.

Value

A list of matrix.

Author(s)

Wagner Hugo Bonat

mc_derivative_C_rho *Derivative of C with respect to rho.*

Description

Compute the derivative of the C matrix with respect to the correlation parameters rho.

Usage

```
mc_derivative_C_rho(D_Sigmab, Bdiag_chol_Sigma_within,
  t_Bdiag_chol_Sigma_within, II)
```

Arguments

`D_Sigmab`	A matrix.
`Bdiag_chol_Sigma_within`	A block-diagonal matrix.
`t_Bdiag_chol_Sigma_within`	A block-diagonal matrix.
`II`	A diagonal matrix.

Details

It is an internal function used to build the derivatives of the C matrix.

Value

A matrix.

Author(s)

Wagner Hugo Bonat

`mc_derivative_expm` *Derivative of exponential-matrix function*

Description

Compute the derivative of the exponential-matrix covariance link function.

Usage

```
mc_derivative_expm(dU, UU, inv_UU, Q, n = dim(UU)[1], sparse = FALSE)
```

Arguments

`dU`	A matrix.
`UU`	A matrix.
`inv_UU`	A matrix.
`Q`	A numeric vector.
`n`	A numeric.
`sparse`	Logical.

Details

Many arguments required by this function are provide by the `link[mcglm]{mc_dexpm}`. The argument dU is the derivative of the U matrix with respect to the models parameters. It should be computed by the user.

Value

A matrix.

Author(s)

Wagner Hugo Bonat

See Also

expm, `link[mcglm]{mc_dexp_gold}` and `link[mcglm]{mc_dexpm}`.

---

mc_derivative_sigma_beta

Derivatives of V^1/2 with respect to beta.

Description

Compute the derivatives of $V^{1/2}$ matrix with respect to the regression parameters beta.

Usage

```
mc_derivative_sigma_beta(D, D_V_sqrt_mu, Omega, V_sqrt, variance)
```

Arguments

`D`	A matrix.
`D_V_sqrt_mu`	A matrix.
`Omega`	A matrix.
`V_sqrt`	A matrix.
`variance`	A string specifying the variance function name.

Value

A list of matrices, containg the derivatives of $V^{1/2}$ with respect to the regression parameters.

Author(s)

Wagner Hugo Bonat

`mc_dexp_gold` *Exponential-matrix and its derivatives*

Description

Given a matrix M and its derivative dM the function `dexp_gold` returns the exponential-matrix $expm(M)$ and its derivative. This function is based on the `expm` function. It is not really used in the package, but I keep this function to test my own implementation based on eigen values decomposition.

Usage

```
mc_dexp_gold(M, dM)
```

Arguments

`M`	A matrix.
`dM`	A matrix.

Value

A list with two elements: $expm(M)$ and its derivatives.

Author(s)

Wagner Hugo Bonat

See Also

`expm`, `eigen`.

Examples

```
M <- matrix(c(1,0.8,0.8,1), 2,2)
dM <- matrix(c(0,1,1,0),2,2)
mcglm::mc_dexp_gold(M = M, dM = dM)
```

mc_dist *Distances models*

Description

Builds distances model matrix.

Usage

```
mc_dist(id, time, data, method = "euclidean")
```

Arguments

id	Subject index. Note that this structure was designed to deal with longitudinal data.
time	Index indicating the time.
data	Data set
method	The distance measure to be used. This must be one of "euclidean", "maximum","manhattan","canberra","binary" or "minkowski".

Value

A matrix of dgCMatrix class.

mc_expm *Exponential-matrix covariance link function*

Description

Given a matrix U the function mc_expm returns the exponential-matrix $expm(U)$ and some auxiliares matrices to compute its derivatives. This function is based on the eigen-value decomposition it means that it is very slow.

