Package 'mixpoissonreg'

October 13, 2022

Title Mixed Poisson Regression for Overdispersed Count Data

Version 1.0.0

Maintainer Alexandre B. Simas <alexandre.impa@gmail.com>

Description Fits mixed Poisson regression models (Poisson-Inverse Gaussian or Negative-Binomial) on data sets with response variables being count data. The models can have varying precision parameter, where a linear regression structure (through a link function) is assumed to hold on the precision parameter. The Expectation-Maximization algorithm for both these models (Poisson Inverse Gaussian and Negative Binomial) is an important contribution of this package. Another important feature of this package is the set of functions to perform global and local influence analysis. See Barreto-Souza and Simas (2016) <doi:10.1007/s11222-015-9601-6> for further details.

License GPL-2

Encoding UTF-8

LazyData true

URL https://github.com/vpnsctl/mixpoissonreg/,

https://vpnsctl.github.io/mixpoissonreg/

BugReports https://github.com/vpnsctl/mixpoissonreg/issues

Imports pbapply, Formula, Rfast, dplyr, gamlss.dist, generics, ggplot2, gridExtra, lmtest, magrittr, statmod, tibble, rlang, ggrepel, gamlss

RoxygenNote 7.1.1

NeedsCompilation no

Suggests rmarkdown, knitr, devtools, bbreg, testthat, covr, tidyr, ggfortify, broom

VignetteBuilder knitr

Author Alexandre B. Simas [aut, cre] (<https://orcid.org/0000-0003-2562-2829>), Wagner Barreto-Souza [aut] (<https://orcid.org/0000-0003-0831-7881>)

Repository CRAN

Date/Publication 2021-03-10 19:50:06 UTC

R **topics documented:**

Attendance   *Attendance Records data set*

Description

School administrators study the attendance behavior of high school juniors at two schools.

Usage

```
data("Attendance")
```

Format

Data frame containing 314 observations on 4 variables.

daysabs number of days absent.

gender gender of the student.

prog three-level factor indicating the type of instructional program in which the student is enrolled.

math standardized math score.

Details

School administrators study the attendance behavior of high school juniors at two schools. Predictors of the number of days of absence include the type of program in which the student is enrolled and a standardized test in math. Attendance data on 314 high school juniors from two urban high schools. The response variable of interest is days absent, daysabs. The variable math is the standardized math score for each student. The variable prog is a three-level factor indicating the type of instructional program in which the student is enrolled.

Source

Data can be obtained from Introduction to Statistical Modeling Github Repository. See also *Barreto-Souza and Simas (2020)* for further details.

References

Hughes, M. and Fisher, T. (2020) Introduction to Statistical Modeling.

Examples

```
data("Attendance", package = "mixpoissonreg")

daysabs_fit <- mixpoissonregML(daysabs ~ gender + math + prog | gender +
math + prog, data = Attendance)
summary(daysabs_fit)
```

---

augment.mixpoissonreg   *Augment data with information from a* mixpoissonreg *object*

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. It includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column, if the type of prediction is 'link'. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Usage

```
## S3 method for class 'mixpoissonreg'
augment(
  x,
  data = stats::model.frame(x),
  newdata = NULL,
  type.predict = c("response", "link", "precision", "variance"),
  type.residuals = c("pearson", "score"),
  se_fit = FALSE,
  conf_int = TRUE,
```

```
    pred_int = FALSE,
    ...
)
```

Arguments

x	A `mixpoissonreg` object.
data	A [`base::data.frame`](base::data.frame) or [`tibble::tibble()`](tibble::tibble()) containing the original data that was used to produce the object x.
newdata	A [`base::data.frame`](base::data.frame) or [`tibble::tibble()`](tibble::tibble()) containing all the original predictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.
type.predict	Type of prediction. The options are 'response', 'link', 'precision' and 'variance'. The default is "response".
type.residuals	Type of residuals. The options are 'pearson' and 'score'. The default is 'pearson'.
se_fit	Logical indicating whether or not a .se.fit column should be added to the augmented output. If TRUE, it only returns a non-NA value if type of prediction is 'link'.
conf_int	Logical indicating whether or not confidence intervals for the fitted variable with type chosen from type.predict should be built. The available type options are 'response' and 'link'.
pred_int	Logical indicating whether or not prediction intervals for future observations should be built. It only works with type.predict = 'response'. The arguments `level`, `nsim_pred`, `nsim_pred_y` are passed through additional arguments '...'. Notice that this can be computationally intensive.
...	Additional arguments. Possible additional arguments are `level`, `nsim_pred`, `nsim_pred_y`, that are passed to `predict.mixpoissonreg` function.

Value

A [`tibble::tibble()`](tibble::tibble()) with columns:

- `.cooksd` Cook's distance.
- `.fitted` Fitted or predicted value.
- `.fittedlwrconf` Lower bound of the confidence interval, if conf_int = TRUE
- `.fitteduprconf` Upper bound of the confidence interval, if conf_int = TRUE
- `.fittedlwrpred` Lower bound of the prediction interval, if pred_int = TRUE
- `.fitteduprpred` Upper bound of the prediction interval, if pred_int = TRUE
- `.hat` Diagonal of the hat matrix.
- `.resid` The chosen residual.
- `.resfit` The chosen residual of the fitted object.
- `.se.fit` Standard errors of fitted values, if se_fit = TRUE.
- `.gencooksd` Generalized Cook's distance.

- .lwrenv Lower bound of the simulated envelope, if the fitted `mixpoissonreg` object, was fitted with envelopes > 0.
- .mdnenv Median of the simulated envelope, if the fitted `mixpoissonreg` object, was fitted with envelopes > 0.
- .uprenv Upper bound of the simulated envelope, if the fitted `mixpoissonreg` object, was fitted with envelopes > 0.

See Also

[glance.mixpoissonreg](), [tidy.mixpoissonreg](), [tidy_local_influence.mixpoissonreg](), [autoplot.mixpoissonreg](), [local_influence_autoplot.mixpoissonreg]()

---

autoplot.mixpoissonreg

Autoplot Method for mixpoissonreg *Objects*

Description

This function provides **ggplot2**-based counterparts to the plots produced by [plot.mixpoissonreg](). Currently there are six plots available. They contain residual analysis and global influence diagnostics. The plots are selectable by the `which` argument. The plots are: Residuals vs. obs. numbers; Normal Q-Q plots, which may contain simulated envelopes, if the fitted object has simulated envelopes; Cook's distances vs. obs. numbers; Generalized Cook's distances vs. obs. numbers; Cook's distances vs. Generalized Cook's distances; Response variables vs. fitted means. By default, the first two plots and the last two plots are provided.

If both ncol and nrow are NULL, the plots will be placed one at a time. To place multiple plots, set the values for `nrow` or `ncol`.

Usage

```
## S3 method for class 'mixpoissonreg'
autoplot(
  object,
  which = c(1, 2, 5, 6),
  title = list("Residuals vs Obs. number", "Normal Q-Q", "Cook's distance",
    "Generalized Cook's distance", "Cook's dist vs Generalized Cook's dist",
    "Response vs Fitted means"),
  title.bold = FALSE,
  title.size = NULL,
  title.colour = NULL,
  label.repel = TRUE,
  x.axis.col = NULL,
  y.axis.col = NULL,
  x.axis.size = NULL,
  y.axis.size = NULL,
  cook.plot.type = "linerange",
```

```
    cook.plot.pointshape = NULL,
    nrow = NULL,
    ncol = NULL,
    qqline = TRUE,
   ask = prod(graphics::par("mfcol")) < length(which) && grDevices::dev.interactive(),
    include.modeltype = TRUE,
    include.residualtype = FALSE,
    sub.caption = NULL,
    sub.caption.col = NULL,
    sub.caption.size = NULL,
    sub.caption.face = NULL,
    sub.caption.hjust = 0.5,
    env_alpha = 0.5,
    env_fill = "grey70",
    gpar_sub.caption = list(fontface = "bold"),
    colour = "#444444",
    size = NULL,
    linetype = NULL,
    alpha = NULL,
    fill = NULL,
    shape = NULL,
    label = TRUE,
    label.label = NULL,
    label.colour = "#000000",
    label.alpha = NULL,
    label.size = NULL,
    label.angle = NULL,
    label.family = NULL,
    label.fontface = NULL,
    label.lineheight = NULL,
    label.hjust = NULL,
    label.vjust = NULL,
    label.n = 3,
    ad.colour = "#888888",
    ad.linetype = "dashed",
    ad.size = 0.2,
    ...
  )
```

Arguments

object	A `mixpoissonreg` object.
which	a list or vector indicating which plots should be displayed. If a subset of the plots is required, specify a subset of the numbers 1:6, see title below for the different kinds. In plot number 2, 'Normal Q-Q', if the `mixpoissonreg` object was fitted with envelopes, a quantile-quantile plot with simulated envelopes will be displayed.
title	titles to appear above the plots; character vector or list of valid graphics annota-

	tions. Can be set to "" to suppress all titles.
`title.bold`	logical indicating whether the titles should be bold. The default is FALSE.
`title.size`	numerical indicating the size of the titles.
`title.colour`	title colour.
`label.repel`	Logical flag indicating whether to use **ggrepel** to place the labels.
`x.axis.col`	colour of the x axis title.
`y.axis.col`	colour of the y axis title.
`x.axis.size`	size of the x axis title.
`y.axis.size`	size of the y axis title.
`cook.plot.type`	character indicating the type of plot for Cook's distance and generalized Cook's distance. Default is "linerange". The options are "linerange" and "points".
`cook.plot.pointshape`	
	the shape of points if "cook.plot.type" is set to "points".
`nrow`	Number of facet/subplot rows. If both `nrow` and `ncol` are NULL, the plots will be placed one at a time. For multiple plots, set values for `nrow` or `ncol`.
`ncol`	Number of facet/subplot columns. If both `nrow` and `ncol` are NULL, the plots will be placed one at a time. For multiple plots, set values for `nrow` or `ncol`.
`qqline`	logical; if TRUE and the fit does *not* contain simulated envelopes, a qqline passing through the first and third quartiles of a standard normal distribution will be added to the normal Q-Q plot.
`ask`	logical; if TRUE, the user is asked before each plot.
`include.modeltype`	
	logical. Indicates whether the model type ('NB' or 'PIG') should be displayed on the titles.
`include.residualtype`	
	local. Indicates whether the name of the residual ('Pearson' or 'Score') should be displayed on the title of plot 1 (Residuals vs. Index).
`sub.caption`	common title-above the figures if there are more than one. If NULL, as by default, a possible abbreviated version of `deparse(x$call)` is used.
`sub.caption.col`	
	color of subcaption (when one figure at a time).
`sub.caption.size`	
	size of subcaption (when one figure at a time).
`sub.caption.face`	
	font face for subcaption, options are: "plain", "bold", "italic" and "bold.italic".
`sub.caption.hjust`	
	indicates the position of the subcaption (when one figure at a time). The default is 0.5, which indicates that the subcaption is centered, a value 0 places the subcaption at the left side of the plot whereas a value of 1 places the subcaption at the right side of the plot.
`env_alpha`	alpha of the envelope region (when the fitted model has envelopes)
`env_fill`	the colour of the filling in the envelopes.

