

Package ‘MGLM’

March 27, 2017

Version 0.0.8

Date 2017-03-15

Title Multivariate Response Generalized Linear Models

Author Yiwen Zhang <zhangyiwen1015@gmail.com> and Hua Zhou <huazhou@ucla.edu>

Maintainer Yiwen Zhang <zhangyiwen1015@gmail.com>

Depends R (>= 3.0.0)

Imports methods, stats, parallel

Suggests ggplot2, plyr, reshape2

Description Provides functions that (1) fit multivariate discrete distributions, (2) generate random numbers from multivariate discrete distributions, and (3) run regression and penalized regression on the multivariate categorical response data. Implemented models include: multinomial logit model, Dirichlet multinomial model, generalized Dirichlet multinomial model, and negative multinomial model. Making the best of the minorization-maximization (MM) algorithm and Newton-Raphson method, we derive and implement stable and efficient algorithms to find the maximum likelihood estimates. On a multi-core machine, multi-threading is supported.

LazyLoad yes

LazyData yes

Repository CRAN

License GPL (>= 2)

NeedsCompilation no

Date/Publication 2017-03-27 15:21:50 UTC

R topics documented:

MGLM-package	2
ddirm	3
dgdirm	4
dist	5
dmn	7
dnegmn	8

kr	9
MGLMfit	10
MGLMfit-class	12
MGLMreg	13
MGLMreg-class	15
MGLMsparsereg	16
MGLMsparsereg-class	20
MGLMtune	21
MGLMtune-class	23
predict-methods	24
print-methods	24
rdirn	24
rgdirn	25
maseq	27
megmn	27

Index	29
--------------	-----------

MGLM-package	<i>Multivariate response generalized linear models</i>
--------------	--

Description

The package provides functions that (1) fit multivariate discrete distributions, (2) generate random numbers from multivariate discrete distributions, and (3) run regression and penalized regression on the multivariate categorical response data. Implemented models include: multinomial logit model, Dirichlet multinomial model, generalized Dirichlet multinomial model, and negative multinomial model. Making the best of the minorization-maximization (MM) algorithm and Newton-Raphson method, we derive and implement stable and efficient algorithms to find the maximum likelihood estimates. On a multi-core machine, multi-threading is supported.

Details

Package:	MGLM
Type:	Package
Version:	0.0.8
Date:	2017-03-15
License:	GPL (>= 2)
Depends:	R (>= 3.0.0), methods, stats, parallel

Author(s)

Yiwen Zhang and Hua Zhou

ddirm	<i>Calculate the log of the Dirichlet multinomial probability mass function</i>
-------	---

Description

Calculate the log of the Dirichlet multinomial probability mass function.

Usage

```
ddirm(Y, alpha)
```

Arguments

Y	The multivariate count matrix with dimensions $n \times d$, where $n = 1, 2, \dots$ is the number of observations and $d = 2, 3, \dots$ is the number of categories.
alpha	The parameter of the Dirichlet multinomial distribution. alpha can either be a vector or a matrix which matches the size of Y. If alpha is a vector, it will be replicated n times to match the dimension of Y.

Details

$Y = (y_1, \dots, y_d)$, $d \geq 2$, are the d category count vectors. Given the parameter vector $\alpha = (\alpha_1, \dots, \alpha_d)$, $\alpha_j > 0$, it calculates the log of the Dirichlet multinomial pmf

$$P(y|\alpha) = \binom{m}{y_1, \dots, y_d} \prod_{j=1}^d \frac{\Gamma(\alpha_j + y_j)}{\Gamma(\alpha_j)} \frac{\Gamma(\sum_{j'=1}^d \alpha_{j'})}{\Gamma(\sum_{j'=1}^d \alpha_{j'} + \sum_{j'=1}^d y_j)},$$

where $m = \sum_{j=1}^d y_j$.

For each count vector and each corresponding parameter vector α , the function `ddirm` returns the value $\log(P(y|\alpha))$. When Y is a matrix of n rows, the function returns a vector of length n .

The parameter α can be a vector of length d , such as the results from the distribution fitting. α can also be a matrix with n rows, such as the inverse link calculated from the regression parameter estimate $\exp(X\beta)$

Value

A vector of length n with value $\log(P(y|\alpha))$

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
m <- 20
alpha <- c(0.1, 0.2)
dm.Y <- rdir(m=10, m, alpha)
pdfln <- ddir(dm.Y, alpha)
```

dgdirm	<i>Calculate the log of the generalized Dirichlet multinomial probability mass function</i>
--------	---

Description

Calculate the log of the generalized Dirichlet multinomial probability mass function.

Usage

```
dgdirm(Y, alpha, beta)
```

Arguments

Y	The multivariate count matrix with dimensions $n \times d$, where $n = 1, 2, \dots$ is the number of observations and $d = 3, 4, \dots$ is the number of categories.
alpha	The parameter of the generalized Dirichlet multinomial distribution. alpha can either be a vector or a matrix which matches the size of Y. If alpha is a vector, it will be replicated n times to match the dimension of Y.
beta	The parameter of the generalized Dirichlet multinomial distribution. beta should have the same dimension as alpha.

Details

$Y = (y_1, \dots, y_d)$ are the d category count vectors. Given the parameter vector $\alpha = (\alpha_1, \dots, \alpha_{d-1})$, $\alpha_j > 0$, and $\beta = (\beta_1, \dots, \beta_{d-1})$, $\beta_j > 0$, it calculates the generalized Dirichlet multinomial pmf

$$P(y|\alpha, \beta) = \binom{m}{y_1, \dots, y_d} \prod_{j=1}^{d-1} \frac{\Gamma(\alpha_j + y_j)}{\Gamma(\alpha_j)} \frac{\Gamma(\beta_j + z_{j+1})}{\Gamma(\beta_j)} \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j + \beta_j + z_j)},$$

where $z_j = \sum_{k=j}^d y_k$ and $m = \sum_{j=1}^d y_j$.

For each count vector and each corresponding parameter vector α and β , the function `dgdirm` returns the value of $\log(P(y|\alpha, \beta))$. When Y is a matrix of n rows, the function returns a vector of length n .

The α and β parameters can be vectors, like the results from the distribution fitting function, or they can be matrices with n rows, like the estimate from the regression function multiplied by the covariate matrix $\exp(X\alpha)$ and $\exp(X\beta)$

Value

A vector of length n with value $\log(P(y|\alpha, \beta))$

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
m <- 20
alpha <- c(0.2, 0.5)
beta <- c(0.7, 0.4)
Y <- rgdirm(10, m, alpha, beta)
dgdirm(Y, alpha, beta)
```

 dist

Details of the distributions

Description

An object that specifies the distribution to be fitted by the `MGLMfit` function, or the regression model to be fitted by the `MGLMreg` or `MGLMsparsereg` functions. Can be chosen from "MN", "DM", "NegMN", or "GDM".