Usage

```
mc_expm(U, n = dim(U)[1], sparse = FALSE, inverse = FALSE)
```

Arguments

U	A matrix.
n	A number specifing the dimension of the matrix U. Default n = dim(U)[1].
sparse	Logical defining the class of the output matrix. If sparse = TRUE the output class will be 'dgCMatrix' if sparse = FALSE the class will be 'dgMatrix'.
inverse	Logical defining if the inverse will be computed or not.

Value

A list with $\Omega = expm(U)$ its inverse (if `inverse = TRUE`) and auxiliares matrices to compute the derivatives.

Author(s)

Wagner Hugo Bonat

See Also

`expm`, `eigen`, `link[mcglm]{mc_dexp_gold}`.

`mc_getInformation` *Getting information about model parameters*

Description

This computes all information required about the number of model parameters.

Usage

```
mc_getInformation(list_initial, list_power_fixed, n_resp)
```

Arguments

`list_initial`	A list of initial values.
`list_power_fixed`	A list of logical specyfing if the power parameters should be estimated or not.
`n_resp`	A number specyfing the nmber of response variables.

Value

The number of β's, τ's, power and correlation parameters.

Author(s)

Wagner Hugo Bonat

mc_id *Independent model structure*

Description

Builds a identity matrix to be used as a component of the matrix linear predictor.

Usage

```
mc_id(data)
```

Arguments

data	The data set to be used.

Value

A list of matrix.

mc_initial_values *Automatic initial values for McGLMs.*

Description

This function provides o list of initial values to be used while fitting McGLMs.

Usage

```
mc_initial_values(linear_pred, matrix_pred, link, variance, covariance, offset,
  Ntrial, contrasts = NULL, data)
```

Arguments

linear_pred	A list of formula see formula for details.
matrix_pred	A list of known matrices to be used on the matrix linear predictor. Details can be obtained on mc_matrix_linear_predictor.
link	A list of link functions names, see mc_link_function for details.
variance	A list of variance functions names, see mc_variance_function for details.
covariance	A list of covariance link functions names, current options are: identity, inverse and exponential-matrix (expm).
offset	A list with values of offset values if any.
Ntrial	A list with values of the number of trials on Bernoulli experiments. It is useful only for binomialP and binomialPQ variance functions.
contrasts	List of contrasts to be used in the model.matrix.
data	A data frame.

Value

Return a list of initial values to be used while fitting McGLMs.

Author(s)

Wagner Hugo Bonat

mc_link_function *Link functions*

Description

The mc_link_function is a customized call of the make.link function.

Given the name of a link function, it returns a list with two elements. The first element is the inverse of the link function applied on the linear predictor $\mu = g^{-1}(X\beta)$. The second element is the derivative of mu with respect to the regression parameters β. It will be useful when computing the quasi-score function.

Usage

```
mc_link_function(beta, X, offset, link)

mc_logit(beta, X, offset)

mc_probit(beta, X, offset)

mc_cauchit(beta, X, offset)

mc_cloglog(beta, X, offset)

mc_loglog(beta, X, offset)

mc_identity(beta, X, offset)

mc_log(beta, X, offset)

mc_sqrt(beta, X, offset)

mc_invmu2(beta, X, offset)

mc_inverse(beta, X, offset)
```

Arguments

beta	A numeric vector of regression parameters.
X	A design matrix, see model.matrix for details.

`offset` A numeric vector of offset values. It will be sum up on the linear predictor as a covariate with known regression parameter equals one ($\mu = g^{-1}(X\beta + offset)$). If no offset is present in the model, set offset = NULL.

`link` A string specifing the name of the link function. mcglm implements the following link functions: logit, probit, cauchit, cloglog, loglog, identity, log, sqrt, 1/mu^2 and inverse. A user defined link function can be used (see Details).