gpar_sub.caption

        list of gpar parameters to be used as common title in the case of multiple plots. The title will be given in sub.caption argument. See the help of [gpar](#) function from the **grid** package for all the available options.

colour	line colour.
size	point size.
linetype	line type.
alpha	alpha of the plot.
fill	fill colour.
shape	point shape.
label	Logical value whether to display labels.
label.label	vector of labels. If NULL, rownames will be used as labels.
label.colour	Colour for text labels.
label.alpha	Alpha for text labels.
label.size	Size for text labels.
label.angle	Angle for text labels.
label.family	Font family for text labels.
label.fontface	Fontface for text labels.

label.lineheight

        Lineheight for text labels.

label.hjust	Horizontal adjustment for text labels.
label.vjust	Vertical adjustment for text labels.
label.n	Number of points to be laeled in each plot, starting with the most extreme.
ad.colour	Line colour for additional lines.
ad.linetype	Line type for additional lines.
ad.size	Fill colour for additional lines.
...	other arguments passed to methods.

Details

Based on `autoplot.lm` from the excellent **ggfortify** package, [ggfortify](#).

sub.caption - by default the function call - is shown as a subtitle (under the x-axis title) on each plot when plots are on separate pages, or as a subtitle in the outer margin when there are multiple plots per page.

Value

Called for its side effects.

Examples

```
daysabs_prog <- mixpoissonregML(daysabs ~ prog, data = Attendance)
autoplot(daysabs_prog)
```

---

coef.mixpoissonreg *Coef Method for* mixpoissonreg *Objects.*

Description

Extract model coefficients of fitted mixed Poisson regression models. The parameters arguments allows one to chose if all coefficients should be extracted, with parameters = 'all'; if the coefficients of the mean-related parameters should be extracted, with parameters = 'mean'; if the coefficients of the precision-related parameters should be extracted, with parameters = 'precision'.

Usage

```
## S3 method for class 'mixpoissonreg'
coef(object, parameters = c("all", "mean", "precision"), ...)
```

Arguments

object	object of class "mixpoissonreg" containing results from the fitted model.
parameters	a string to determine which coefficients should be extracted: 'all' extracts all coefficients, 'mean' extracts the coefficients of the mean parameters and 'precision' extracts coefficients of the precision parameters.
...	further arguments passed to or from other methods.

Value

A vector containing the coefficients of a *mixpoissonreg* object.

See Also

[vcov.mixpoissonreg](#)

Examples

```
daysabs_prog <- mixpoissonreg(daysabs ~ prog, data = Attendance)
coef(daysabs_prog)
```

---

fitted.mixpoissonreg   *Fitted Method for* mixpoissonreg *Objects*

Description

Function providing the fitted means, linear predictors, precisions or variances for mixed Poisson regression models.

Usage

```
## S3 method for class 'mixpoissonreg'
fitted(object, type = c("response", "link", "precision", "variance"), ...)
```

Arguments

object	object of class "mixpoissonreg" containing results from the fitted model.
type	the type of variable to get the fitted values. The default is the "response" type, which provided the estimated values for the means. The type "link" provides the estimates for the linear predictor of the mean. The type "precision" provides estimates for the precision parameters whereas the type "variance" provides estimates for the variances.
...	Currently not used.

Value

A vector containing the fitted values of a *mixpoissonreg* object.

See Also

[predict.mixpoissonreg](), [summary.mixpoissonreg](), [coef.mixpoissonreg](), [vcov.mixpoissonreg](),
[plot.mixpoissonreg]()

Examples

```
daysabs_prog <- mixpoissonreg(daysabs ~ prog, data = Attendance)
fitted(daysabs_prog)
```

glance.mixpoissonreg *Glance at a* mixpoissonreg *object*

Description

Glance accepts a `mixpoissonreg` object and returns a [`tibble::tibble()`](tibble::tibble()) with exactly one row of model summaries. The summaries are Efron's pseudo-R^2, degrees of freedom, AIC, BIC, log-likelihood, the type of model used in the fit ('NB' or 'PIG'), the total number of observations and the estimation method.

Usage

```
## S3 method for class 'mixpoissonreg'
glance(x, ...)
```

Arguments

x   A `mixpoissonreg` object.

...   Additional arguments. Currently not used.

Value

A [`tibble::tibble()`](tibble::tibble()) with exactly one row and columns:

- `efron.pseudo.r2` Efron's pseudo-R^2, that is, the squared correlation between the fitted values and the response values.

- `df.null` Degrees of freedom used by the null model.

- `logLik` The log-likelihood of the model.

- `AIC` Akaike's Information Criterion for the model.

- `BIC` Bayesian Information Criterion for the model.

- `df.residual` Residual degrees of freedom.

- `nobs` Number of observations used.

- `model.type` Type of model fitted, "NB" or "PIG".

- `est.method` The estimation method of the fitted model, "EM" or "ML".

See Also

augment.mixpoissonreg, tidy.mixpoissonreg, tidy_local_influence.mixpoissonreg, autoplot.mixpoissonreg, local_influence_autoplot.mixpoissonreg

---

influence.mixpoissonreg

Global Influence Diagnostics for Mixed Poisson Regression Models

---

Description

These functions provides global influence diagnostic quantities such as Cook's distance, hat values, generalized Cook's distance (through argument on `cooks.distance.mixpoissonreg` function), likelihood displacement (through argument on `cooks.distance.mixpoissonreg` function) and Q-displacement (through argument on `cooks.distance.mixpoissonreg` function).

Usage

```
## S3 method for class 'mixpoissonreg'
hatvalues(model, parameters = c("mean", "precision"), ...)

## S3 method for class 'mixpoissonreg'
cooks.distance(
  model,
  type = c("CD", "GCD", "GCDmean", "GCDprecision", "LD", "QD"),
  hat = c("mean", "precision"),
  ...
)

## S3 method for class 'mixpoissonreg'
influence(model, do.coef = TRUE, ...)
```

Arguments

model	a `mixpoissonreg` object.
parameters	the parameter to which the hat values will be computed. The options are 'mean' and 'precision'. The default is 'mean'. For hatvalues with respect to the mean the model must be fitted with 'x' set to TRUE, and for hatvalues with respect to the precision the model must be fitted with 'w' set to TRUE.
...	Currently not used.
type	the type of Cook's distance to be used. The options are 'CD', the standard Cook's distance; 'GCD', the generalized Cook's distance with respect to all parameters, Zhu et al. (2001); 'GCDmean', the generalized Cook's distance with respect to the mean parameters; 'GCDprecision', the generalized Cook's distance with respect to the precision parameters; 'LD', the likelihood displacement (also known as likelihood distance), see Cook and Weisberg (1982); 'QD', the Q-displacement, see Zhu et al. (2001). See 'details' for more informations. For 'GCD', 'GCDmean', 'LD' and 'QD', the model must be fitted with 'x' set to TRUE, and for 'GCD', 'GCDprecision', 'LD' and 'QD', the model must be fitted with 'w' set to TRUE. For 'CD', if 'hat' is set to 'mean', the model must be fitted with 'x' set to TRUE, whereas if 'hat' is set to 'precision', the model must be fitted with 'w' set to TRUE.

hat	hat values H[i,i]. The default is obtained through the second-derivative of the Q-function in the spirit of Zhu et al. (2001) and Pregibon (1981), see details.
do.coef	logical indicating if the the approximation to the change of coefficients values after case removal are desired. The model must be fitted with x = TRUE. See details for further explanations.

Details

For hat values of mixed Poisson regression models, we follow Zhu et al. (2001) to consider the negative of the hessian of the Q-function as weight matrix, and follow Pregibon (1981) to define the 'hat' matrix with respect to this weight matrix. We can consider the hessian of the Q-function with respect to mean-related parameters, which is the default. We can also consider the hessian of the Q-function with respect to the precision-related parameters to give rise to hat values related to the precision parameters.

The Generalized Cook's distance and Q-displacement for EM-based models were defined in Zhu et al. (2001) and computed for mixed Poisson regression models in Barreto-Souza and Simas (2016). We implemented first-order approximation to these quantities to make it computationally feasible. These first-order approximations are available in Barreto-Souza and Simas (2016). We also provide versions of generalized Cook's distance for mean-related or precision-related parameters, whose details can be found in Barreto-Souza and Simas (2016).

In the influence method we provide a 'do.coef' argument that computes first-order approximations to the impact of removal of each case to each parameter, in the same spirit as the 'do.coeff' argument in 'influence.lm'.

Value

The method influence.mixpoissonreg returns a list containing *hat.mean*, a vector containing the hat values with respect to the mean, *hat.precision*, a vector containing the hat values with respect to the precision, *coefficients.mean* (if do.coef=TRUE), a matrix containing the first order approximations for the mean-related coefficients after the removal of each observation, *coefficients.precision* (if do.coef=TRUE), a matrix containing the first order approximations for the precision-related coefficients after the removal of each observation, *pear.res*, a vector containing the Pearson residuals, *score.res*, a vector containing the score residuals.

The *cooks.distance.mixpoissonreg* returns a vector containing the Cook's distances.

The *hatvalues.mixpoissonreg* returns a vector containing the hat values.

References

DOI:10.1007/s11222-015-9601-6 doi: [10.1007/s1122201596016](Barreto-Souza and Simas, 2016)

Cook, D.R. and Weisberg, S. (1982) *Residuals and Influence in Regression*. (New York: Chapman and Hall, 1982)

Pregibon, D. (1981) *Logistic Regression Diagnostics*. Ann. Stat. 9, 705-724.

Zhu, H.T., Lee, S.Y., Wei, B.C., Zhu, J. (2001) *Case-deletion measures formodels with incomplete data*. Biometrika, 88, 727–737.

---

local_influence	*Local Influence Diagnostics*

Description

Local Influence Diagnostics

Usage

```
local_influence(model, ...)

local_influence_plot(model, ...)
```

Arguments

model	an object for which the local influence is desired
...	further arguments passed to or from other methods.