Details

"MN": Multinomial distribution: A multinomial distribution models the counts of d possible outcomes. The counts of categories are negatively correlated. The density of a d category count vector y with parameter $p = (p_1, \dots, p_d)$ is

$$P(y|p) = \binom{m}{y_1, \dots, y_d} \prod_{j=1}^d p_j^{y_j},$$

where $m = \sum_{j=1}^d y_j$, $0 < p_j < 1$, and $\sum_{j=1}^d p_j = 1$.

The `MGLMreg` function with `dist="MN"` calculates the MLE of regression coefficients β_j of the multinomial logit model, which has link function $p_j = \exp(X\beta_j) / (1 + \sum_{j=1}^{d-1} \exp(X\beta_j))$, $j = 1, \dots, d-1$. The `MGLMsparsereg` function with `dist="MN"` fits regularized multinomial logit model.

"DM": Dirichlet multinomial distribution:

When the multivariate count data exhibits over-dispersion, the traditional multinomial model is insufficient. Dirichlet multinomial distribution models the probabilities of the categories by a Dirichlet distribution. The density of a d category count vector y , with parameter $\alpha = (\alpha_1, \dots, \alpha_d)$, $\alpha_j > 0$, is

$$P(y|\alpha) = \binom{m}{y_1, \dots, y_d} \prod_{j=1}^d \frac{\Gamma(\alpha_j + y_j)}{\Gamma(\alpha_j)} \frac{\Gamma(\sum_j \alpha_j)}{\Gamma(\sum_j \alpha_j + \sum_j y_j)},$$

where $m = \sum_{j=1}^d y_j$.

The `MGLMfit` function with `dist="DM"` calculates the maximum likelihood estimate (MLE) of $(\alpha_1, \dots, \alpha_d)$. The `MGLMreg` function with `dist="DM"` calculates the MLE of regression coefficients β_j of the Dirichlet multinomial regression model, which has link function $\alpha_j = \exp(X\beta_j)$, $j = 1, \dots, d$. The `MGLMsparsereg` function with `dist="DM"` fits regularized Dirichlet multinomial regression model.

"GDM": Generalized Dirichlet multinomial distribution:

The more flexible Generalized Dirichlet multinomial model can be used when the counts of categories have both positive and negative correlations. The probability mass of a count vector y over m trials with parameter $(\alpha, \beta) = (\alpha_1, \dots, \alpha_{d-1}, \beta_1, \dots, \beta_{d-1})$, $\alpha_j, \beta_j > 0$, is

$$P(y|\alpha, \beta) = \binom{m}{y_1, \dots, y_d} \prod_{j=1}^{d-1} \frac{\Gamma(\alpha_j + y_j) \Gamma(\beta_j + z_{j+1})}{\Gamma(\alpha_j) \Gamma(\beta_j)} \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j + \beta_j + z_j)},$$

where $z_j = \sum_{k=j}^d y_k$ and $m = \sum_{j=1}^d y_j$.

The `MGLMfit` with `dist="GDM"` calculates the MLE of $(\alpha, \beta) = (\alpha_1, \dots, \alpha_{d-1}, \beta_1, \dots, \beta_{d-1})$. The `MGLMreg` function with `dist="GDM"` calculates the MLE of regression coefficients α_j, β_j of the generalized Dirichlet multinomial regression model, which has link functions $\alpha_j = \exp(X\alpha_j)$ and $\beta_j = \exp(X\beta_j)$, $j = 1, \dots, d-1$. The `MGLMsparsereg` function with `dist="GDM"` fits regularized generalized Dirichlet multinomial regression model.

"NegMN": Negative multinomial distribution:

Both the multinomial distribution and Dirichlet multinomial distribution are good for negatively correlated counts. When the counts of categories are positively correlated, the negative multinomial distribution is preferred. The probability mass function of a d category count vector y with parameter $(p_1, \dots, p_{d+1}, \beta)$, $\sum_{j=1}^{d+1} p_j = 1$, $p_j > 0$, $\beta > 0$, is

$$P(y|p, \beta) = \binom{\beta + m - 1}{m} \binom{m}{y_1, \dots, y_d} \prod_{j=1}^d p_j^{y_j} p_{d+1}^\beta = \frac{\beta_m}{m!} \binom{m}{y_1, \dots, y_d} \prod_{j=1}^d p_j^{y_j} p_{d+1}^\beta,$$

where $m = \sum_{j=1}^d y_j$.

The `MGLMfit` function with `dist="NegMN"` calculates the MLE of $(p_1, \dots, p_{d+1}, \beta)$. The `MGLMreg` function with `dist="NegMN"` and `regBeta=FALSE` calculates the MLE of regression coefficients $(\alpha_1, \dots, \alpha_d, \beta)$ of the negative multinomial regression model, which has link function $p_{d+1} = 1/(1 + \sum_{j=1}^d \exp(X\alpha_j))$, $p_j = \exp(X\alpha_j)p_{d+1}$, $j = 1, \dots, d$. When `dist="NegMN"` and `regBeta=TRUE`, the overdispersion parameter is linked to covariates via $\beta = \exp(X\alpha_{d+1})$, and the function `MGLMreg` outputs an estimated matrix of $(\alpha_1, \dots, \alpha_{d+1})$. The `MGLMsparsereg` function with `dist="NegMN"` fits regularized negative multinomial regression model.

Author(s)

Yiwen Zhang and Hua Zhou

See Also

`MGLMfit`, `MGLMreg`, `MGLMsparsereg`, `dmn`, `ddirm`, `dgdirm`, `dnegmn`

dmn

Calculate the log of the multinomial probability mass function

Description

Calculate the log of the multinomial probability mass function.