Details

The link function is an important component of the multivariate covariance generalized linear model, since it link the expectation of the response variable with the covariates. Let β a $px1$ regression parameter vector and X an nxp design matrix. The expected value of a response variable Y is given by

$$E(Y) = g^{-1}(X\beta),$$

where g is the link function and $\eta = X\beta$ is the linear predictor. Let D be a $n \times p$ matrix whose entries are given by the derivatives of mu with respect to β. Such matrix will be required by the fitting algorithm. The function `mc_link_function` returns a list where the first element is mu (n x 1) vector and the second D (n x p) matrix. A user defined function can be used. It must be a function with arguments `beta`, `X` and `offset` (set to `NULL` if non needed). The function must return a length 2 named list with `mu` and `D` elements as a vector and a matrix of proper dimensions dimensions.

Value

A list with two elements: mu and D.

Author(s)

Wagner Hugo Bonat

See Also

`model.matrix`, `make.link`.

Examples

```
x1 <- seq(-1, 1, l = 5)
X <- model.matrix(~ x1)
mc_link_function(beta = c(1,0.5), X = X,
                 offset = NULL, link = 'log')
mc_link_function(beta = c(1,0.5), X = X,
                 offset = rep(10,5), link = 'identity')
```

mc_list2vec *Auxiliar function transforms list to a vector.*

Description

This function takes a list of parameters and tranforms to a vector.

Usage

```
mc_list2vec(list_initial, list_power_fixed)
```

Arguments

`list_initial`	A list specifying initial values.
`list_power_fixed`	A list of logical operators specyfing if the power parameter should be estimated or not.

Details

It is an internal function, in general the users never will use this function. It will be useful, only if the user wants to implement a different variance-covariance matrix.

Value

A vector of model parameters.

Author(s)

Wagner Hugo Bonat

mc_ma *Moving average models*

Description

Builds moving average model matrix of order k.

Usage

```
mc_ma(id, time, data, order = 1)
```

Arguments

id	Subject index. Note that this structure was designed to deal with longitudinal data.
time	Index indicating the time.
data	Data set.
order	Order of the random walk model.

Value

A matrix of dgCMatrix class.

mc_matrix_linear_predictor
Matrix linear predictor

Description

Compute the matrix linear predictor.

Usage

```
mc_matrix_linear_predictor(tau, Z)
```

Arguments

tau	A numeric vector.
Z	A list of known matrices.

Details

Given a list with a set of known matrices ($Z_0, ..., Z_D$) the function mc_matrix_linear_predictor returns $U = \tau_0 Z_0 + ... + \tau_D Z_D$.

Value

A matrix.

Author(s)

Wagner Hugo Bonat

Examples

```
require(Matrix)
Z0 <- Diagonal(5, 1)
Z1 <- Matrix(rep(1,5)%*%t(rep(1,5)))
Z <- list(Z0, Z1)
mc_matrix_linear_predictor(tau = c(1,0.8), Z = Z)
```

mc_mixed *Mixed model structure*

Description

Builds a mixed model structure

Usage

```
mc_mixed(formula, data)
```

Arguments

`formula`	A formula model to build the matrix linear predictor (see Details).
`data`	The data set to be used.

Value

A list matrices.

mc_pearson *Pearson estimating function*

Description

Compute the Pearson estimating function its sensitivity and variability matrices.

Usage

```
mc_pearson(y_vec, mu_vec, Cfeatures, inv_J_beta = NULL, D = NULL,
  correct = FALSE, compute_variability = FALSE)
```

Arguments

`y_vec`	A vector.
`mu_vec`	A vector.
`Cfeatures`	A list of matrices.
`inv_J_beta`	A matrix.
`D`	A matrix.
`correct`	Logical.
`compute_variability`	Logical.

Details

Compute the Pearson estimating function its sensitivity and variability matrices. For more details see Bonat and Jorgensen (2015) equations 6, 7 and 8.

Value

A list with three components: (i) a vector of quasi-score values, (ii) the sensitivity and (iii) variability matrices associated with the Pearson estimating function.