Details

`local_influence` is a generic function to return local influence diagnostics under different perturbation schemes and different directions. `local_influence_plot` is a generic function to provide friendly plots of such diagnostics.

Local influence diagnostics were first introduced by Cook (1986), where several perturbation schemes were introduced and normal curvatures were obtained. Poon and Poon (1999) introduced the conformal normal curvature, which has nice properties and takes values on the unit interval $[0, 1]$. Zhu and Lee (2001) following Cook (1986) and Poon and Poon (1999) introduced normal and conformal normal curvatures for EM-based models.

Value

The *local_influence* method returns a list containing the resulting perturbation schemes as elements. The *local_influence_plot* is called for its side effects.

References

Cook, R. D. (1986) *Assessment of Local Influence.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 48, pp.133-169. https://rss.onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1986.tb01398.x

Poon, W.-Y. and Poon, Y.S. (1999) *Conformal normal curvature and assessment of local influence.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 61, pp.51-61. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00162

Zhu, H.-T. and Lee, S.-Y. (2001) *Local influence for incomplete data models.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 63, pp.111-126. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00279

See Also

local_influence.mixpoissonreg, local_influence_plot.mixpoissonreg, local_influence_autoplot.mixpoisson

---

local_influence.mixpoissonreg

Local Influence Diagnostics for Mixed Poisson Regression Models

---

Description

This function provides local influence diagnostic quantities. Currently the conformal normal and normal curvatures are available under several perturbation schemes. The default is the conformal normal curvature since it takes values on $[0, 1]$ and other nice properties (see Zhu and Lee, 2001 and Poon and Poon, 1999 for further details).

Usage

```
## S3 method for class 'mixpoissonreg'
local_influence(
  model,
  perturbation = c("case_weights", "hidden_variable", "mean_explanatory",
    "precision_explanatory", "simultaneous_explanatory"),
  curvature = c("conformal", "normal"),
  direction = c("canonical", "max.eigen"),
  parameters = c("all", "mean", "precision"),
  mean.covariates = NULL,
  precision.covariates = NULL,
  ...
)
```

Arguments

model	a mixpoissonreg object.
perturbation	a list or vector of perturbation schemes to be returned. The currently available schemes are "case_weights", "hidden_variable", "mean_explanatory", "precision_explanatory", "simultaneous_explanatory". See Barreto-Souza and Simas (2016) for further details.
curvature	the curvature to be returned, 'conformal' for the conformal normal curvature (see Zhu and Lee, 2001 and Poon and Poon, 1999) or 'normal' (see Zhu and Lee, 2001 and Cook, 1986).
direction	the 'max.eigen' returns the eigenvector associated to the largest eigenvalue of the perturbation matrix. The 'canonical' considers the curvatures under the canonical directions, which is known as "total local curvature" (see Lesaffre and Verbeke, 1998). For conformal normal curvatures both of them coincide. The default is 'canonical'.

parameters   the parameter to which the local influence will be computed. The options are
                    'all', 'mean' and 'precision'. This argument affects the 'case_weights' and 'hid-
                    den_variable' perturbation schemes. The default is 'all'.

mean.covariates

                    a list or vector of characters containing the mean-explanatory variables to be
                    used in the 'mean-explanatory' and 'simultaneous-explanatory' perturbation schemes.
                    If NULL, the 'mean-explanatory' and 'simultaneous-explanatory' perturbation
                    schemes will be computed by perturbing all mean-related covariates. The de-
                    fault is NULL.

precision.covariates

                    a list or vector of characters containing the precision-explanatory variables to be
                    used in the 'precision-explanatory' and 'simultaneous-explanatory' perturbation
                    schemes. If NULL, the 'precision-explanatory' and 'simultaneous-explanatory'
                    perturbation schemes will be computed by perturbing all precision-related co-
                    variates. The default is NULL.

...   other graphical arguments to be passed.

Details

local_influence.mixpoissonreg provides local influence diagnostics for mixed Poisson regres-
sion models for all perturbation schemes considered in Barreto-Souza and Simas (2016), for normal
and conformal normal curvatures. Further, it is also provides results for the canonical directions,
which is called the total local influence (see Lesaffre and Verbeke, 1998), as well as for the direction
of largest curvature, which is the direction of the eigenvector of the perturbation matrix associated
to the largest eigenvalue.

local_influence_plot.mixpoissonreg provides a plot of the local influence diagnostics. Each
plot corresponds to a perturbation scheme. The first plot considers the 'case-weights' perturbation;
the second plot considers the 'hidden-variable' perturbation (which was introduced in Barreto-
Souza and Simas, 2016); the third plot considers the mean-explanatory perturbation; the fourth
plot considers the precision-explanatory perturbation; the fifth plot considers the simultanous-
explanatory perturbation.

For both local_influence.mixpoissonreg and local_influence_plot.mixpoissonreg, one
can select which covariates will be perturbed in the 'mean-explanatory', 'precision-explanatory'
and 'simultaneous-explanatory' perturbation schemes. These are chosen in the 'mean.covariates'
and 'precision.covariates' arguments.

If one considers the total local influence, then Zhu and Lee (2001) provides benchmark for influen-
tial observations for all perturbation schemes. These are returned as attributes in the returned list
from local_influence.mixpoissonreg. When using the local_influence_plot.mixpoissonreg,
only points above the benchmark will be displayed. One can also set the option 'draw_benchmark'
to TRUE to plot the benchmark line.

Value

a list containing the resulting perturbation schemes as elements. Each returned element has an
attribute 'benchmark', which for the conformal normal curvature, it is computed following Zhu and
Lee (2001), and for normal curvature it is computed following Verbeke and ... If the 'direction' is
'max.eigen' the 'benchmark' attribute is NA.

The 'mean_explanatory', 'precision_explanatory' and 'simultaneous_explanatory' elements of the list contain an attribute 'covariates' indicating which covariates were used in the perturbation schemes.

References

DOI:10.1007/s11222-015-9601-6 doi: 10.1007/s1122201596016(Barreto-Souza and Simas; 2016)

Cook, R. D. (1986) *Assessment of Local Influence.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 48, pp.133-169. https://rss.onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1986.tb01398.x

Lesaffre, E. and Verbeke, G. (1998) *Local Influence in Linear Mixed Models.* Biometrics, 54, pp. 570-582.

Poon, W.-Y. and Poon, Y.S. (1999) *Conformal normal curvature and assessment of local influence.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 61, pp.51-61. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00162

Zhu, H.-T. and Lee, S.-Y. (2001) *Local influence for incomplete data models.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 63, pp.111-126. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00279

Examples

```
daysabs_progML <- mixpoissonregML(daysabs ~ prog | prog, data = Attendance)
local_influence(daysabs_progML)
```

---

local_influence_autoplot.mixpoissonreg

Local Influence Autoplots for mixpoissonreg *Objects*

Description

Function to provide customizable ggplot2-based plots of local influence diagnostics.

Usage

```
## S3 method for class 'mixpoissonreg'
local_influence_autoplot(
  model,
  which = c(1, 2, 3, 4),
  title = list("Case Weights Perturbation", "Hidden Variable Perturbation",
    "Mean Explanatory Perturbation", "Precision Explanatory Perturbation",
    "Simultaneous Explanatory Perturbation"),
  title.size = NULL,
  title.bold = FALSE,
  title.colour = NULL,
  x.axis.col = NULL,
  y.axis.col = NULL,
```

```
        x.axis.size = NULL,
        y.axis.size = NULL,
        type.plot = "linerange",
        curvature = c("conformal", "normal"),
        direction = c("canonical", "max.eigen"),
        parameters = c("all", "mean", "precision"),
        mean.covariates = NULL,
        precision.covariates = NULL,
        label.repel = TRUE,
        nrow = NULL,
        ncol = NULL,
       ask = prod(graphics::par("mfcol")) < length(which) && grDevices::dev.interactive(),
        include.modeltype = TRUE,
        sub.caption = NULL,
        sub.caption.col = NULL,
        sub.caption.size = NULL,
        sub.caption.face = NULL,
        sub.caption.hjust = 0.5,
        gpar_sub.caption = list(fontface = "bold"),
        detect.influential = TRUE,
        n.influential = 5,
        draw.benchmark = FALSE,
        colour = "#444444",
        size = NULL,
        linetype = NULL,
        alpha = NULL,
        fill = NULL,
        shape = NULL,
        label = TRUE,
        label.label = NULL,
        label.colour = "#000000",
        label.alpha = NULL,
        label.size = NULL,
        label.angle = NULL,
        label.family = NULL,
        label.fontface = NULL,
        label.lineheight = NULL,
        label.hjust = NULL,
        label.vjust = NULL,
        ad.colour = "#888888",
        ad.linetype = "dashed",
        ad.size = 0.2,
        ...
    )
```

Arguments

model   A mixpoissonreg model.

which	a list or vector indicating which plots should be displayed. If a subset of the plots is required, specify a subset of the numbers 1:5, see caption below (and the 'Details') for the different kinds.
title	titles to appear above the plots; character vector or list of valid graphics annotations. Can be set to "" to suppress all titles.
title.size	numerical indicating the size of the titles.
title.bold	logical indicating whether the titles should be bold. The default is FALSE.
title.colour	title colour.
x.axis.col	colour of the x axis title.
y.axis.col	colour of the y axis title.
x.axis.size	size of the x axis title.
y.axis.size	size of the y axis title.
type.plot	a character indicating the type of the plots. The default is "linerange". The options are "linerange" and "points".
curvature	the curvature to be returned, 'conformal' for the conformal normal curvature (see Zhu and Lee, 2001 and Poon and Poon, 1999) or 'normal' (see Zhu and Lee, 2001 and Cook, 1986).
direction	the 'max.eigen' returns the eigenvector associated to the largest eigenvalue of the perturbation matrix. The 'canonical' considers the curvatures under the canonical directions, which is known as "total local curvature" (see Lesaffre and Verbeke, 1998). For conformal normal curvatures both of them coincide. The default is 'canonical'.
parameters	the parameter to which the local influence will be computed. The options are 'all', 'mean' and 'precision'. This argument affects the 'case_weights' and 'hidden_variable' perturbation schemes. The default is 'all'.
mean.covariates	a list or vector of characters containing the mean-explanatory variables to be used in the 'mean-explanatory' and 'simultaneous-explanatory' perturbation schemes. If NULL, the 'mean-explanatory' and 'simultaneous-explanatory' perturbation schemes will be computed by perturbing all mean-related covariates. The default is NULL.
precision.covariates	a list or vector of characters containing the precision-explanatory variables to be used in the 'precision-explanatory' and 'simultaneous-explanatory' perturbation schemes. If NULL, the 'precision-explanatory' and 'simultaneous-explanatory' perturbation schemes will be computed by perturbing all precision-related covariates. The default is NULL.
label.repel	Logical flag indicating whether to use **ggrepel** to place the labels.
nrow	Number of facet/subplot rows. If both nrow and ncol are NULL, the plots will be placed one at a time. For multiple plots, set values for nrow or ncol.
ncol	Number of facet/subplot columns. If both nrow and ncol are NULL, the plots will be placed one at a time. For multiple plots, set values for nrow or ncol.
ask	logical; if TRUE, the user is asked before each plot.

