Package: lacm (via r-universe)

October 8, 2024

Version 0.1.1

Priority optional

Title Latent Autoregressive Count Models

Description Perform pairwise likelihood inference in latent autoregressive count models. See Pedeli and Varin (2020) for details.

Maintainer Cristiano Varin <`cristiano.varin@unive.it`>

Depends R (>= 3.4.0)

Imports graphics, numDeriv, statmod, stats

License GPL (>= 2)

NeedsCompilation yes

Author Xanthi Pedeli [aut], Cristiano Varin [aut, cre]

Date/Publication 2020-06-20 22:00:02 UTC

Repository https://crisvarin.r-universe.dev

RemoteUrl https://github.com/cran/lacm

RemoteRef HEAD

RemoteSha 9cae798b73ae6f2ccf92f01b77e80b40e12e0296

Contents

`CLIC` *Composite Likelihood Information Criterion*

Description

Calculates the composite likelihood information criterion for a latent autoregressive count model fitted through maximum pairwise likelihood.

Usage

```
CLIC(object, ...)
```

Arguments

`object`	a fitted model object of class `"lacm"`.
`...`	optional arguments.

Details

Function `CLIC` computes the composite likelihood information criterion (Varin and Vidoni, 2005) for a latent autoregressive count model estimated by maximum pairwise likelihood. See Pedeli and Varin (2020) for details.

When comparing models fitted by maximum pairwise likelihood to the same data, the smaller the CLIC, the better the fit.

Value

a numeric value with the corresponding CLIC.

Author(s)

Xanthi Pedeli and Cristiano Varin.

References

Pedeli, X. and Varin, C. (2020). Pairwise likelihood estimation of latent autoregressive count models. *Statistical Methods in Medical Research*.doi: 10.1177/0962280220924068.

Varin, C. and Vidoni, P. (2005). A note on composite likelihood inference and model selection. *Biometrika*, **92**, 519–528.

See Also

`lacm`.

Examples

```
data("polio", package = "lacm")
## model components
trend <- 1:length(polio)
sin.term <- sin(2 * pi * trend / 12)
cos.term <- cos(2 * pi * trend / 12)
sin2.term <- sin(2 * pi * trend / 6)
cos2.term <- cos(2 * pi * trend / 6)
## fit model with pairwise likelihood of order 1
mod1 <- lacm(polio ~ I(trend * 10^(-3)) + sin.term + cos.term + sin2.term + cos2.term)
CLIC(mod1)
```

`lacm` *Fitting Latent Autoregressive Count Models by Maximum Pairwise Likelihood*

Description

Fits latent autoregressive count models by maximum pairwise likelihood.

Usage

```
lacm(formula, data, subset, offset, contrasts = NULL, start.theta = NULL, fixed, d = 1,
     kernel.type = c("Rectangular", "Trapezoidal"), fit = TRUE, gh.num = 20,
    reltol.opt = 1e-06, opt.method = c("BFGS", "Nelder-Mead"), maxit.opt = 1000,
     sandwich.lag = NULL, bread.method = c("Outer-product", "Hessian"), ...)
```

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 details of model specification are given under 'Details'.
`data`	an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which `lacm` is called.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process.
`offset`	this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be `NULL` or a numeric vector of length equal to the number of cases. One or more `offset` terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See `model.offset`.
`contrasts`	an optional list. See the `contrasts.arg` of `model.matrix.default`.
`start.theta`	optional numeric vector with starting values for the model parameters.

`fixed`	optional numeric vector of the same length as the total number of parameters. If supplied, only `NA` entries in fixed will be varied.
`d`	order of the pairwise likelihood. Defaults to 1.
`kernel.type`	one of `"Rectangular"`, `"Trapezoidal"` indicating the type of kernel weights to be used in the weighted pairwise likelihood specification. The default `"Rectangular"` corresponds to equal contribution from all pairs of observations that are distant apart up to lag `d`. Can be abbreviated.
`fit`	a logical value indicating whether to compute the maximum pairwise likelihood estimates or not.
`gh.num`	number of the Gauss-Hermite quadrature nodes. Defaults to 20.
`reltol.opt`	relative convergence tolerance to be passed to `optim`. Defaults to 1e-6.
`opt.method`	one of `"BFGS"` or `"Nelder-Mead"` indicating the optimization method to be passed to `optim`. Can be abbreviated. See `optim` for details.
`maxit.opt`	the maximum number of iterations to be passed to `optim`. Defaults to 1000.
`sandwich.lag`	the lag length used for computing the bandwith of the sandwich variance. See 'Details'.
`bread.method`	one of `"Outer-product"` or `"Hessian"` indicating whether the bread matrix of the sandwich variance is estimated with the outer-product of the individual gradients or with a numerical approximation of the Hessian matrix. Can be abbreviated. See 'Details'.
`...`	further arguments passed to or from other methods.

Details

Function `lacm` performs maximum pairwise likelihood estimation in latent autoressive count models, see Pedeli and Varin (2020) for details.

Evaluation of the pairwise likelihood is performed through double Gauss-Hermite quadrature with the `gh.num` nodes and weights calculated by `gauss.quad`.

Standard formula `y ~ x1 + x2` indicates that the mean response is modelled as a function of covariates `x1` and `x2` through a log link function.

Starting values supplied by the user can be specified through `start.theta`. If `start.theta=NULL`, then starting values are obtained using the method-of-moments as discussed in Ng et al. (2011).

Sandwich standard errors that are robust to heteroschedasticity and autocorrelation (HAC-type) are computed. The "meat matrix" is estimated using a lag length equal to `sandwich.lag`. If `sandwich.lag` is `NULL`, then `L` is set equal to `10 * log10(n)`, where `n` is the time series length and `d` is the pairwise likelihood order. The "bread matrix" is computed with the average of outer products of the individual grandients (`bread.matrix = "Outer-product"`) or with a numerical approximation of the Hessian (`bread.method = "Hessian"`). Details are given in Pedeli and Varin (2020).

