Package ‘bgeva’

October 12, 2022

Version 0.3-1

Author Giampiero Marra, Raffaella Calabrese and Silvia Angela Osmetti

Maintainer Giampiero Marra <giampiero.marra@ucl.ac.uk>

Title Binary Generalized Extreme Value Additive Models

Description

Routine for fitting regression models for binary rare events with linear and nonlinear covariate effects when using the quantile function of the Generalized Extreme Value random variable.

Depends R (>= 2.15.1), mgcv

Imports magic, trust

LazyLoad yes

License GPL (>= 2)

URL http://www.ucl.ac.uk/statistics/people/giampieromarra

Repository CRAN

Date/Publication 2017-05-19 06:07:36 UTC

NeedsCompilation no

R topics documented:

bgeva-package *Binary Generalized Extreme Value Additive Modelling*

Description

bgeva provides a function for univariate modelling for binary rare events data with linear and nonlinear predictor effects when using the quantile function of the Generalized Extreme Value random variable.

Details

bgeva provides a function for flexible regression models for binary rare events data. The underlying representation and estimation of the model is based on a penalized regression spline approach, with automatic smoothness selection. The numerical routine carries out function minimization using a trust region algorithm from the package trust in combination with an adaptation of a low level smoothness selection fitting procedure from the package mgcv.

bgeva supports the use of many smoothers as extracted from mgcv. Scale invariant tensor product smooths are not currently supported. Estimation is by penalized maximum likelihood with automatic smoothness selection achieved by using the approximate Un-Biased Risk Estimator (UBRE).

Confidence intervals for smooth components are derived using a Bayesian approach. Approximate p-values for testing individual smooth terms for equality to the zero function are also provided. Functions plot.bgeva and summary.bgeva extract such information from a fitted bgevaObject. Variable selection is also possible via the use of shrinakge smoothers or information criteria.

Consider also using the faster and more stable version implemented in the gamlss() function of the SemiParBIVProbit package. gamlss() also allows for a much wider choice of smoothers.

Author(s)

Raffaella Calabrese (University of Milano-Bicocca, Department of Statistics and Quantitative Methods), Giampiero Marra (University College London, Department of Statistical Science) and Silvia Osmetti (University Cattolica del Sacro Cuore, Department of Statistics)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

References

Calabrese R., Marra G., Osmetti S.A. (2016), Bankruptcy Prediction of Small and Medium Enterprises Using a Flexible Binary Generalized Extreme Value Model. *Journal of the Operational Research Society*, 67(4), 604-615.

See Also

bgeva

bg.checks *Some convergence diagnostics*

Description

It takes a fitted bgeva object produced by bgeva() and produces some diagnostic information about the fitting procedure.

Usage

```
bg.checks(x)
```

Arguments

x	A bgeva object produced by bgeva().

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also

bgeva

bgeva *Binary Generalized Extreme Value Additive Modelling*

Description

bgeva can be used to fit regression models for binary rare events where the link function is the quantile function of the Generalized Extreme Value random variable. The linear predictor can be flexibly specified using parametric and regression spline components. Regression spline bases are extracted from the package mgcv. Multi-dimensional smooths are available via the use of penalized thin plate regression splines (isotropic). The current implementation does not support scale invariant tensor product smooths.

Consider also using the faster and more stable version implemented in the gamlss() function of the SemiParBIVProbit package. gamlss() also allows for a much wider choice of smoothers.

Usage

```
bgeva(formula.eq, data=list(), tau=-0.25, Hes=TRUE, gIM="a", iterlimSP=50,
                  pr.tol=1e-6,
                  gamma=1, aut.sp=TRUE, fp=FALSE, start.v=NULL, start.vo=1,
                  rinit=1, rmax=100, fterm=sqrt(.Machine$double.eps),
                  mterm=sqrt(.Machine$double.eps),
                  control=list(maxit=50,tol=1e-6,step.half=25,
                               rank.tol=sqrt(.Machine$double.eps)))
```

Arguments

`formula.eq` A GAM formula. `s` terms are used to specify smooth functions of predictors. See the examples below and the documentation of `mgcv` for further details on GAM formula specifications.