include.modeltype

      logical. Indicates whether the model type ('NB' or 'PIG') should be displayed on the captions.

sub.caption   common title-above the figures if there are more than one. If NULL, as by default, a possible abbreviated version of deparse(x$call) is used.

sub.caption.col

      color of subcaption (when one figure at a time).

sub.caption.size

      size of subcaption (when one figure at a time).

sub.caption.face

      font face for subcaption, options are: "plain", "bold", "italic" and "bold.italic".

sub.caption.hjust

      indicates the position of the subcaption (when one figure at a time). The default is 0.5, which indicates that the subcaption is centered, a value 0 places the sub-caption at the left side of the plot whereas a value of 1 places the subcaption at the right side of the plot.

gpar_sub.caption

      list of gpar parameters to be used as common title in the case of multiple plots. The title will be given in sub.caption argument. See the help of [gpar](#) function from the **grid** package for all the available options.

detect.influential

      logical. Indicates whether the benchmark should be used to detect influential observations and identify them on the plot. If there is no benchmark available, the top 'n.influential' observations will be identified in the plot by their indexes.

n.influential   interger. The maximum number of influential observations to be identified on the plot.

draw.benchmark logical. Indicates whether a horizontal line identifying the benchmark should be drawn.

colour   line colour.

size   point size.

linetype   line type.

alpha   alpha of the plot.

fill   fill colour.

shape   point shape.

label   Logical value whether to display labels.

label.label   vector of labels. If NULL, rownames will be used as labels.

label.colour   Colour for text labels.

label.alpha   Alpha for text labels.

label.size   Size for text labels.

label.angle   Angle for text labels.

label.family   Font family for text labels.

label.fontface Fontface for text labels.

```
label.lineheight
                  Lineheight for text labels.
```

`label.hjust`	Horizontal adjustment for text labels.
`label.vjust`	Vertical adjustment for text labels.
`ad.colour`	Line colour for additional lines.
`ad.linetype`	Line type for additional lines.
`ad.size`	Fill colour for additional lines.
`...`	Currently not used.

Value

Called for its side effects.

References

DOI:10.1007/s11222-015-9601-6 doi: [10.1007/s1122201596016](10.1007/s1122201596016)(Barreto-Souza and Simas; 2016)

Cook, R. D. (1986) *Assessment of Local Influence.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 48, pp.133-169. https://rss.onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1986.tb01398.x

Lesaffre, E. and Verbeke, G. (1998) *Local Influence in Linear Mixed Models.* Biometrics, 54, pp. 570-582.

Poon, W.-Y. and Poon, Y.S. (2002) *Conformal normal curvature and assessment of local influence.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 61, pp.51-61. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00162

Zhu, H.-T. and Lee, S.-Y. (2001) *Local influence for incomplete data models.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 63, pp.111-126. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00279

See Also

glance.mixpoissonreg, augment.mixpoissonreg, tidy.mixpoissonreg, autoplot.mixpoissonreg

Examples

```
daysabs_prog <- mixpoissonreg(daysabs ~ prog | prog, data = Attendance)
local_influence_autoplot(daysabs_prog)
```

local_influence_plot.mixpoissonreg

Local Influence Plot Diagnostics for Mixed Poisson Regression Models

Description

Local influence plots for mixed Poisson regression models. Currently the conformal normal and normal curvatures are available under several perturbation schemes. The default is the conformal normal curvature since it takes values on $[0, 1]$ and other nice properties (see Zhu and Lee, 2001 and Poon and Poon, 1999 for further details).

Usage

```
## S3 method for class 'mixpoissonreg'
local_influence_plot(
  model,
  which = c(1, 2, 3, 4),
  caption = list("Case Weights Perturbation", "Hidden Variable Perturbation",
    "Mean Explanatory Perturbation", "Precision Explanatory Perturbation",
    "Simultaneous Explanatory Perturbation"),
  sub.caption = NULL,
  detect.influential = TRUE,
  n.influential = 5,
  draw.benchmark = FALSE,
  lty.benchmark = 2,
  type_plot = "h",
  curvature = c("conformal", "normal"),
  direction = c("canonical", "max.eigen"),
  parameters = c("all", "mean", "precision"),
  mean.covariates = NULL,
  precision.covariates = NULL,
  main = "",
 ask = prod(graphics::par("mfcol")) < length(which) && grDevices::dev.interactive(),
  labels.id = names(stats::residuals(model)),
  cex.id = 0.75,
  cex.oma.main = 1.25,
  cex.caption = 1,
  include.modeltype = TRUE,
  ...
)
```

Arguments

model   a mixpoissonreg object.

which	a list or vector indicating which plots should be displayed. If a subset of the plots is required, specify a subset of the numbers 1:5, see caption below (and the 'Details') for the different kinds.
caption	captions to appear above the plots; character vector or list of valid graphics annotations. Can be set to "" or NA to suppress all captions.
sub.caption	common title-above the figures if there are more than one. If NULL, as by default, a possible abbreviated version of deparse(x$call) is used.
detect.influential	
	logical. Indicates whether the benchmark should be used to detect influential observations and identify them on the plot. If there is no benchmark available, the top 'n.influential' observations will be identified in the plot by their indexes.
n.influential	interger. The maximum number of influential observations to be identified on the plot.
draw.benchmark	logical. Indicates whether a horizontal line identifying the benchmark should be drawn.
lty.benchmark	the line type of the benchmark if drawn.
type_plot	what type of plot should be drawn. The default is 'h'.
curvature	the curvature to be returned, 'conformal' for the conformal normal curvature (see Zhu and Lee, 2001 and Poon and Poon, 1999) or 'normal' (see Zhu and Lee, 2001 and Cook, 1986).
direction	the 'max.eigen' returns the eigenvector associated to the largest eigenvalue of the perturbation matrix. The 'canonical' considers the curvatures under the canonical directions, which is known as "total local curvature" (see Lesaffre and Verbeke, 1998). For conformal normal curvatures both of them coincide. The default is 'canonical'.
parameters	the parameter to which the local influence will be computed. The options are 'all', 'mean' and 'precision'. This argument affects the 'case_weights' and 'hidden_variable' perturbation schemes. The default is 'all'.
mean.covariates	
	a list or vector of characters containing the mean-explanatory variables to be used in the 'mean-explanatory' and 'simultaneous-explanatory' perturbation schemes. If NULL, the 'mean-explanatory' and 'simultaneous-explanatory' perturbation schemes will be computed by perturbing all mean-related covariates. The default is NULL.
precision.covariates	
	a list or vector of characters containing the precision-explanatory variables to be used in the 'precision-explanatory' and 'simultaneous-explanatory' perturbation schemes. If NULL, the 'precision-explanatory' and 'simultaneous-explanatory' perturbation schemes will be computed by perturbing all precision-related covariates. The default is NULL.
main	character; title to be placed at each plot additionally (and above) all captions.
ask	logical; if TRUE, the user is asked before each plot.
labels.id	vector of labels, from which the labels for extreme points will be chosen. The default uses the observation numbers.

cex.id   magnification of point labels.

cex.oma.main   controls the size of the sub.caption only if that is above the figures when there is
                  more than one.

cex.caption   controls the size of caption.

include.modeltype

                  logical. Indicates whether the model type ('NB' or 'PIG') should be displayed
                  on the captions.

...   other graphical arguments to be passed.

Details

`local_influence.mixpoissonreg` provides local influence diagnostics for mixed Poisson regres-
sion models for all perturbation schemes considered in Barreto-Souza and Simas (2016), for normal
and conformal normal curvatures. Further, it is also provides results for the canonical directions,
which is called the total local influence (see Lesaffre and Verbeke, 1998), as well as for the direction
of largest curvature, which is the direction of the eigenvector of the perturbation matrix associated
to the largest eigenvalue.

`local_influence_plot.mixpoissonreg` provides a plot of the local influence diagnostics. Each
plot corresponds to a perturbation scheme. The first plot considers the 'case-weights' perturbation;
the second plot considers the 'hidden-variable' perturbation (which was introduced in Barreto-
Souza and Simas, 2016); the third plot considers the mean-explanatory perturbation; the fourth
plot considers the precision-explanatory perturbation; the fifth plot considers the simultanous-
explanatory perturbation.

For both `local_influence.mixpoissonreg` and `local_influence_plot.mixpoissonreg`, one
can select which covariates will be perturbed in the 'mean-explanatory', 'precision-explanatory'
and 'simultaneous-explanatory' perturbation schemes. These are chosen in the 'mean.covariates'
and 'precision.covariates' arguments.

If one considers the total local influence, then Zhu and Lee (2001) provides benchmark for influen-
tial observations for all perturbation schemes. These are returned as attributes in the returned list
from `local_influence.mixpoissonreg`. When using the `local_influence_plot.mixpoissonreg`,
only points above the benchmark will be displayed. One can also set the option 'draw_benchmark'
to TRUE to plot the benchmark line.

Value

Called for its side effects.

References

DOI:10.1007/s11222-015-9601-6 doi: [10.1007/s1122201596016](https://doi.org/10.1007/s11222-015-9601-6)(Barreto-Souza and Simas; 2016)

Cook, R. D. (1986) *Assessment of Local Influence.* Journal of the Royal Statistical Society. Series
B (Methodological), Vol. 48, pp.133-169. https://rss.onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1986.tb01398.x

Lesaffre, E. and Verbeke, G. (1998) *Local Influence in Linear Mixed Models.* Biometrics, 54, pp.
570-582.