Author(s)

Wagner Hugo Bonat

`mc_quasi_score` *Quasi-score function*

Description

Compute the quasi-score function, its sensitivy and variability matrix.

Usage

```
mc_quasi_score(D, inv_C, y_vec, mu_vec)
```

Arguments

`D`	A matrix. In general the output from `mc_link_function`.
`inv_C`	A matrix. In general the output from `mc_build_C`.
`y_vec`	A vector.
`mu_vec`	A vector.

Value

The quasi-score vector, the Sensivity and variability matrices.

Author(s)

Wagner Hugo Bonat

`mc_robust_std` *Robust standard error for regression parameters*

Description

Compute robust standard error for regression parameters in the context of clustered observations.

Usage

```
mc_robust_std(object, id)
```

Arguments

`object` An object of mcglm class.

`id` a vector which identifies the clusters. The length and order of id should be the same as the number of observations. Data are assumed to be sorted so that observations on a cluster are contiguous rows for all entities in the formula.

Value

A matrix. Note that the function assumes that the data are in the correct order.

Author(s)

Wagner Hugo Bonat

`mc_rw` *Random walk models*

Description

Builds a random walk model matrix of order k.

Usage

```
mc_rw(id, time, data, order = 1, proper = FALSE)
```

Arguments

`id` Subject index. Note that this structure was designed to deal with longitudinal data.

`time` Index indicating the time.

`data` Data set.

`order` Order of the random walk model.

`proper` Logical.

Value

A matrix of dgCMatrix class.

---

mc_sandwich *Matrix product in sandwich form*

Description

The function mc_sandwich is just an auxiliar function to compute product matrix in the sandwich form bord1 * middle * bord2. An special case appears when computing the derivative of the covariance matrix with respect to the power parameter. Always the bord1 and bord2 should be diagonal matrix. If it is not true, this product is too slow.

Usage

```
mc_sandwich(middle, bord1, bord2)

mc_sandwich_negative(middle, bord1, bord2)

mc_sandwich_power(middle, bord1, bord2)

mc_sandwich_cholesky(bord1, middle, bord2)

mc_multiply(bord1, bord2)

mc_multiply2(bord1, bord2)
```

Arguments

middle	A matrix.
bord1	A matrix.
bord2	A matrix.

Value

The matrix product bord1 * middle * bord2.

Author(s)

Wagner Hugo Bonat

`mc_sensitivity` *Sensitivity matrix*

Description

Compute the sensitivity matrix associated with the Pearson estimating function.

Usage

```
mc_sensitivity(product)
```

Arguments

`product`	A list of matrix.

Details

This function implements the equation 7 of Bonat and Jorgensen (2015).

Value

The sensitivity matrix associated with the Pearson estimating function.

Author(s)

Wagner Hugo Bonat

`mc_sic` *Compute the score information criterion (SIC) for multivariate covariance generalized linear models.*

Description

Compute the SIC for McGLMS.

Usage

```
mc_sic(object, scope, data, response, penalty = 2)
```

Arguments

`object`	an object representing a model of `mcglm` class.
`scope`	a vector containing all covariate names to be tested.
`data`	data frame containing the all variables envolved
`response`	Indicate for which response variable SIC is computed.
`penalty`	penalty term (default = 2).

Value

A data frame with SIC values for each covariate in the scope argument.

Author(s)

Wagner Hugo Bonat

---

mc_sic_covariance *Compute the score information criterion (SIC) for multivariate covariance generalized linear models.*

Description

Compute SIC for covariance parameters in McGLMS.

Usage

```
mc_sic_covariance(object, scope, idx, data, penalty = 2, response)
```

Arguments

`object`	an object representing a model of `mcglm` class.
`scope`	a list of matrices to be tested in the matrix linear predictor.
`idx`	Indicator of matrices belong to the same effect.
`data`	data frame containing all variables envolved in the model.
`penalty`	penalty term (default = 2).
`response`	Indicate for which response variable SIC is computed.