Usage

```
dmn(Y, prob)
```

Arguments

Y The multivariate count matrix with dimension $n \times d$, where $n = 1, 2, \dots$ is number of observations and $d = 2, \dots$ is number of categories.

prob The probability parameter of the multinomial distribution. `prob` can be either a vector of length d or a matrix with matching size of `Y`. If `prob` is a vector, it will be replicated n times to match the dimension of `Y`. If the sum(s) of `prob` is not 1, it will be automatically scaled to have sum 1.

Details

$y = (y_1, \dots, y_d)$ is a d category count vector. Given the parameter vector $p = (p_1, \dots, p_d)$, $0 < p_j < 1$, $\sum_{j=1}^d p_j = 1$, the function calculates the log of the multinomial pmf

$$P(y|p) = \binom{m}{y_1, \dots, y_d} \prod_{j=1}^d p_j^{y_j},$$

where $m = \sum_{j=1}^d y_j$.

For each count vector and each corresponding parameter vector p , the function `dmn` returns the value of $\log(P(y|p))$. When `Y` is a matrix of n rows, the function returns a vector of length n .

The parameter p can be one vector, like the result from the distribution fitting function; or, p can be a matrix with n rows, like the estimate from the regression function, $p_j = \frac{\exp(X\beta_j)}{1 + \sum_{j'=1}^{d-1} \exp(X\beta_{j'})}$, $j = 1, \dots, d-1$, $p_d = \frac{1}{1 + \sum_{j'=1}^{d-1} \exp(X\beta_{j'})}$. The d -th column of the coefficient matrix β is set to 0 to avoid the identifiability issue.

Value

A vector of length n with value $\log(P(y|p))$

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
m <- 20
prob <- c(0.1, 0.2)
dm.Y <- rdirn(n=10, m, prob)
pdfln <- dmn(dm.Y, prob)
```

dnegmn	<i>Calculate the log of the negative multinomial probability mass function</i>
--------	--

Description

Calculate the log of the negative multinomial probability mass function.

Usage

```
dnegmn(Y, prob, beta)
```

Arguments

Y	The multivariate response matrix of dimension $n \times d$, where $n = 1, 2, \dots$ is number of observations and $d = 2, 3, \dots$ is number of categories.
prob	The probability parameter of the negative multinomial distribution. prob can be either a vector of length d ($d \geq 2$) or a matrix with matching size of Y. If prob is a vector, it will be replicated n times to match the dimension of Y. The sum of each row of prob should be smaller than 1.
beta	The over dispersion parameter of the negative multinomial distribution. beta can be either a scalar or a vector of length n .

Details

$y = (y_1, \dots, y_d)$ is a d category vector. Given the parameter vector $p = (p_1, \dots, p_d)$, $p_{d+1} = 1/(1 + \sum_{j=1}^d p_j)$, and $\beta, \beta > 0$, it calculates the log of the negative multinomial pmf

$$P(y|p, \beta) = \binom{\beta + m - 1}{m} \binom{m}{y_1, \dots, y_d} \prod_{j=1}^d p_j^{y_j} p_{d+1}^\beta = \frac{\beta_m}{m!} \binom{m}{y_1, \dots, y_d} \prod_{j=1}^d p_j^{y_j} p_{d+1}^\beta,$$

where $m = \sum_{j=1}^d y_j$.

For each negative multinomial sample, and each corresponding parameter α and β , the function dnegmn returns the value of $\log(P(y|p, \beta))$. When Y is a matrix of n rows, the function returns a vector of length n .

The parameter prob can be a vector and beta is a scalar; prob can also be a matrix with n rows, and beta is a vector of length n like the estimate from the regression function multiplied by the covariate matrix.

Value

A vector of length n with value $\log(P(y|p, \beta))$

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
m <- 20
n <- 10
p <- 5
d <- 6
a <- -matrix(1,p,d)
X <- matrix(runif(n*p), n, p )
alpha <- exp(X%%a)
prob <- alpha/(rowSums(alpha)+1)
b <- exp(X%%rep(0.3,p))
Y <- rnegmn(prob=prob, beta=rep(10, n))
dnegmn(Y, prob, b)
```

kr

Khatri-Rao product of two matrices

Description

Return the Khatri-Rao product of two matrices, which is a column-wise Kronecker product.

Usage

```
kr(A, B, w, byrow)
```

Arguments

A, B	Both A and B are matrices. The two matrices should have the same number of columns. We also give the user an option to do row-wise Kronecker product, to avoid transpose. When doing row-wise Kronecker product, the number of rows of A and B should be the same.
w	The weights vector. The length of the vector should match with the dimension of the matrices. If performing column-wise Kronecker product, the length of w should be the same as the column number of A and B. If performing row-wise Kronecker product, the length of w should be the same as the row number of A and B. The default is a vector of 1 if no value provided.
byrow	A logical variable controlling whether to perform row/column-wise Kronecker product. The default is byrow=TRUE.

Details

The column/row-wise Kronecker product.

Value

A matrix of the Khatri-Rao product.

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
X <- matrix(rnorm(30), 10, 3)
Y <- matrix(runif(50), 10, 5)
C <- kr(X, Y)
```

MGLMfit

Fit multivariate discrete distributions

Description

Fit the specified multivariate discrete distribution.

Usage

```
MGLMfit(data, dist, init, weight, epsilon=1e-8, maxiters=150, display=FALSE)