Poon, W.-Y. and Poon, Y.S. (1999) *Conformal normal curvature and assessment of local influence.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 61, pp.51-61. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00162

Zhu, H.-T. and Lee, S.-Y. (2001) *Local influence for incomplete data models.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 63, pp.111-126. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00279

Examples

```
daysabs_progML <- mixpoissonregML(daysabs ~ prog | prog, data = Attendance)
local_influence_plot(daysabs_progML)
```

---

logLik.mixpoissonreg   *logLik Method for* mixpoissonreg *Objects*

Description

Function to compute the log-likelihood at the estimated parameters for mixed Poisson regression models.

Usage

```
## S3 method for class 'mixpoissonreg'
logLik(object, ...)
```

Arguments

object	an object of class "mixpoissonreg" containing results from the fitted model.
...	further arguments passed to or from other methods.

Value

Returns an object of class LogLik containing the log-likelihood of the fitted *mixpoissonreg* object.

See Also

vcov.mixpoissonreg

Examples

```
daysabs_prog <- mixpoissonreg(daysabs ~ prog, data = Attendance)
logLik(daysabs_prog)
```

---

mixpoissonreg   *Mixed Poisson Regression for Overdispersed Count Data*

Description

Fits mixed Poisson regression models (Poisson-Inverse Gaussian or Negative-Binomial) on data
sets with response variables being count data. The models can have varying precision parameter,
where a linear regression structure (through a link function) is assumed to hold on the precision pa-
rameter. The Expectation-Maximization algorithm for both these models (Poisson Inverse Gaussian
and Negative Binomial) is an important contribution of this package. Another important feature of
this package is the set of functions to perform global and local influence analysis.

Usage

```
mixpoissonreg(
  formula,
  data,
  link.mean = c("log", "sqrt"),
  link.precision = c("identity", "log", "inverse.sqrt"),
  model = c("NB", "PIG"),
  method = c("EM", "ML"),
  residual = c("pearson", "score"),
  y = TRUE,
  x = TRUE,
  w = TRUE,
  envelope = 0,
  prob = 0.95,
  model.frame = TRUE,
  em_controls = list(maxit = 5000, em_tol = 10^(-5), em_tolgrad = 10^(-2)),
  optim_method = "L-BFGS-B",
  optim_controls = list()
)

mixpoissonreg.fit(
  x,
  y,
  w = NULL,
  link.mean = c("log", "sqrt"),
  link.precision = c("identity", "log", "inverse.sqrt"),
  model = c("NB", "PIG"),
  method = c("EM", "ML"),
  residual = c("pearson", "score"),
  envelope = 0,
  prob = 0.95,
  em_controls = list(maxit = 5000, em_tol = 10^(-5), em_tolgrad = 10^(-2)),
  optim_method = "L-BFGS-B",
  optim_controls = list()
```

)

Arguments

formula	symbolic description of the model (examples: y ~ x1 + ... + xnbeta and y ~ x1 + ... + xnbeta | w1 + ... + wnalpha); see details below.
data	elements expressed in formula. This is usually a data frame composed by: (i) the observations formed by count data z, with z_i being non-negative integers, (ii) covariates for the mean submodel (columns x1, ..., xnbeta) and (iii) covariates for the precision submodel (columns w1, ..., wnalphla).
link.mean	optionally, a string containing the link function for the mean. If omitted, the 'log' link function will be used. The possible link functions for the mean are "log" and "sqrt".
link.precision	optionally, a string containing the link function the precision parameter. If omitted and the only precision covariate is the intercept, the 'identity' link function will be used, if omitted and there is a precision covariate other than the intercept, the 'log' link function will be used. The possible link functions for the precision parameter are "identity" and "inverse.sqrt" (which is $\phi^{-1/2} = w_i^T alpha$).
model	character ("NB" or "PIG") indicating the type of model to be fitted, with "NB" standing for Negative-Binomial and "PIG" standing for Poisson Inverse Gaussian. The default is "NB".
method	estimation method to be chosen between "EM" (Expectation-Maximization) and "ML" (Maximum-Likelihood). The default method is "EM".
residual	character indicating the type of residual to be evaluated ("pearson" or "score"). The default is "pearson". Notice that they coincide for Negative-Binomial models.
y	For mixpoissonreg: logical values indicating if the response vector should be returned as component.
	For mixpoissonreg.fit: a numerical vector of response variables with length n. Each coordinate must be a nonnegative-integer.
x	For mixpoissonreg: logical values indicating if the model matrix x should be returned as component.
	For mixpoissonreg.fit: a matrix of covariates with respect to the mean with dimension (n,nbeta).
w	For mixpoissonreg: logical values indicating if the model matrix w should be returned as component.
	For mixpoissonreg.fit a matrix of covariates with respect to the precision parameter. The default is NULL. If not NULL must be of dimension (n,nalpha).
envelope	number of simulations (synthetic data sets) to build envelopes for residuals (with 100*prob% confidence level). The default envelope = 0 dismisses the envelope analysis.
prob	probability indicating the confidence level for the envelopes (default: prob = 0.95). If envelope = 0, prob is ignored.
model.frame	logical indicating whether the model frame should be returned as component of the returned value.

em_controls	only used with the 'EM' method. A list containing two elements: maxit that contains the maximum number of iterations of the EM algorithm, the default is set to 5000; em_tol that defines the tolerance value to control the convergence criterion in the EM-algorithm, the default is set to 10^(-5); em_tolgrad that defines the tolerance value of the maximum-norm of the the gradient of the Q-function, the default is set to 10^(-2).
optim_method	main optimization algorithm to be used. The available methods are the same as those of optim function. The default is set to "L-BFGS-B".
optim_controls	a list of control arguments to be passed to the optim function in the optimization of the model. For the control options, see the 'Details' in the help of optim for the possible arguments.

Details

Among the regression models with discrete response variables, Poisson regression is the most popular for modeling count data. See, for instance Sellers and Shmueli (2010). It is well-known that this model is equidispersed (that is, the mean is equal to the variance), which in practice may be an unrealistic assumption. Several models have been introduced in the literature to overcome this problem such as negative binomial (NB) and Poisson inverse gaussian (PIG) distributions (see Lawless, 1987). The most common way to do this is to consider a mixed Poisson distribution, which is defined as follows. Let Z be a positive random variable (generally being continuous) with distribution function $G_\tau(\cdot)$, where τ denotes the parameter vector associated to the G distribution. Let $Y|Z = z \sim$ Poisson(μz), for some constant $\mu > 0$. Therefore Y follows a mixed Poisson (MP) distribution with probability function given by

$$P(Y = y) = \int_0^\infty \frac{e^{-\mu z}(\mu z)^y}{y!} dG_\tau(z),$$

for $y = 0, 1, \ldots$. With this, Y has an overdispersed distribution and hence it is a natural alternative to the Poisson distribution. The most common choices for Z are gamma and inverse-gaussian distributions, which yields Y following, respectively, NB and PIG distributions. General properties of the MP distributions can be found in Karlis and Xekalaki (2005) and in the references therein.

In mixpoissonreg two regression models are implemented, namely, the NB and PIG regression models. We follow the definitions and notations given in Barreto-Souza and Simas (2016). The mixed Poisson regression model is defined by assuming $Y_1, \ldots, Y_n$ is a random sample where $Y_i \sim NB(\mu_i, \phi_i)$ or $Y_i \sim PIG(\mu_i, \phi_i)$ for $i = 1, \ldots, n$. Under this parameterization we have $E(Y_i) = \mu_i$ and $Var(Y_i) = \mu_i(1 + \mu_i \phi_i^{-1} b''(\xi_0))$, where $b(\theta) = -\log(-\theta)$ and $\xi_0 = -1$ for the NB case, and $b(\theta) = -(-2\theta)^{1/2}$ and $\xi_0 = -1/2$ for the PIG case, with $b''(\cdot)$ being the second derivative of the function $b(\cdot)$. The following linear relations are assumed

$$\Lambda_1(\mu_i) = x_i^T \beta$$

and

$$\Lambda_2(\phi_i) = w_i^T \alpha,$$

where $\beta = (\beta_1, ..., \beta_p)$ and $\alpha = (\alpha_1, ..., \alpha_q)$ are real valued vectors. The terms x_i^T and v_i^T represent, respectively, the i-th row of the matrices "x" ($n \times p$) and "w" ($n \times q$) containing covariates in their columns ($x_{i,1}$ and $v_{i,1}$ may be 1 to handle intercepts).

Therefore, the mixpoissonreg package handles up to two regression structures at the same time: one for the mean parameter, one for the precision parameter. The regression structure for the mean is

determined through a formula y ~ x1 + ... + xn, whereas the regression structure for the precision parameter is determined through the right-hand side of the formula using the separator "|". So, for example, a regression with x1,...,xn as covariates for the mean and z1,...,zm as covariates for the precision parameter corresponds to the formula y ~ x1 + ... + xn | z1 + ... + zm. If only there is only formula for the regression structure for the mean, the regression structure for the precision parameter will only have the intercept, that is, y ~ x1 + ... + xn is the same as y ~ x1 + ... + xn | 1.

In general, in this package, the EM-algorithm estimation method obtains estimates closer to the maximum likelihood estimate than the maximum likelihood estimation method, in the sense that the likelihood function evaluated at the EM-algorithm estimate is greater or equal (usually strictly greater) than the likelihood function evaluated at the maximum likelihood estimate. So, unless the processing time is an issue, we strongly recommend the EM-algorithm as the estimation method.

In Barreto-Souza and Simas (2016) two residuals were studied: the pearson residuals and the score residuals. Both these residuals are implemented in the mixpoissonreg package. They coincide for NB regression models. They can be accessed via the [residuals](#) method.

It is also noteworthy that all the global and local influence analysis tools developed in Barreto-Souza and Simas (2016) are implemented in this package. See [influence.mixpoissonreg](#), [local_influence.mixpoissonreg](#), [local_influence_plot.mixpoissonreg](#) and [local_influence_autoplot.mixpoissonreg](#).

Value

mixpoissonreg returns an object of class "mixpoissonreg" whereas mixpoissonreg.fit returns an object of class "mixpoissonreg_fit". Both objects are given by lists containing the outputs from the model fit (Negative-Binomial or Poisson Inverse Gaussian regression).