Value

A data frame with SIC values for each matrix in the scope argument.

Author(s)

Wagner Hugo Bonat

`mc_transform_list_bdiag` *Auxiliar function to compute the derivatives of the C matrix.*

Description

This function take a list of matrices and return a list of block-diagonal matrices, where the original matrices are one block non-zero of the matrix.

Usage

```
mc_transform_list_bdiag(list_mat, mat_zero, response_number)
```

Arguments

`list_mat`	A list of matrices.
`mat_zero`	A list of zero matrices. In general the output of `link[mcglm]{mc_build_bdiag}`.
`response_number`	A numeric specifying the response variable number.

Value

A list of block-diagonal matrices.

Author(s)

Wagner Hugo Bonat

`mc_updateBeta` *Updated regression parameters*

Description

This function update a list of regression parameters. It will be useful only inside the fitting algorithm.

Usage

```
mc_updateBeta(list_initial, betas, information, n_resp)
```

Arguments

`list_initial`	A list of initial values.
`betas`	A vector with actual regression parameters values.
`information`	A list with information about the number of parameters in the model. In general the output from mc_getInformation.
`n_resp`	A numeric specyfing the number of response variables.

Value

A list with updated values of the regression parameters.

Author(s)

Wagner Hugo Bonat

`mc_updateCov` *Updated covariance parameters*

Description

This function update a list of covariance parameters. It will be useful only inside the fitting algorithm.

Usage

```
mc_updateCov(list_initial, covariance, list_power_fixed, information, n_resp)
```

Arguments

`list_initial`	A list of initial values.
`covariance`	A vector with actual covariance parameters values.
`list_power_fixed`	A list of logicals indicating if the power parameter should be estimated or not.
`information`	A list with information about the number of parameters in the model. In general the output from mc_getInformation.
`n_resp`	A numeric specyfing the number of response variables.

Value

A list with updated values of the covariance parameters.

Author(s)

Wagner Hugo Bonat

`mc_variability` *Variability matrix*

Description

Compute the variability matrix associated with the Pearson estimating function.

Usage

```
mc_variability(sensitivity, product, inv_C, C, res)
```

Arguments

`sensitivity`	A matrix. In general the output from `mc_sensitivity`.
`product`	A list of matrix.
`inv_C`	A matrix. In general the output from `mc_build_C`.
`C`	A matrix. In general the output from `mc_build_C`.
`res`	A vector. The residuals vector, i.e. (y_vec - mu_vec).

Details

This function implements the equation 8 of Bonat and Jorgensen (2015).

Value

The variability matrix associated witht the Pearson estimating function.

Author(s)

Wagner Hugo Bonat

`mc_variance_function` *Variance function*

Description

Compute features related with variance functions.

Usage

```
mc_variance_function(mu, power, Ntrial, variance, inverse, derivative_power,
  derivative_mu)

mc_power(mu, power, inverse, derivative_power, derivative_mu)

mc_binomialP(mu, power, inverse, Ntrial, derivative_power, derivative_mu)

mc_binomialPQ(mu, power, inverse, Ntrial, derivative_power, derivative_mu)
```

Arguments

`mu` A numeric vector. In general the output from `mc_link_function`.

`power` A numeric value (`power` and `binomialP`) or vector (`binomialPQ`) of the power parameters.

`Ntrial` Number of trials, useful only when dealing with binomial response variables.

`variance` A string specifing the name (`power, binomialP or binomialPQ`) of the variance function.

`inverse` Logical.

`derivative_power`
Logical if compute (TRUE) or not (FALSE) the derivatives with respect to the power parameter.

`derivative_mu` Logical if compute (TRUE) or not (FALSE) the derivative with respect to the mu parameter.