DMD.DM.fit(data, init, weight, epsilon=1e-8, maxiters=150, display=FALSE)
DMD.GDM.fit(data, init, weight, epsilon=1e-8, maxiters=150, display=FALSE)
DMD.NegMN.fit(data, init, weight, epsilon=1e-8, maxiters=150, display=FALSE)
```

Arguments

<code>data</code>	A data frame or matrix containing the count data. Rows of the matrix represent observations and columns are the categories. Rows and columns of all zeros are automatically removed.
<code>dist</code>	A description of the distribution to fit. Choose from "MN", "DM", "GDM", "NegMN". See dist for details.
<code>weight</code>	An optional vector of weights assigned to each row of the data. Should be Null or a numeric vector with the length equal to the number of rows of data. If <code>weight=NULL</code> , equal weights of all ones will be assigned.
<code>init</code>	An optional vector of initial value of the parameter estimates. Should have the same dimension as the estimated parameters. See dist for details.
<code>epsilon</code>	An optional numeric controlling the stopping criterion. The algorithm terminates when the relative change in the log-likelihoods of two successive iterates is less than <code>epsilon</code> . The default value is <code>epsilon=1e-8</code> .

maxiters	An optional number controlling the maximum number of iterations. The default value is maxiters=150.
display	An optional logical variable controlling the display of iterations. The default value is FALSE.

Details

See [dist](#) for details about model parameterization.

Value

Returns an object of class "MGLMfit". An object of class "MGLMfit" is a list containing at least the following components:

estimate	The vector of the distribution parameter estimates.
SE	The vector of standard errors of the estimates.
vcov	The variance-covariance matrix of the estimates.
logL	The loglikelihood value.
iter	The number of iterations used.
BIC	Bayesian information criterion.
AIC	Akaike information criterion.
distribution	The distribution fitted.
LRT	When <code>dist="DM"</code> or <code>"GDM"</code> , it is the likelihood ratio test statistic for comparing the current model to the multinomial model. No LRT provided when <code>dist="NegMN"</code> .
LRTpvalue	The likelihood ratio test P value.
DoF	The degrees of freedom of the model.

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
data(rnaseq)
Y <- as.matrix(rnaseq[, 1:6])
fit <- MGLMfit(data=Y, dist="GDM")
```

MGLMfit-class

Class "MGLMfit"

Description

A list containing the model fitting results from the MGLMfit.

Objects from the Class

Objects can be created by calls of the form `new("MGLMfit", ...)`.

Slots

`estimate`: Object of class "vector", containing the parameter estimates.

`SE`: Object of class "vector", containing the standard errors of the estimates.

`vcov`: Object of class "matrix", the variance covariance matrix of the parameter estimates.

`logL`: Object of class "numeric", the fitted log likelihood.

`BIC`: Object of class "numeric", Bayesian information criterion.

`AIC`: Object of class "numeric", Akaike information criterion.

`LRTpvalue`: Object of class "numeric", likelihood ratio test p value.

`gradient`: Object of class "numeric", containing the gradient.

`iter`: Object of class "numeric", number of iteration used.

`distribution`: Object of class "character", the distribution fitted.

`fitted`: Object of class "vector", the fitted mean of each category.

Methods

print signature(`x = "MGLMfit"`): print out the results from the class MGLMfit.

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
showClass("MGLMfit")
```

MGLMreg

*Fit multivariate response GLM regression***Description**

MGLMreg fits multivariate response generalized linear models, specified by a symbolic description of the linear predictor and a description of the error distribution.

Usage

```
MGLMreg(formula, data, dist, init, weight, epsilon=1e-8,
maxiters=150, display=FALSE, LRT=FALSE, parallel=FALSE, cores, regBeta=FALSE)
```

```
DMD.MN.reg(Y, init, X, weight, epsilon, maxiters, display, parallel, cores, cl, sys)
DMD.DM.reg(Y, init, X, weight, epsilon, maxiters, display, parallel, cores, cl, sys)
DMD.GDM.reg(Y, init, X, weight, epsilon, maxiters, display, parallel, cores, cl, sys)
DMD.NegMN.reg(Y, init, X, weight, epsilon, maxiters, display, parallel, cores, cl, sys)
DMD.NegMN.Alpha.reg(
Y, init, X, weight, epsilon, maxiters, display, parallel, cores, cl, sys)
```

Arguments

formula	An object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The response has to be on the left hand side of ~.
data	An optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data when using function MGLMreg, the variables are taken from <code>environment(formula)</code> , typically the environment from which MGLMreg is called.
Y, X	For <code>DMD.MN.reg</code> , <code>DMD.DM.reg</code> , <code>DMD.GDM.reg</code> , and <code>DMD.NegMN.reg</code> , X is a design matrix of dimension $n \times (p+1)$ and Y is the response matrix of dimension $n \times d$.
dist	A description of the error distribution to fit. See dist for details.
weight	An optional vector of weights assigned to each row of the data. Should be NULL or a numeric vector. Could be a variable from data, or a variable from <code>environment(formula)</code> with the length equal to the number of rows of the data. If <code>weight=NULL</code> , equal weights of ones will be assigned.
init	An optional matrix of initial value of the parameter estimates. Should have the compatible dimension with data. See dist for details of the dimensions in each distribution.
epsilon	An optional numeric controlling the stopping criterion. The algorithm terminates when the relative change in the loglikelihoods of two successive iterates is less than epsilon. The default value is <code>epsilon=1e-8</code> .
maxiters	An optional numeric controlling the maximum number of iterations. The default value is <code>maxiters=150</code> .

display	An optional logical variable controlling the display of iterations. The default value is display=FALSE.
LRT	An optional logical variable controlling whether to perform likelihood ratio test on each predictor. The default value is LRT=FALSE, in which case only the Wald test is performed.
parallel	An optional logical variable controlling whether to perform parallel computing. On a multi-core Windows machine, a cluster is created based on socket; on a multi-core Linux/Mac machine, a cluster is created based on forking. The default value is parallel=FALSE.
cores	An optional value specifying the number of cores to use. Default value is half of the logical cores.
cl	A cluster object, created by the package parallel or by package snow . If parallel=TRUE, use the registered default cluster; if parallel=FALSE, any given value to cl will be ignored.
sys	The operating system. Will be used when choosing parallel type.
regBeta	An optional logical variable. When dist="NegMN", the user can decide whether to run regression on the over dispersion parameter β . The default is regBeta=FALSE.

Details

The formula should be in the form responses ~ covariates where the responses are the multivariate count matrix or a few columns from a data frame which is specified by data. The covariates are either matrices or from the data frame. The covariates can be numeric or character or factor.

See [dist](#) for details about distributions.

Value

Returns an object of class "MGLMreg". An object of class "MGLMreg" is a list containing at least the following components:

coefficients	The estimated regression coefficients.
SE	The standard errors of the estimates.
Hessian	The Hessian at the estimated parameter values.
gradient	The gradient at the estimated parameter values.
wald.value	The Wald statistics.
wald.p	The p values of Wald test.
test	Test statistic and the corresponding p-value. If LRT=FALSE, only returns test results from Wald test; if LRT=TRUE, returns the test results from both Wald test and likelihood ratio test.
logL	The final loglikelihood.
BIC	Bayesian information criterion.
AIC	Akaike information criterion.
fitted	The fitted values from the regression model
iter	The number of iterations used.

call The matched call.
distribution The distribution fitted.
data The data used to fit the model.

Author(s)

Yiwen Zhang and Hua Zhou

See Also

See also [MGLMfit](#) for distribution fitting.

Examples