An object of the class "mixpoissonreg" is a list containing the following elements:

- coefficients - a list with elements "mean" and "precision" containing the estimated coefficients of the model;
- call - the formula used by the model. If using mixpoissonreg.fit, this returns NULL.
- modelname - the fitted model, NB or PIG;
- modeltype - the abbreviated model name
- residualname - the name of the chosen residual in the call, 'pearson' or 'score';
- niter - number of iterations of the EM algorithm if method = "EM" and number of iterations of the optim function, if method = "ML";
- start - the initial guesses of the parameters
- intercept - vector indicating if the intercept is present in the mean and/or in the precision regressions;
- link.mean - link function of the mean;
- link.precision - link function of the precision parameter;
- fitted.values - a vector of fitted values in the response scale;
- fitted.precisions - a vector of fitted precisions;
- efron.pseudo.r2 - Efron's pseudo R^2: the squared correlation between the response variables and the predicted values;
- vcov - covariance matrix of the parameters of the fitted model;

- `logLik` - log-likelihood at the estimated parameters;
- `Qfunction` - Q-function at the estimated parameters;
- `x` - the covariates related to the mean (if x = TRUE);
- `w` - the covariates related to the precision parameter (if w = TRUE);
- `y` - the response variables (if y = TRUE);
- `model` - if requested (the default), the model frame;
- `formula` - the formula supplied;
- `nobs` - number of observations
- `df.null` - the residual degrees of freedom for the model with constant mean and constant precision;
- `df.residual` - the residual degrees of freedom of the fitted model;
- `estimation_method` - the estimation method, "EM" or "ML"
- `residuals` - vector of raw residuals, that is, the response variable minus the fitted means;
- `std_errors` - the standard errors of the estimated parameters;
- `envelope` - the numerical envelopes used to build the Q-Q plot with simulated envelopes;
- `terms` - (only for `mixpoissonreg`)the `terms` object used;
- `levels` - (where relevant, only for `mixpoissonreg`) the levels of the factors used;
- `contrasts` - (where relevant, only for `mixpoissonreg`) the contrasts used.

References

DOI:10.1007/s11222-015-9601-6 doi: [10.1007/s1122201596016](#)(Barreto-Souza and Simas; 2016)

URL:https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1751-5823.2005.tb00250.x ([Karlis and Xekalaki; 2005](#))

DOI:10.2307/3314912 doi: [10.2307/3314912](#)(Lawless; 1987)

Sellers, K.F. and Shmueli, G. (2010) *A flexible regression model for count data.* Ann. Appl. Stat., 4, 943-961

See Also

[summary.mixpoissonreg](#), [plot.mixpoissonreg](#), [autoplot.mixpoissonreg](#), [residuals.mixpoissonreg](#), [predict.mixpoissonreg](#),[influence.mixpoissonreg](#), [cooks.distance.mixpoissonreg](#), [local_influence.mixpoisso](#) [local_influence_plot.mixpoissonreg](#), [local_influence_autoplot.mixpoissonreg](#)

Examples

```
# Examples using the Attendance dataset:

daysabs_prog <- mixpoissonreg(daysabs ~ prog, data = Attendance)
summary(daysabs_prog)
```

---

mixpoissonregML | *Maximum Likelihood Mixed Poisson Regression Models for Overdispersed Count Data*

Description

Uses maximum likelihood estimators to fit mixed Poisson regression models (Poisson-Inverse Gaussian or Negative-Binomial) on data sets with response variables being count data. The models can have varying precision parameter, where a linear regression structure (through a link function) is assumed to hold on the precision parameter.

Usage

```
mixpoissonregML(
  formula,
  data,
  link.mean = c("log", "sqrt"),
  link.precision = c("identity", "log", "inverse.sqrt"),
  model = c("NB", "PIG"),
  residual = c("pearson", "score"),
  y = TRUE,
  x = TRUE,
  w = TRUE,
  envelope = 0,
  prob = 0.95,
  model.frame = TRUE,
  em_controls = list(maxit = 5000, em_tol = 10^(-5), em_tolgrad = 10^(-2)),
  optim_method = "L-BFGS-B",
  optim_controls = list()
)

mixpoissonregML.fit(
  x,
  y,
  w = NULL,
  link.mean = c("log", "sqrt"),
  link.precision = c("identity", "log", "inverse.sqrt"),
  model = c("NB", "PIG"),
  residual = c("pearson", "score"),
  envelope = 0,
  prob = 0.95,
  em_controls = list(maxit = 5000, em_tol = 10^(-5), em_tolgrad = 10^(-2)),
  optim_method = "L-BFGS-B",
  optim_controls = list()
)
```

Arguments

formula	symbolic description of the model (examples: y ~ x1 + ... + xnbeta and y ~ x1 + ... + xnbeta | w1 + ... + wnalpha); see details below.
data	elements expressed in formula. This is usually a data frame composed by: (i) the observations formed by count data z, with z_i being non-negative integers, (ii) covariates for the mean submodel (columns x1, ..., xnbeta) and (iii) covariates for the precision submodel (columns w1, ..., wnalphla).
link.mean	optionally, a string containing the link function for the mean. If omitted, the 'log' link function will be used. The possible link functions for the mean are "log" and "sqrt".
link.precision	optionally, a string containing the link function the precision parameter. If omitted and the only precision covariate is the intercept, the 'identity' link function will be used, if omitted and there is a precision covariate other than the intercept, the 'log' link function will be used. The possible link functions for the precision parameter are "identity" and "inverse.sqrt" (which is $\phi^{-1/2} = w_i^T alpha$).
model	character ("NB" or "PIG") indicating the type of model to be fitted, with "NB" standing for Negative-Binomial and "PIG" standing for Poisson Inverse Gaussian. The default is "NB".
residual	character indicating the type of residual to be evaluated ("pearson" or "score"). The default is "pearson". Notice that they coincide for Negative-Binomial models.
y	For mixpoissonregML: logical values indicating if the response vector should be returned as component.
	For mixpoissonregML.fit: a numerical vector of response variables with length n. Each coordinate must be a nonnegative-integer.
x	For mixpoissonregML: logical values indicating if the model matrix x should be returned as component.
	For mixpoissonregML.fit: a matrix of covariates with respect to the mean with dimension (n,nbeta).
w	For mixpoissonregML: logical values indicating if the model matrix w should be returned as component.
	For mixpoissonregML.fit a matrix of covariates with respect to the precision parameter. The default is NULL. If not NULL must be of dimension (n,nalpha).
envelope	number of simulations (synthetic data sets) to build envelopes for residuals (with 100*prob% confidence level). The default envelope = 0 dismisses the envelope analysis.
prob	probability indicating the confidence level for the envelopes (default: prob = 0.95). If envelope = 0, prob is ignored.
model.frame	logical indicating whether the model frame should be returned as component of the returned value.
em_controls	only used with the 'EM' method. A list containing two elements: maxit that contains the maximum number of iterations of the EM algorithm, the default is set to 5000; em_tol that defines the tolerance value to control the convergence criterion in the EM-algorithm, the default is set to 10^(-5). em_tolgrad that

defines the tolerance value of the maximum-norm of the the gradient of the Q-function, the default is set to 10^(-2).

optim_method   main optimization algorithm to be used. The available methods are the same as those of optim function. The default is set to "L-BFGS-B".

optim_controls   a list of control arguments to be passed to the optim function in the optimization of the model. For the control options, see the 'Details' in the help of optim for the possible arguments.

Details

Among the regression models with discrete response variables, Poisson regression is the most popular for modeling count data. See, for instance Sellers and Shmueli (2010). It is well-known that this model is equidispersed (that is, the mean is equal to the variance), which in practice may be an unrealistic assumption. Several models have been introduced in the literature to overcome this problem such as negative binomial (NB) and Poisson inverse gaussian (PIG) distributions (see Lawless, 1987). The most common way to do this is to consider a mixed Poisson distribution, which is defined as follows. Let Z be a positive random variable (generally being continuous) with distribution function $G_\tau(\cdot)$, where τ denotes the parameter vector associated to the G distribution. Let $Y|Z = z \sim$Poisson(μz), for some constant $\mu > 0$. Therefore Y follows a mixed Poisson (MP) distribution with probability function given by

$$P(Y = y) = \int_0^\infty \frac{e^{-\mu z}(\mu z)^y}{y!} dG_\tau(z),$$

for $y = 0, 1, \ldots$. With this, Y has an overdispersed distribution and hence it is a natural alternative to the Poisson distribution. The most common choices for Z are gamma and inverse-gaussian distributions, which yields Y following, respectively, NB and PIG distributions. General properties of the MP distributions can be found in Karlis and Xekalaki (2005) and in the references therein.

In mixpoissonreg two regression models are implemented, namely, the NB and PIG regression models. We follow the definitions and notations given in Barreto-Souza and Simas (2016). The mixed Poisson regression model is defined by assuming $Y_1, \ldots, Y_n$ is a random sample where $Y_i \sim NB(\mu_i, \phi_i)$ or $Y_i \sim PIG(\mu_i, \phi_i)$ for $i = 1, \ldots, n$. Under this parameterization we have $E(Y_i) = \mu_i$ and $Var(Y_i) = \mu_i(1 + \mu_i\phi_i^{-1}b''(\xi_0))$, where $b(\theta) = -\log(-\theta)$ and $\xi_0 = -1$ for the NB case, and $b(\theta) = -(-2\theta)^{1/2}$ and $\xi_0 = -1/2$ for the PIG case, with $b''(\cdot)$ being the second derivative of the function $b(\cdot)$. The following linear relations are assumed

$$\Lambda_1(\mu_i) = x_i^T \beta$$

and

$$\Lambda_2(\phi_i) = w_i^T \alpha,$$

where $\beta = (\beta_1, ..., \beta_p)$ and $\alpha = (\alpha_1, ..., \alpha_q)$ are real valued vectors. The terms x_i^T and v_i^T represent, respectively, the i-th row of the matrices "x" ($n \times p$) and "w" ($n \times q$) containing covariates in their columns ($x_{i,1}$ and $v_{i,1}$ may be 1 to handle intercepts).

Therefore, the mixpoissonreg package handles up to two regression structures at the same time: one for the mean parameter, one for the precision parameter. The regression structure for the mean is determined through a formula y ~ x1 + ... + xn, whereas the regression structure for the precision parameter is determined through the right-hand side of the formula using the separator "|". So, for example, a regression with x1,...,xn as covariates for the mean and z1,...,zm as covariates for

the precision parameter corresponds to the formula y ~ x1 + ... + xn | z1 + ... + zm. If only there is only formula for the regression structure for the mean, the regression structure for the precision parameter will only have the intercept, that is, y ~ x1 + ... + xn is the same as y ~ x1 + ... + xn | 1.

In general, in this package, the EM-algorithm estimation method obtains estimates closer to the maximum likelihood estimate than the maximum likelihood estimation method, in the sense that the likelihood function evaluated at the EM-algorithm estimate is greater or equal (usually strictly greater) than the likelihood function evaluated at the maximum likelihood estimate. So, unless the processing time is an issue, we strongly recommend the EM-algorithm as the estimation method.