Details

The function `mc_variance_function` computing three features related with the variance function. Depending on the Logical arguments, the function returns $V^{1/2}$ and its derivatives with respect to the parameters power and mu, respectivelly. The output is a named list, completely informative about what the function has been computed. For example, if `inverse = FALSE`, `derivative_power = TRUE` and `derivative_mu = TRUE`. The output will be a list, with three elements: V_sqrt, D_V_sqrt_power and D_V_sqrt_mu.

Value

A list with from one to four elements depends on the arguments.

Author(s)

Wagner Hugo Bonat

See Also

`mc_link_function`.

Examples

```
x1 <- seq(-1, 1, l = 5)
X <- model.matrix(~x1)
mu <- mc_link_function(beta = c(1, 0.5), X = X, offset = NULL,
                       link = "logit")
mc_variance_function(mu = mu$mu, power = c(2, 1), Ntrial = 1,
                     variance = "binomialPQ", inverse = FALSE,
                     derivative_power = TRUE, derivative_mu = TRUE)
```

NewBorn *Respiratory physiotherapy on premature newborns.*

Description

The NewBorn dataset consist of a prospective study to assess the effect of respiratory physiotherapy on the cardiopulmonary function of ventilated preterm newborn infants with birth weight lower than 1500 g. The data set was collected and kindly made available by the nursing team of the Waldemar Monastier hospital, Campo Largo, PR, Brazil. The NewBorn dataset was analysed in Bonat and Jorgensen (2016) as an example of mixed outcomes regression model.

- `Sex` - Factor two levels `Female` and `Male`.
- `GA` - Gestational age (weeks).
- `BW` - Birth weight (mm).
- `APGAR1M` - APGAR index in the first minute of life.
- `APGAR5M` - APGAR index in the fifth minute of life.
- `PRE` - Factor, two levels (Premature: YES; NO).
- `HD` - Factor, two levels (Hansen's disease, YES; NO).
- `SUR` - Factor, two levels (Surfactant, YES; NO).
- `JAU` - Factor, two levels (Jaundice, YES; NO).
- `PNE` - Factor, two levels (Pneumonia, YES; NO).
- `PDA` - Factor, two levels (Persistence of ductus arteriosus, YES; NO).
- `PPI` - Factor, two levels (Primary pulmonary infection, YES; NO).
- `OTHERS` - Factor, two levels (Other diseases, YES; NO).
- `DAYS` - Age (days).
- `AUX` - Factor, two levels (Type of respiratory auxiliary, HOOD; OTHERS).
- `RR` - Respiratory rate (continuous).
- `HR` - Heart rate (continuous).
- `SPO2` - Oxygen saturation (bounded).
- `TREAT` - Factor, three levels (Respiratory physiotherapy, Evaluation 1; Evaluation 2; Evaluation 3).
- `NBI` - Newborn index.
- `TIME` - Days of treatment.

Usage

```
data(NewBorn)
```

Format

a `data.frame` with 270 records and 21 variables.

Source

Bonat, et. al. (2016). Multivariate covariance generalized linear models. Journal of Royal Statistical Society - Series C, to appear.

Examples

```
library(mcglm)
library(Matrix)
data(NewBorn, package="mcglm")
formu <- SPO2 ~ Sex + APGAR1M + APGAR5M + PRE + HD + SUR
Z0 <- Diagonal(dim(NewBorn)[1],1)
fit <- mcglm(linear_pred = c(formu), matrix_pred = list(list(Z0)),
            link = c("logit"), variance = c("binomialP"),
            power_fixed = c(TRUE),
            data = NewBorn,
            control_algorithm = list(verbose = FALSE, tunning = 0.5))
summary(fit)
```

---

pAIC *Extract pseudo Akaike Information Criterion (pAIC) for multivariate covariance generalized linear models.*

Description

Extract the pAIC for a fitted McGLM.

Usage

```
pAIC(object, verbose = TRUE)
```

Arguments

`object`	an object or a list of objects representing a model of `mcglm` class.
`verbose`	Logical

Value

Returns the value of the pAIC.