```
##-----##
## Generate data
n <- 2000
p <- 5
d <- 4
m <- rep(20, n)
set.seed(1234)
X <- 0.1* matrix(rnorm(n*p),n, p)
alpha <- matrix(1, p, d-1)
beta <- matrix(1, p, d-1)
Alpha <- exp(X%%alpha)
Beta <- exp(X%%beta)
gdm.Y <- rgdirm(n, m, Alpha, Beta)

##-----##
## Regression
gdm.reg <- MGLMreg(gdm.Y~X, dist="GDM", LRT=FALSE)
```

MGLMreg-class

Class "MGLMreg"

Description

Objects can be created by calls of the form `new("MGLMreg", ...)`.

Objects from the Class

Objects can be created by calls of the form `new("MGLMreg", ...)`.

Slots

call: Object of class "function".
data: Object of class "list", consists of both the predictor matrix and the response matrix.
coefficients: Object of class "matrix", the estimated parameters.
SE: Object of class "matrix", the standard errors of the parameters.
test: Object of class "matrix", the test statistics and p-values.
Hessian: Object of class "matrix", the Hessian matrix.
logL: Object of class "numeric", the loglikelihood.
BIC: Object of class "numeric", Bayesian information criterion.
AIC: Object of class "numeric", Akaike information criterion.
iter: Object of class "numeric", the number of iterations used.
LRT: Object of class "numeric", likelihood ratio test statistics.
distribution: Object of class "character", the error distribution fitted.
fitted: Object of class "matrix", the fitted value.

Methods

print signature(x = "MGLMreg"): Print out regression results from the class MGLMreg.

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
showClass("MGLMreg")
```

MGLMsparsereg

Fit multivariate GLM sparse regression

Description

Fit sparse regression in multivariate generalized linear models.

Usage

```

MGLMsparsereg(formula, data, dist, lambda, penalty, weight,
  init, penidx, maxiters = 150, ridgedelta, epsilon = 1e-5, regBeta=FALSE, overdisp)
MGLMsparsereg.fit(Y, X, dist, lambda, penalty, weight,
  init, penidx, maxiters = 150, ridgedelta, epsilon = 1e-5, regBeta=FALSE, overdisp)

```


Arguments

formula	An object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The response has to be on the left hand side of ~.
data	An optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data when using function <code>MGLMsparsereg</code> , the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>MGLMsparsereg</code> is called.
Y	A matrix containing the multivariate categorical response data. Rows of the matrix represent observations, while columns are the different categories. Rows and columns of all zeros are automatically removed when <code>dist="MN"</code> , <code>"DM"</code> , or <code>"GDM"</code> .
X	Design matrix (including intercept). Number of rows of the matrix should match that of Y.
dist	A description of the error distribution to fit. See dist for details.
weight	An optional vector of weights assigned to each row of the data. Should be NULL or a numeric vector. Could be a variable from data, or a variable from <code>environment(formula)</code> with the length equal to the number of rows of the data. If <code>weight=NULL</code> , equal weights of ones will be assigned.
init	An optional matrix of initial value of the parameter estimates. Should have the compatible dimension with the data. See dist for details of the dimensions in each distribution.
lambda	Penalty parameter.
penalty	Penalty type for the regularization term. Can be chosen from "sweep", "group", or "nuclear". See Details for the description of each penalty type.
penidx	A logical vector indicating the variables to be penalized. The default value is <code>rep(TRUE, p)</code> , which means all predictors are subject to regularization. If X contains intercept, use <code>penidx=c(FALSE, rep(TRUE, p-1))</code> .
maxiters	An optional numeric controlling the maximum number of iterations. The default value is <code>maxiters=150</code> .
ridgedelta	An optional numeric controlling the behavior of the Nesterov's accelerated proximal gradient method. The default value is $\frac{1}{pd}$.
epsilon	An optional numeric controlling the stopping criterion. The algorithm terminates when the relative change in the objective values of two successive iterates is less than epsilon. The default value is <code>epsilon=1e-5</code> .
regBeta	An optional logical variable used when running negative multinomial regression (<code>dist="NegMN"</code>). <code>regBeta</code> controls whether to run regression on the overdispersion parameter. The default is <code>regBeta=FALSE</code> .
overdisp	An optional numerical variable used only when fitting sparse negative multinomial model <code>dist="NegMN"</code> and <code>regBeta=FALSE</code> . <code>overdisp</code> gives the overdispersion value for all the observations. The default value is estimated using negative-multinomial regression. When <code>dist="MN"</code> , <code>"DM"</code> , <code>"GDM"</code> or <code>regBeta=TRUE</code> , the value of <code>overdisp</code> is ignored.

Details

In general, we consider regularization problem

$$\min_B h(B) = -l(B) + J(B),$$

where $l(B)$ is the loglikelihood function and $J(B)$ is the regularization function.

Sparsity in the individual elements of the parameter matrix B is achieved by the lasso penalty (dist="sweep")

$$J(B) = \lambda \sum_{k \in \text{penidx}} \sum_{j=1}^d \|B_{kj}\|$$

Sparsity in the rows of the regression parameter matrix B is achieved by the group penalty (dist="group")

$$J(B) = \lambda \sum_{k \in \text{penidx}} \|B_{k\cdot}\|_2,$$

where $\|v\|_2$ is the l_2 norm of a vector v . In other words, $\|B_{k\cdot}\|_2$ is the l_2 norm of the k -th row of the parameter matrix B .

Sparsity in the rank of the parameter matrix B is achieved by the nuclear norm penalty (dist="nuclear")

$$J(B) = \lambda \|B\|_* = \lambda \sum_{i=1}^{\min(p,d)} \sigma_i(B),$$

where $\sigma_i(B)$ are the singular values of the parameter matrix B . The nuclear norm $\|B\|_*$ is a convex relaxation of $\text{rank}(B) = \|\sigma(B)\|_0$.

See [dist](#) for details about distributions.