In Barreto-Souza and Simas (2016) two residuals were studied: the pearson residuals and the score residuals. Both these residuals are implemented in the mixpoissonreg package. They coincide for NB regression models. They can be accessed via the residuals method.

It is also noteworthy that all the global and local influence analysis tools developed in Barreto-Souza and Simas (2016) are implemented in this package. See influence.mixpoissonreg, local_influence.mixpoissonreg, local_influence_plot.mixpoissonreg and local_influence_autoplot.mixpoissonreg.

Value

mixpoissonregML returns an object of class "mixpoissonreg" whereas mixpoissonregML.fit returns an object of class "mixpoissonreg_fit". Both objects are given by lists containing the outputs from the model fit (Negative-Binomial or Poisson Inverse Gaussian regression).

An object of the class "mixpoissonreg" is a list containing the following elements:

- coefficients - a list with elements "mean" and "precision" containing the estimated coefficients of the model;
- call - the formula used by the model. If using mixpoissonreg.fit, this returns NULL.
- modelname - the fitted model, NB or PIG;
- modeltype - the abbreviated model name
- residualname - the name of the chosen residual in the call, 'pearson' or 'score';
- niter - number of iterations of the EM algorithm if method = "EM" and number of iterations of the optim function, if method = "ML";
- start - the initial guesses of the parameters
- intercept - vector indicating if the intercept is present in the mean and/or in the precision regressions;
- link.mean - link function of the mean;
- link.precision - link function of the precision parameter;
- fitted.values - a vector of fitted values in the response scale;
- fitted.precisions - a vector of fitted precisions;
- efron.pseudo.r2 - Efron's pseudo R^2: the squared correlation between the response variables and the predicted values;
- vcov - covariance matrix of the parameters of the fitted model;
- logLik - log-likelihood at the estimated parameters;
- Qfunction - Q-function at the estimated parameters;

- x - the covariates related to the mean (if x = TRUE);

- w - the covariates related to the precision parameter (if w = TRUE);

- y - the response variables (if y = TRUE);

- model - if requested (the default), the model frame;

- formula - the formula supplied;

- nobs - number of observations

- df.null - the residual degrees of freedom for the model with constant mean and constant precision;

- df.residual - the residual degrees of freedom of the fitted model;

- estimation_method - the estimation method, "EM" or "ML"

- residuals - vector of raw residuals, that is, the response variable minus the fitted means;

- std_errors - the standard errors of the estimated parameters;

- envelope - the numerical envelopes used to build the Q-Q plot with simulated envelopes;

- terms - (only for mixpoissonreg)the terms object used;

- levels - (where relevant, only for mixpoissonreg) the levels of the factors used;

- contrasts - (where relevant, only for mixpoissonreg) the contrasts used.

References

DOI:10.1007/s11222-015-9601-6 doi: 10.1007/s1122201596016(Barreto-Souza and Simas; 2016)

URL:https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1751-5823.2005.tb00250.x (Karlis and Xekalaki; 2005)

DOI:10.2307/3314912 doi: 10.2307/3314912(Lawless; 1987)

Sellers, K.F. and Shmueli, G. (2010) *A flexible regression model for count data.* Ann. Appl. Stat., 4, 943-961

See Also

summary.mixpoissonreg, plot.mixpoissonreg, autoplot.mixpoissonreg, residuals.mixpoissonreg, predict.mixpoissonreg,influence.mixpoissonreg, cooks.distance.mixpoissonreg, local_influence.mixpoisso local_influence_plot.mixpoissonreg, local_influence_autoplot.mixpoissonreg

Examples

```
# Examples using the Attendance dataset:

daysabs_progML <- mixpoissonregML(daysabs ~ prog, data = Attendance)
summary(daysabs_progML)
```

plot.mixpoissonreg   *Plot Diagnostics for* mixpoissonreg *Objects*

Description

Currently there are six plots available. They contain residual analysis and global influence diagnostics. The plots are selectable by the which argument. The plots are: Residuals vs. obs. numbers; Normal Q-Q plots, which may contain simulated envelopes, if the fitted object has simulated envelopes; Cook's distances vs. obs. numbers; Generalized Cook's distances vs. obs. numbers; Cook's distances vs. Generalized Cook's distances; Response variables vs. fitted means. By default, the first two plots and the last two plots are provided.

Usage

```
## S3 method for class 'mixpoissonreg'
plot(
  x,
  which = c(1, 2, 5, 6),
  caption = list("Residuals vs Obs. number", "Normal Q-Q", "Cook's distance",
    "Generalized Cook's distance", "Cook's dist vs Generalized Cook's dist",
    "Response vs Fitted means"),
  sub.caption = NULL,
  qqline = TRUE,
  col.qqline = "black",
  main = "",
  line_col_env = "gray",
  line_col_median = "black",
  fill_col_env = "gray",
  fill_alpha_env = 0.7,
 ask = prod(graphics::par("mfcol")) < length(which) && grDevices::dev.interactive(),
  labels.id = names(stats::residuals(x)),
  label.pos = c(4, 2),
  type.cookplot = "h",
  id.n = 3,
  col.id = NULL,
  cex.id = 0.75,
  cex.oma.main = 1.25,
  cex.caption = 1,
  col.caption = "black",
  cex.points = 1,
  col.points = "black",
  include.modeltype = TRUE,
  include.residualtype = FALSE,
  ...
)
```

Arguments

x	object of class "mixpoissonreg" containing results from the fitted model. If the model was fitted with envelope = 0, the Q-Q plot will be produced without envelopes.
which	a list or vector indicating which plots should be displayed. If a subset of the plots is required, specify a subset of the numbers 1:6, see caption below for the different kinds. In plot number 2, 'Normal Q-Q', if the mixpoissonreg object was fitted with envelopes, a quantile-quantile plot with simulated envelopes will be displayed.
caption	captions to appear above the plots; character vector or list of valid graphics annotations. Can be set to "" or NA to suppress all captions.
sub.caption	common title-above the figures if there are more than one. If NULL, as by default, a possible abbreviated version of deparse(x$call) is used.
qqline	logical; if TRUE and the fit does *not* contain simulated envelopes, a qqline passing through the first and third quartiles of a standard normal distribution will be added to the normal Q-Q plot.
col.qqline	color of the qqline.
main	character; title to be placed at each plot additionally (and above) all captions.
line_col_env	line color for the upper and lower quantile curves of the simulated envelopes if the mixpoissonreg object was fitted with envelopes.
line_col_median	
	line color for the median curve of the simulated envelopes if the mixpoissonreg object was fitted with envelopes.
fill_col_env	fill color for the simulated envelopes if the mixpoissonreg object was fitted with envelopes.
fill_alpha_env	alpha of the envelope region, when the mixpoissonreg object was fitted with envelopes.
ask	logical; if TRUE, the user is asked before each plot.
labels.id	vector of labels, from which the labels for extreme points will be chosen. The default uses the observation numbers.
label.pos	positioning of labels, for the left half and right half of the graph respectively, for plots 2 and 6.
type.cookplot	character; what type of plot should be drawn for Cook's and Generalized Cook's distances (plots 3 and 4). The default is 'h'.
id.n	number of points to be labelled in each plot, starting with the most extreme.
col.id	color of point labels.
cex.id	magnification of point labels.
cex.oma.main	controls the size of the sub.caption only if that is above the figures when there is more than one.
cex.caption	controls the size of caption.
col.caption	controls the caption color.
cex.points	controls the size of the points.

col.points   controls the colors of the points.

include.modeltype

                      logical. Indicates whether the model type ('NB' or 'PIG') should be displayed
                      on the captions.

include.residualtype

                      local. Indicates whether the name of the residual ('Pearson' or 'Score') should
                      be displayed on the caption of plot 1 (Residuals vs. Index).

...   graphical parameters to be passed.

Details

The `plot` method is implemented following the same structure as the [plot.lm](), so it will be easy to
be used by practitioners that are familiar with `glm` objects.

These plots allows one to perform residuals analsysis and influence diagnostics. There are other
global influence functions, see [influence.mixpoissonreg]().

See Barreto-Souza and Simas (2016), Cook and Weisberg (1982) and Zhu et al. (2001).

Value

It is called for its side effects.

References

DOI:10.1007/s11222-015-9601-6 doi: [10.1007/s1122201596016](http://doi.org/10.1007/s1122201596016)(Barreto-Souza and Simas; 2016)

Cook, D.R. and Weisberg, S. (1982) *Residuals and Influence in Regression*. (New York: Chapman
and Hall, 1982)

Zhu, H.T., Lee, S.Y., Wei, B.C., Zhu, J. (2001) *Case-deletion measures formodels with incomplete
data.* Biometrika, 88, 727–737.

See Also

[autoplot.mixpoissonreg](), [local_influence_plot.mixpoissonreg](), [local_influence_autoplot.mixpoissonreg](),
[summary.mixpoissonreg](), [predict.mixpoissonreg](), [influence.mixpoissonreg]()

Examples

```
daysabs_prog <- mixpoissonreg(daysabs ~ prog, data = Attendance)
plot(daysabs_prog)
```

predict.mixpoissonreg *Predict Method for* mixpoissonreg *Objects*

Description

Function to obtain various predictions based on the fitted mixed Poisson regression models.

Usage

```
## S3 method for class 'mixpoissonreg'
predict(
  object,
  newdata = NULL,
  type = c("response", "link", "precision", "variance"),
  se.fit = FALSE,
  interval = c("none", "confidence", "prediction"),
  level = 0.95,
  nsim_pred = 100,
  nsim_pred_y = 100,
  ...
)
```

Arguments

object	object of class "mixpoissonreg" containing results from the fitted model.
newdata	optionally, a data frame in which to look for variables with which to predict. If omitted, the fitted response values will be provided.
type	the type of prediction. The default is the "response" type, which provided the estimated values for the means. The type "link" provides the estimates for the linear predictor. The type "precision" provides estimates for the precision parameters whereas the type "variance" provides estimates for the variances.
se.fit	logical switch indicating if standard errors on the scale of linear predictors should be returned. If TRUE, it only returns the standard deviations of the linear predictors when type = 'link', otherwise returns NA and a warning indicating that the type must be 'link'. When using mixpoissonreg objects, the fit must be done with x = TRUE.
interval	Type of interval calculation for the response variables, 'none', 'confidence' or 'prediction'. If 'confidence', the confidence intervals for the means are returned. If 'prediction', prediction intervals for future response variables are reported. For confidence intervals, the type of the prediction must be 'response' or 'link'. For prediction intervals the type of prediction must be 'response'. For 'confidence' intervals, when using mixpoissonreg objects, the fit must be done with x = TRUE and for predictions intervals, the fit must be done with x = TRUE and w = TRUE.
level	Tolerance/confidence level. The default is set to 0.95.

nsim_pred	number of means and predictions to be generated in each step of the simulation. The default is set to 100.
nsim_pred_y	number of response variables generated for each pair of mean and precision to compute the prediction intervals. The default is set to 100.
...	further arguments passed to or from other methods.