Value

An object of class `"lacm"` with the following components:

`nobs`	the number of observations.

`p`	the number of regressors, including the intercept.
`d`	the order of the pairwise likelihood.
`npar`	the number of parameters.
`Y`	the response used.
`X`	the model matrix used for the mean response.
`offset`	the offset used.
`sandwich.lag`	the lag length used for the bandwith of the HAC-type standard errors.
`fit`	logical. Was the model fitted or not?
`gh.num`	number of Gauss-Hermite nodes used.
`call`	the matched call.
`terms`	the `terms` object used.
`latent`	logical. Does the model include the latent part?
`fixed`	the numeric vector indicating which parameters are constants.
`ifree`	indeces of the free parameters.
`kweights`	the kernel weights used.
`start.theta`	the starting values.
`objfun`	function computing the logarithm of the pairwise likelihood of order d.
`grad`	function computing the gradient of the pairwise likelihood of order d.
`gh`	Gauss-Hermite nodes and weights used.
`opt.method`	a character string specifying the method argument passed to optim. The default optimization routine is the quasi-Newton algorithm BFGS. See `optim` for details.
`convergence`	an integer code indicating convergence of the optimizer. See `link{optim}` for details.
`gh`	a list with components the Gauss-Hermite `nodes` and the `weights` used for approximating the pairwise likelihood.
`plik`	the maximum pairwise likelihood value.
`theta`	the maximum pairwise likelihood estimate.
`jacobian`	the jacobian of the individual pairwise likelihood terms.
`outer-product`	logical. Was the bread matrix of the sandwich variance computed with the outer product of the individual scores?
`H`	the bread matrix.
`J`	the meat matrix.
`vcov`	the variance-covariance matrix of the maximum pairwise likelihood estimate.
`CLIC`	the composite likelihood information criterion.

Functions `summary.lacm`, `coefficients` and `vcov.lacm` can be used to obtain or print a summary of the results, extract coefficients and their estimated variance-covariance matrix of the model fitted by `lacm`.

Author(s)

Xanthi Pedeli and Cristiano Varin.

References

Ng, C., Joe, H., Karlis, D., and Liu, J. (2011). Composite likelihood for time series models with a latent autoregressive process. *Statistica Sinica*, **21**, 279–305.

Pedeli, X. and Varin, C. (2020). Pairwise likelihood estimation of latent autoregressive count models. *Statistical Methods in Medical Research*.doi: 10.1177/0962280220924068.

See Also

`CLIC`.

Examples

```
data("polio", package = "lacm")
## model components
trend <- 1:length(polio)
sin.term <- sin(2 * pi * trend / 12)
cos.term <- cos(2 * pi * trend / 12)
sin2.term <- sin(2 * pi * trend / 6)
cos2.term <- cos(2 * pi * trend / 6)
## fit model with pairwise likelihood of order 1
mod1 <- lacm(polio ~ I(trend * 10^(-3)) + sin.term + cos.term + sin2.term + cos2.term)
mod1
summary(mod1)
## refit with d = 3
mod3 <- update(mod1, d = 3)
summary(mod3)
```

`polio` *Polio Time Series*

Description

Time series of Polio incidences in USA from 1970 to 1983.

Usage

```
data(polio)
```

Format

Time series of monthly Polio cases in USA from January 1970 to December 1983.

Source

Zeger, S.L. (1988). A regression model for time series of counts. *Biometrika* **75**, 822–835.

Examples

```
data(polio, package = "lacm")
```

`summary.lacm` *Methods for* `lacm` *Objects*

Description

Methods for fitted latent autoregressive count model objects of class "`lacm`"

Usage

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

## S3 method for class 'lacm'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'lacm'
coef(object, ...)

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

## S3 method for class 'lacm'
simulate(object, nsim = 1, seed = NULL, ...)
```

Arguments

`object, x`	a fitted model object of class "`lacm`".
`digits`	the number of significant digits to use when printing.
`nsim`	number of response vectors to simulate. Defaults to 1.
`seed`	an object specifying if and how the random number generator should be initialized ('seeded'). See `simulate`.
`...`	additional optional arguments.

Value

The function `summary.lacm` returns an object of class `"summary.lacm"`, a list of some components of the `"lacm"` object, plus

`coefficients`	a summary of the parameter estimates, standard errors, z-values and corresponding p-values.
`clic`	the composite likelihood information criterion.

The function `simulate.lacm` returns a list of simulated responses.

The function `print` returns the `call` and `coefficients`, `coef` returns the estimated coefficients and `vcov` the corresponding variance-covariance matrix.

Author(s)

Xanthi Pedeli and Cristiano Varin.

References

Pedeli, X. and Varin, C. (2020). Pairwise likelihood estimation of latent autoregressive count models. *Statistical Methods in Medical Research*.doi: 10.1177/0962280220924068.

See Also

`CLIC`.

Examples

```
data("polio", package = "lacm")
## model components
trend <- 1:length(polio)
sin.term <- sin(2 * pi * trend / 12)
cos.term <- cos(2 * pi * trend / 12)
sin2.term <- sin(2 * pi * trend / 6)
cos2.term <- cos(2 * pi * trend / 6)
## fit model with pairwise likelihood of order 1
mod1 <- lacm(polio ~ I(trend * 10^(-3)) + sin.term + cos.term + sin2.term + cos2.term)
mod1
summary(mod1)
## refit with d = 3
mod3 <- update(mod1, d = 3)
summary(mod3)
```

Index