Value

Returns an object of class "MGLMsparsereg". An object of class "MGLMsparsereg" is a list containing at least the following components:

coefficients	the estimated matrix of regression coefficients.
logL	The final loglikelihood value.
AIC	Akaike information criterion.
BIC	Bayesian information criterion.
Dof	Degrees of freedom of the estimated parameter.
iter	Number of iterations used.
maxlambda	The maximum tuning parameter such that the estimated coefficients are not all zero. This value is returned only when the tuning parameter lambda given to the function is large enough such that all the parameter estimates are zero; otherwise, maxlambda is not computed.
call	A matched call.
data	The data used to fit the model: a list of the predictor matrix and the response matrix.
penalty	The penalty chosen when running the penalized regression.

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```

## Not run:
##-----##
## Generate Dirichlet Multinomial data
dist <- "DM"
n <- 100
p <- 15
d <- 5
m <- runif(n, min=0, max=25) + 25
set.seed(134)
X <- matrix(rnorm(n*p),n, p)
alpha <- matrix(0, p, d)
alpha[c(1,3, 5), ] <- 1
Alpha <- exp(X%%alpha)
Y <- rdirn(size=m, alpha=Alpha)
##-----##

## Tuning
ngridpt <- 10
p <- ncol(X)
d <- ncol(Y)

pen <- 'nuclear'
fit <- MGLMsparsereg(formula=Y~0+X, dist=dist, lambda=Inf, penalty=pen)
maxlambda <- fit$maxlambda
lambdas <- exp(seq(from=log(maxlambda), to=log(maxlambda/100),
length.out=ngridpt))
BICs <- rep(0, ngridpt)
AICs <- rep(0, ngridpt)
LogLs <- rep(0, ngridpt)
Dofs <- rep(0, ngridpt)

ptm <- proc.time()
for(j in 1:ngridpt){
print(j)
if(j ==1 ) B0 <- matrix(0, p, ncol(fit$coefficients)) else B0 <- B_hat
select.fit <- MGLMsparsereg(formula=Y~0+X, dist=dist,lambda=lambdas[j],
penalty=pen, init=B0)
B_hat <- select.fit$coefficients
BICs[j] <- select.fit$BIC
LogLs[j] <- select.fit$logL
AICs[j] <- select.fit$AIC
Dofs[j] <- select.fit$Dof
}
proc.time() - ptm

chosen.lambda <- lambdas[which.min(BICs)]

```

```
select <- MGLMsparsereg(formula=Y~0+X, dist=dist, lambda=chosen.lambda,
penalty=pen)

## End(Not run)
```

MGLMsparsereg-class *Class "MGLMsparsereg"*

Description

A list containing the results from the MGLMsparsereg.

Objects from the Class

Objects can be created by calls of the form `new("MGLMsparsereg", ...)`.

Slots

`call`: Object of class "function".

`data`: Object of class "list", consists of both the predictor matrix and the response matrix

`coefficients`: Object of class "matrix", the estimated regression coefficients.

`logL`: Object of class "numeric", the loglikelihood.

`BIC`: Object of class "numeric", Bayesian information criterion.

`AIC`: Object of class "numeric", Akaike information criterion.

`Beta`: Object of class "numeric", the over dispersion parameter of the negative multinomial regression.

`Dof`: Object of class "numeric", the degrees of freedom.

`iter`: Object of class "numeric", number of iterations used.

`maxlambda`: Object of class "numeric", the maximum tuning parameter that ensures the estimated regression coefficients are not all zero.

`lambda`: Object of class "numeric", the tuning parameter used.

`distribution`: Object of class "character", the distribution fitted.

`penalty`: Object of class "character", the chosen penalty when running penalized regression.

Methods

print `signature(x = "MGLMsparsereg")`: print out sparse regression results from class "MGLMsparsereg".

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
showClass("MGLMsparsereg")
```

MGLMtune *Choose the tuning parameter value in sparse regression.*

Description

Finds the tuning parameter value that yields the smallest BIC.

Usage

```
MGLMtune(formula, data, dist, penalty, lambdas, ngridpt, warm.start=TRUE,
  keep.path=FALSE, display=FALSE, init, weight, penidx, ridgedelta,
  maxiters=150, epsilon=1e-5, regBeta=FALSE, overdisp)
```

Arguments

formula	An object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The response has to be on the left hand side of ~.
data	An optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data when using function <code>MGLMtune</code> , the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>MGLMtune</code> is called.
dist	A description of the distribution to fit. See dist for the details.
penalty	penalty type for the regularization term. Can be chosen from "sweep", "group", or "nuclear". See MGLMsparereg for the description of each penalty type.
lambdas	An optional vector of the penalty values to tune. If missing, the vector of penalty values will be set inside the function. <code>ngridpt</code> must be provided if <code>lambdas</code> is missing.
ngridpt	An optional numeric variable specifying the number of grid points to tune. If <code>lambdas</code> is given, <code>ngridpt</code> will be ignored. Otherwise, the maximum λ is determined from the data. The smallest λ is set to $1/n$, where n is the sample size.
warm.start	An optional logical variable to specify whether to give warm start at each tuning grid point. If <code>warm.start=TRUE</code> , the fitted sparse regression coefficients will be used as the initial value when fitting the sparse regression with the next tuning grid.
keep.path	An optional logical variable controlling whether to output the whole solution path. The default is <code>keep.path=FALSE</code> . If <code>keep.path=TRUE</code> , the sparse regression result at each grid point will be kept, and saved in the output object <code>select.list</code> .
display	An optional logical variable to specify whether to show each tuning step.
weight	An optional vector of weights assigned to each row of the data. Should be <code>NULL</code> or a numeric vector. Could be a variable from the data, or a variable from <code>environment(formula)</code> with the length equal to the number of rows of the data. If <code>weight=NULL</code> , equal weights of ones will be assigned.