Details

The se.fit argument only returns a non-NA vector for type = 'link', that is, on the scale of the linear predictor for the mean parameter. For the response scale, one can obtain confidence or prediction intervals. It is important to notice that confidence intervals *must not* be used for future observations as they will underestimate the uncertainty. In this case prediction intervals should be used. Currently, we do not have closed-form expressions for the prediction interval and, therefore, they are obtained by simulation and can be computationally-intensive.

Value

A vector containing the predicted values if se.fit=FALSE, a list with elements *fit* and *se.fit* if se.fit=TRUE, and a matrix if interval is set to *confidence* or *prediction*.

See Also

[fitted.mixpoissonreg](), [summary.mixpoissonreg](), [plot.mixpoissonreg](), [autoplot.mixpoissonreg](), [coef.mixpoissonreg](), [vcov.mixpoissonreg](), [plot.mixpoissonreg]()

Examples

```
daysabs_prog <- mixpoissonreg(daysabs ~ prog, data = Attendance)
predict(daysabs_prog)
```

---

residuals.mixpoissonreg

Residuals Method for mixpoissonreg *Objects*

Description

Function to return 'pearson' or 'score' residuals for mixed Poisson regression models.

Usage

```
## S3 method for class 'mixpoissonreg'
residuals(object, type = c("pearson", "score"), ...)
```

Arguments

`object`	an object of class "mixpoissonreg" containing results from the fitted model.
`type`	the type of residual to be returned. Currently, the options are 'pearson' or 'score'. The default is set to 'pearson'. Notice that these residuals coincide for Negative-Binomial models.
`...`	Currently not used.

Value

A vector containing the residuals of a *mixpoissonreg* object.

See Also

[plot.mixpoissonreg](), [predict.mixpoissonreg](), [autoplot.mixpoissonreg](), [summary.mixpoissonreg]()

Examples

```
daysabs_prog <- mixpoissonreg(daysabs ~ prog, data = Attendance)
residuals(daysabs_prog)
```

---

summary.mixpoissonreg   *Summary Method for* mixpoissonreg *Objects.*

Description

Function providing a summary of results related to mixed Poisson regression models.

Usage

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

Arguments

`object`	an object of class "mixpoissonreg" containing results from the fitted model.
`...`	further arguments passed to or from other methods.

Value

An object of class `summary_mixpoissonreg` containing several informations of a *mixpoissonreg* object.

See Also

[plot.mixpoissonreg](), [autoplot.mixpoissonreg](), [local_influence_plot.mixpoissonreg](), [local_influence_autopl

Examples

```
daysabs_prog <- mixpoissonreg(daysabs ~ prog, data = Attendance)
summary(daysabs_prog)
```

---

tidy.mixpoissonreg   *Tidy a* mixpoissonreg *object*

Description

Tidy returns a [tibble::tibble()](#) containing informations on the coefficients of the model, such as the estimated parameters, standard errors, z-statistics and p-values. Additionally, it may return confidence intervals for the model parameters.

Usage

```
## S3 method for class 'mixpoissonreg'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x	A mixpoissonreg object.
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
...	Additional arguments. Currently not used.

Value

A *tibble* containing the coefficients of the fitted *mixpoissonreg* object along with its estimates, std. errors, z-statistics and p-values.

See Also

[glance.mixpoissonreg](#), [augment.mixpoissonreg](#), [tidy_local_influence.mixpoissonreg](#), [autoplot.mixpoissonreg](#) [local_influence_autoplot.mixpoissonreg](#)

---

tidy_local_influence   *Tidy Functions for Local Influence Diagnostics*

Description

Functions to provide tidy outputs or ggplot2-based plots of local influence diagnostics.

Usage

```
local_influence_autoplot(model, ...)

tidy_local_influence(model, ...)

local_influence_benchmarks(model, ...)
```

Arguments

model   A model object for which local influence diagnostics are desired.

...   additional arguments to be passed.

Details

Local influence diagnostics were first introduced by Cook (1986), where several perturbation schemes were introduced and normal curvatures were obtained. Poon and Poon (1999) introduced the conformal normal curvature, which has nice properties and takes values on the unit interval $[0, 1]$. Zhu and Lee (2001) following Cook (1986) and Poon and Poon (1999) introduced normal and conformal normal curvatures for EM-based models.

Value

The *tidy_local_influence* method returns a *tibble* containing the resulting perturbation schemes as columns. The *local_influence_benchmarks* method returns a *tibble* with the benchmarks as columns. The *local_influence_autoplot* method is called for its side effects.

References

Cook, R. D. (1986) *Assessment of Local Influence.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 48, pp.133-169. https://rss.onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1986.tb01398.x

Poon, W.-Y. and Poon, Y.S. (1999) *Conformal normal curvature and assessment of local influence.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 61, pp.51-61. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00162

Zhu, H.-T. and Lee, S.-Y. (2001) *Local influence for incomplete data models.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 63, pp.111-126. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00279

---

tidy_local_influence.mixpoissonreg

> *Tidy Functions for Local Influence Diagnostics for* mixpoissonreg
> *Objects*

Description

Functions to provide tidy outputs of local influence diagnostics. tidy_local_influence.mixpoissonreg provides a tibble::tibble() containing the local influence diagnostics under the chosen perturbation schemes. local_influence_benchmarks.mixpoissonreg provides a tibble::tibble() with a single row and one column for each selected perturbation scheme containing influential benchmarks for each perturbation scheme.

Usage

```
## S3 method for class 'mixpoissonreg'
tidy_local_influence(
  model,
  perturbation = c("case_weights", "hidden_variable", "mean_explanatory",
    "precision_explanatory", "simultaneous_explanatory"),
  curvature = c("conformal", "normal"),
  direction = c("canonical", "max.eigen"),
  parameters = c("all", "mean", "precision"),
  mean.covariates = NULL,
  precision.covariates = NULL,
  ...
)

## S3 method for class 'mixpoissonreg'
local_influence_benchmarks(
  model,
  perturbation = c("case_weights", "hidden_variable", "mean_explanatory",
    "precision_explanatory", "simultaneous_explanatory"),
  curvature = c("conformal", "normal"),
  direction = c("canonical", "max.eigen"),
  parameters = c("all", "mean", "precision"),
  mean.covariates = NULL,
  precision.covariates = NULL,
  ...
)
```

Arguments

model   A mixpoissonreg model.

perturbation   a list or vector of perturbation schemes to be returned. The currently available
                 schemes are "case_weights", "hidden_variable", "mean_explanatory", "preci-

sion_explanatory", "simultaneous_explanatory". See Barreto-Souza and Simas (2016) for further details.

curvature   the curvature to be returned, 'conformal' for the conformal normal curvature (see Zhu and Lee, 2001 and Poon and Poon, 1999) or 'normal' (see Zhu and Lee, 2001 and Cook, 1986).

direction   the 'max.eigen' returns the eigenvector associated to the largest eigenvalue of the perturbation matrix. The 'canonical' considers the curvatures under the canonical directions, which is known as "total local curvature" (see Lesaffre and Verbeke, 1998). For conformal normal curvatures both of them coincide. The default is 'canonical'.

parameters   the parameter to which the local influence will be computed. The options are 'all', 'mean' and 'precision'. This argument affects the 'case_weights' and 'hidden_variable' perturbation schemes. The default is 'all'.

mean.covariates

a list or vector of characters containing the mean-explanatory variables to be used in the 'mean-explanatory' and 'simultaneous-explanatory' perturbation schemes. If NULL, the 'mean-explanatory' and 'simultaneous-explanatory' perturbation schemes will be computed by perturbing all mean-related covariates. The default is NULL.

precision.covariates

a list or vector of characters containing the precision-explanatory variables to be used in the 'precision-explanatory' and 'simultaneous-explanatory' perturbation schemes. If NULL, the 'precision-explanatory' and 'simultaneous-explanatory' perturbation schemes will be computed by perturbing all precision-related covariates. The default is NULL.

...   Currently not used.

Value

A tibble containing the perturbations schemes as columns.

References

DOI:10.1007/s11222-015-9601-6 doi: [10.1007/s1122201596016](10.1007/s1122201596016)(Barreto-Souza and Simas; 2016)

Cook, R. D. (1986) *Assessment of Local Influence.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 48, pp.133-169. https://rss.onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1986.tb01398.x

Lesaffre, E. and Verbeke, G. (1998) *Local Influence in Linear Mixed Models.* Biometrics, 54, pp. 570-582.

Poon, W.-Y. and Poon, Y.S. (1999) *Conformal normal curvature and assessment of local influence.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 61, pp.51-61. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00162

Zhu, H.-T. and Lee, S.-Y. (2001) *Local influence for incomplete data models.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 63, pp.111-126. https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00279

See Also

glance.mixpoissonreg, augment.mixpoissonreg, tidy.mixpoissonreg, autoplot.mixpoissonreg

Examples

```
daysabs_prog <- mixpoissonreg(daysabs ~ prog | prog, data = Attendance)
tidy_local_influence(daysabs_prog)
```

---

vcov.mixpoissonreg   *Calculate Variance-Covariance Matrix for* mixpoissonreg *Objects*

Description

Returns the variance-covariance matrix of the parameters for fitted mixed Poisson regression models. The `parameters` argument indicates for which parameters the variance-covariance matrix should be computed, namely, 'mean' for mean-relatex parameters or 'precision' for precision-related parameters.

Usage

```
## S3 method for class 'mixpoissonreg'
vcov(object, parameters = c("all", "mean", "precision"), ...)
```

Arguments

object	an object of class "mixpoissonreg" containing results from the fitted model.
parameters	a string to determine which coefficients should be extracted: 'all' extracts all coefficients, 'mean' extracts the coefficients of the mean parameters and 'precision' extracts coefficients of the precision parameters.
...	further arguments passed to or from other methods.

Value

A matrix containing the covariance matrix of a *mixpoissonreg* object.

See Also

coef.mixpoissonreg

Examples

```
daysabs_prog <- mixpoissonreg(daysabs ~ prog, data = Attendance)
vcov(daysabs_prog)
```

Index