Author(s)

Wagner Hugo Bonat

pKLIC *Extract pseudo Kullback-Leibler Information Criterion for multivariate covariance generalized linear models.*

Description

Extract the pKLIC for a fitted McGLM.

Usage

```
pKLIC(object, verbose = TRUE)
```

Arguments

`object`	an object or a list of objects representing a model of `mcglm` class.
`verbose`	Logical

Value

Returns the value of the pKLIC.

Author(s)

Wagner Hugo Bonat

plogLik *Extract Gaussian pseudo-loglikelihood (plogLik) for multivariate covariance generalized linear models.*

Description

Extract the Gaussian pseudo-loglikelihood for a fitted McGLM.

Usage

```
plogLik(object, verbose = TRUE)
```

Arguments

`object`	an object or a list of objects representing a model of `mcglm` class.
`verbose`	Logical

Value

Returns the value of the Gaussian pseudo-loglikelihood.

Author(s)

Wagner Hugo Bonat

plot.mcglm *Default Multivariate Covariance Generalized Linear models plotting*

Description

takes a fitted mcglm object and do plots based on residuals, influence diagnostic measures and algorithm check.

Usage

```
## S3 method for class 'mcglm'
plot(x, type = "residuals", ...)
```

Arguments

x	a fitted mcglm object.
type	Specify which graphical analysis will be performed. Options are: "residuals", "influence" and "algorithm".
...	additional arguments affecting the plot produced. Note that there is no extra options for mcglm object class.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

print.mcglm *Print method for Multivariate Covariance Generalized Linear Model*

Description

The default print method for a mcglm object.

Usage

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

Arguments

x	fitted model objects of class mcglm as produced by mcglm().
...	further arguments passed to or from other methods.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

residuals.mcglm *Residuals for Multivariate Covariance Generalized Linear Models (McGLM)*

Description

Compute residuals based on fitting mcglm models.

Usage

```
## S3 method for class 'mcglm'
residuals(object, type = "raw", ...)
```

Arguments

object	An of class mcglm, typically the result of a call to mcglm.
type	the type of residuals which should be returned. The alternatives are: "raw" (default), "pearson" and "standardized".
...	additional arguments affecting the residuals produced. Note that there is no extra options for mcglm object class.

Value

Depending on the number of response variable the function residuals.mcglm returns a vector (univariate models) or a matrix (multivariate models) of residuals values.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

RJC *NA*

Description

Compute the Rotnitzky-Jewell's information criterion for multivariate covariance generalized linear models. WARNINGS: This function is limited to models with ONE response variable.

Usage

```
RJC(object, id, verbose = TRUE)
```

Arguments

`object`	An object of mcglm class.
`id`	a vector which identifies the clusters. The length and order of id should be the same as the number of observations. Data are assumed to be sorted so that observations on a cluster are contiguous rows for all entities in the formula.
`verbose`	Logical.

Value

A matrix. Note that the function assumes that the data are in the correct order.

Author(s)

Wagner Hugo Bonat

`summary.mcglm` *Summarizing Multivariate Covariance Generalized Linear Models fits.*

Description

Summary for McGLMs objects.

Usage

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

Arguments

`object`	an object of class `mcglm`, usually, a result of a call to `mcglm`.
`...`	additional arguments affecting the summary produced. Note the there is no extra options for mcglm object class.

Value

Print an `mcglm` object.

Author(s)

Wagner Hugo Bonat, <`wbonat@ufpr.br`>

vcov.mcglm — *Calculate Variance-Covariance matrix for a fitted McGLM object.*

Description

Returns the variance-covariance matrix for all parameters of a `mcglm` fitted model object.

Usage

```
## S3 method for class 'mcglm'
vcov(object, ...)
```

Arguments

`object`	a fitted model `mcglm` object.
`...`	additional arguments affecting the summary produced. Note that there is no extra options for mcglm object class.

Value

A variance-covariance matrix.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

Index