<code>init</code>	An optional matrix of initial value of the parameter estimates. Should have the compatible dimension with the data. See <code>dist</code> for details of dimensions in each distribution.
<code>penidx</code>	A logical vector indicating the variables to be penalized. The default value is <code>rep(TRUE, p)</code> , which means all predictors are subject to regularization. If <code>X</code> contains intercept, use <code>penidx=c(FALSE, rep(TRUE, p-1))</code> .
<code>maxiters</code>	An optional numeric controlling the maximum number of iterations. The default value is <code>maxiters=150</code> .
<code>ridgedelta</code>	An optional numeric controlling the behavior of the Nesterov's accelerated proximal gradient method. The default value is $\frac{1}{pd}$.
<code>epsilon</code>	An optional numeric controlling the stopping criterion. The algorithm terminates when the relative change in the objective values of two successive iterates is less than <code>epsilon</code> . The default value is <code>epsilon=1e-5</code> .
<code>regBeta</code>	An optional logical variable used when running negative multinomial regression (<code>dist="NegMN"</code>). <code>regBeta</code> controls whether to run regression on the over-dispersion parameter. The default is <code>regBeta=FALSE</code> .
<code>overdisp</code>	An optional numerical variable used only when fitting sparse negative multinomial model and <code>regBeta=FALSE</code> . <code>overdisp</code> gives the over-dispersion value for all the observations. The default value is estimated using negative-multinomial regression. When <code>dist="MN"</code> , <code>"DM"</code> , <code>"GDM"</code> or <code>regBeta=TRUE</code> , the value of <code>overdisp</code> is ignored.

Value

<code>select</code>	The final sparse regression result, using the optimal tuning parameter.
<code>path</code>	A data frame with degrees of freedom and BICs at each lambda.

Author(s)

Yiwen Zhang and Hua Zhou

See Also

[MGLMsparsereg](#)

Examples

```
## Not run:
set.seed(345)
n <- 200
d <- 4
p <- 30
beta <- matrix(0, p, d-1)
beta[c(1,3,5), ] <- 1
beta <- matrix(0, p, d-1)
beta[c(1,3,5), ] <- 1
X <- matrix(rnorm(n*p),n, p)
Alpha <- exp(X*%alpha)
```

```
Beta <- exp(X%*%beta)
m <- rbinom(n, 80, 0.5)
Y <- rgdirm(m, Alpha, Beta)

select <- MGLMtune(Y~0+X, dist="GDM", penalty="group", ngridpt=30, display=TRUE)

## End(Not run)
```

MGLMtune-class	Class "MGLMtune"
----------------	------------------

Description

A list containing the results from the MGLMsparsereg.

Objects from the Class

Objects can be created by calls of the form `new("MGLMsparsereg", ...)`.

Slots

`call`: Object of class "function".

`select`: Object of class "MGLMsparsereg", regularized regression results given by the optimal tuning parameter.

`path`: Object of class "data.frame", the BIC, AIC, log-likelihood and degrees of freedom given each tuning parameter.

`select.list`: Object of class "list", the regularized regression results at each tuning grid point.

Methods

print signature(x = "MGLMtune"): Print out sparse regression solution path from class "MGLMtune".

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
showClass("MGLMtune")
```

predict-methods	<i>Predict using the fitted model</i>
-----------------	---------------------------------------

Description

Predict using the fitted model from MGLMreg when given a new set of covariates.

Methods

signature(object = "ANY") The new covariates data matrix.

signature(object = "MGLMreg") The model object.

print-methods	<i>Print out the defined objects</i>
---------------	--------------------------------------

Description

Print out the objects of class "MGLMfit", "MGLMreg", "MGLMsparse", "MGLMtune" in a nice way.

Methods

signature(x = "ANY") Print main function output.

signature(x = "MGLMfit") Print main function output for the fitted distribution.

signature(x = "MGLMreg") Print main function output for regression.

signature(x = "MGLMsparse") Print main function output for sparse regression.

signature(x = "MGLMtune") Print main function output for sparse regression tuning process.

rdirm	<i>Generate random observations from the Dirichlet multinomial distribution</i>
-------	---

Description

Generate random observations from the Dirichlet multinomial distribution.

Usage

```
rdirm(n, size, alpha)
```


Arguments

n	Number of random vectors to generate. When size is a scalar and alpha is a vector, must specify n. When size is a vector and alpha is a matrix, n is optional. The default value of n is the length of size. If given, n should be equal to the length of size.
size	A number or vector specifying the total number of objects that are put into d categories in the Dirichlet multinomial distribution.
alpha	Numerical positive vector or matrix. If alpha is a vector, size must be a scalar, and all the random vectors will be drawn from the same alpha and size. If alpha is a matrix, the number of rows should match the length of size, and each random vector will be drawn from the corresponding row of alpha and the corresponding element in the size vector. See Details below.

Details

The random vector is generated from the Dirichlet multinomial distribution with probability mass function

$$P(Y_1 = y_1, \dots, Y_d = y_d) = \binom{m}{y_1, \dots, y_d} \prod_{j=1}^d \frac{\Gamma(\alpha_j + y_j)}{\Gamma(\alpha_j)} \frac{\Gamma(\sum_{j'=1}^d \alpha_{j'})}{\Gamma(\sum_{j'=1}^d \alpha_{j'} + \sum_{j'=1}^d y_{j'})}$$

where $m = \sum_{j=1}^d y_j$.

Value

a n*d matrix of the generated random observations.

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
set.seed(100)
alpha <- matrix(abs(rnorm(50)), 10, 5)
size <- rbinom(10, 10, 0.5)
DM.rdm <- rdirnm(size=size, alpha=alpha)
DM.rdm1 <- rdirnm(n=20, size=20, alpha=c(0.2, 0.1, 0.5))
```

 rgdirm

Generate random observations from the generalized Dirichlet multinomial distribution

Description

Generate random observations from the generalized Dirichlet multinomial distribution.

Usage

```
rgdirm(n, size, alpha, beta)
```

Arguments

- n** Number of random vectors to generate. When size is a scalar and alpha is a vector, must specify n. When size is a vector and alpha is a matrix, n is optional. The default value of n is the length of size. If given, n should be equal to the length of size.
- size** A number or vector specifying the total number of objects that are put into d categories in the generalized Dirichlet multinomial distribution.
- alpha** Numerical positive vector or matrix. If alpha is a vector, size must be a scalar. All the random vectors will be drawn from the same alpha and size. If alpha is a matrix, the number of rows should match the length of size. Each random vector will be drawn from the corresponding row of alpha and the corresponding element of size.
- beta** Numerical positive vector or matrix. If beta is a vector, size must be a scalar. All the random samples will be drawn from the same beta and size. If beta is a matrix, the number of rows should match the length of size. Each random vector will be drawn from the corresponding row of beta and the corresponding element of size. See Details below.

Details

The random vectors are generated from the generalized Dirichlet multinomial distribution with probability mass function

$$P(Y_1 = y_1, \dots, Y_d = y_d) = \binom{m}{y_1, \dots, y_d} \prod_{j=1}^{d-1} \frac{\Gamma(\alpha_j + y_j)}{\Gamma(\alpha_j)} \frac{\Gamma(\beta_j + z_{j+1})}{\Gamma(\beta_j)} \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j + \beta_j + z_j)},$$

where $z_j = \sum_{k=j}^d y_k$ and $m = \sum_{j=1}^d y_j$.

Value

A n*d matrix of the generated random observations.

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
set.seed(100)
alpha <- matrix(abs(rnorm(40)), 10, 4)
beta <- matrix(abs(rnorm(40)), 10, 4)
size <- rbinom(10, 10, 0.5)
GDM.rdm <- rgdirm(size=size, alpha=alpha, beta=beta)
GDM.rdm1 <- rgdirm(n=20, size=10, alpha=abs(rnorm(4)), beta=abs(rnorm(4)))
```

rnaseq	<i>RNA-seq count data</i>
--------	---------------------------

Description

RNA-seq data simulated following the standard procedures (provided by Dr. Wei Sun, wsun@bios.unc.edu).

Usage

```
rnaseq
```

Format

A data frame containing 10 columns and 100 rows. The first 6 columns are the expression counts of 6 exons of a gene; the last four columns are the covariates: age, gender, treatment, and total number of reads.

Source

Dr. Sun Wei <wsun@bios.unc.edu>

rnegmn	<i>Generate random observations from the negative multinomial distribution</i>
--------	--

Description

Generate random observations from the negative multinomial distribution.

Usage

```
rnegmn(n, prob, beta)
```

Arguments

n	Number of random vectors to generate. When beta is a scalar and prob is a vector, must specify n. When beta is a vector and prob is a matrix, n is optional. The default value of n is the length of beta. If given, n should be equal to the length of beta.
prob	Numerical non-negative vector or matrix. If prob is a vector, beta must be a scalar. All the n random vectors will be drawn from the same prob and beta. If prob is a matrix, the number of rows should match the length of beta. Each random vector will be drawn from the corresponding row of prob and the corresponding element of beta. Each row of prob should have sum less than 1.
beta	Number or vector specifying the over dispersion in negative multinomial distribution.

Details

The random vectors are generated from the negative multinomial distribution $\text{NegMN}(p_1, p_2, \dots, p_d, \beta)$ with probability mass function

$$P(Y_1 = y_1, \dots, Y_d = y_d) = \binom{\beta + m - 1}{m} \binom{m}{y_1, \dots, y_d} \prod_{j=1}^d p_j^{y_j} p_{d+1}^\beta$$

where $p_{d+1} = 1 - \sum_{j=1}^d p_j$ and $m = \sum_{j=1}^d y_j$.

Value

A $n \times d$ matrix of the generated random observations.

Author(s)

Yiwen Zhang and Hua Zhou

Examples

```
set.seed(128)
n <- 100
d <- 4
p <- 5
a <- -matrix(1,p,d)
X <- matrix(runif(n*p), n, p )
alpha <- exp(X%*%a)
prob <- alpha/(rowSums(alpha)+1)
beta <- exp(X%*%matrix(1,p))
Y <- rnegmn(n, prob, beta)
```

Index

- *Topic **Distribution fitting**
 - dist, [5](#)
 - MGLMfit, [10](#)
 - *Topic **Models**
 - dist, [5](#)
 - MGLMfit, [10](#)
 - MGLMreg, [13](#)
 - *Topic **Random data generator**
 - rdirm, [24](#)
 - rgdirm, [25](#)
 - rnegmn, [27](#)
 - *Topic **Regression**
 - dist, [5](#)
 - MGLMreg, [13](#)
 - *Topic **Sparse regression**
 - MGLMsparsereg, [16](#)
 - MGLMtune, [21](#)
 - *Topic **classes**
 - MGLMfit-class, [12](#)
 - MGLMreg-class, [15](#)
 - MGLMsparsereg-class, [20](#)
 - MGLMtune-class, [23](#)
 - *Topic **dataset**
 - rnaseq, [27](#)
 - *Topic **log pdf**
 - ddirm, [3](#)
 - dgdirm, [4](#)
 - dmn, [7](#)
 - dnegmn, [8](#)
 - *Topic **methods**
 - predict-methods, [24](#)
 - print-methods, [24](#)
- ddirm, [3](#), [6](#)
dgdirm, [4](#), [6](#)
dist, [5](#), [10](#), [11](#), [13](#), [14](#), [17](#), [18](#), [21](#), [22](#)
DMD.DM.fit (MGLMfit), [10](#)
DMD.DM.reg (MGLMreg), [13](#)
DMD.GDM.fit (MGLMfit), [10](#)
DMD.GDM.reg (MGLMreg), [13](#)
- DMD.MN.reg (MGLMreg), [13](#)
DMD.NegMN.Alpha.reg (MGLMreg), [13](#)
DMD.NegMN.fit (MGLMfit), [10](#)
DMD.NegMN.reg (MGLMreg), [13](#)
dmn, [6](#), [7](#)
dnegmn, [6](#), [8](#)
- kr, [9](#)
- MGLM (MGLM-package), [2](#)
MGLM-package, [2](#)
MGLMfit, [6](#), [10](#), [15](#)
MGLMfit-class, [12](#)
MGLMreg, [5](#), [6](#), [13](#)
MGLMreg-class, [15](#)
MGLMsparsereg, [5](#), [6](#), [16](#), [21](#), [22](#)
MGLMsparsereg-class, [20](#)
MGLMtune, [21](#)
MGLMtune-class, [23](#)
- predict, ANY-method (predict-methods), [24](#)
predict, MGLMreg-method
(predict-methods), [24](#)
predict-methods, [24](#)
print, MGLMtune-method
(MGLMtune-class), [23](#)
print, ANY-method (print-methods), [24](#)
print, MGLMfit-method (print-methods), [24](#)
print, MGLMreg-method (print-methods), [24](#)
print, MGLMsparsereg-method
(print-methods), [24](#)
print, MGLMtune-method (print-methods),
[24](#)
print-methods, [24](#)
- rdirm, [24](#)
rgdirm, [25](#)
rnaseq, [27](#)
rnegmn, [27](#)