`data` An optional data frame, list or environment containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which `bgeva` is called.

`tau` Shape parameter of the GEV distribution. It must be provided.

`Hes` If `FALSE`, then the Fisher (rather than the observed) information matrix is employed.

`gIM` Different versions of GEV distribution. Options are `a` and `b`.

`iterlimSP` A positive integer specifying the maximum number of loops to be performed before the smoothing parameter estimation step is terminated.

`pr.tol` Tolerance to use in judging convergence of the algorithm when automatic smoothing parameter selection is used.

`gamma` It is an inflation factor for the model degrees of freedom in the UBRE score. Smoother models can be obtained setting this parameter to a value greater than 1. Typically `gamma=1.4` achieves this.

`aut.sp` If `TRUE`, then automatic multiple smoothing parameter selection is carried out. If `FALSE`, then smoothing parameters are set to the values obtained from the univariate fits.

`fp` If `TRUE`, then a fully parametric model with regression splines is fitted. See the example below.

`start.v` Starting values for the parameters can be provided here.

`start.vo` Default is 1 meaning that starting values are obtained from fitting a logistic model. Otherwise, these can be set as described in Calabrese and Osmetti (2013) (`start.vo=2`) or from a combination of options 1 and 2 (`start.vo=3`).

`rinit` Starting trust region radius. The trust region radius is adjusted as the algorithm proceeds. See the documentation of `trust` for further details.

`rmax` Maximum allowed trust region radius. This may be set very large. If set small, the algorithm traces a steepest descent path.

`fterm` Positive scalar giving the tolerance at which the difference in objective function values in a step is considered close enough to zero to terminate the algorithm.

`mterm` Positive scalar giving the tolerance at which the two-term Taylor-series approximation to the difference in objective function values in a step is considered close enough to zero to terminate the algorithm.

`control` It is a list containing iteration control constants with the following elements: `maxit`: maximum number of iterations of the `magic` algorithm; `tol`: tolerance to use in judging convergence; `step.half`: if a trial step fails then the method tries halving it up to a maximum of `step.half` times; `rank.tol`: constant used to test for numerical rank deficiency of the problem. See the documentation of `magic` in `mgcv` for further details.

Details

The Binary Generalized Extreme Value Additive model has the quantile function of the Generalized Extreme Value (GEV) random variable as link function. The linear predictor is flexibly specified using parametric components and smooth functions of covariates. Replacing the smooth components with their regression spline expressions yields a fully parametric univariate GEV model. In principle, classic maximum likelihood estimation can be employed. However, to avoid overfitting, penalized likelihood maximization has to be employed instead. Here the use of penalty matrices allows for the suppression of that part of smooth term complexity which has no support from the data. The trade-off between smoothness and fitness is controlled by smoothing parameters associated with the penalty matrices. Smoothing parameters are chosen to minimize the approximate Un-Biased Risk Estimator (UBRE).

More details can be found in Calabrese, Marra and Osmetti (2016).

Consider also using the faster and more stable version implemented in the `gamlss()` function of the `SemiParBIVProbit` package. `gamlss()` also allows for a much wider choice of smoothers.

Value

The function returns an object of class `bgeva` as described in `bgevaObject`.

WARNINGS

Any automatic smoothing parameter selection procedure is not likely to work well when the data have low information content. In binary models, this issue is especially relevant the number of observations low. Here, convergence failure is typically associated with an infinite cycling between the two steps detailed above. If this occurs, as some practical solutions, one might either (i) lower the total number of parameters to estimate by reducing the dimension of the regression spline bases, (ii) set the smoothing parameters to the values obtained from the univariate fits (`aut.sp=FALSE`), or (iii) set the smoothing parameters to the values obtained from the non-converged algorithm. The default option is (iii).

The GEV distribution may not be defined for certain combinations of parameter and covariate values. In such cases, a sub-design matrix is formed. This consists of the rows (of the original design matrix) for which the distributrion is defined.

Author(s)

Maintainer: Giampiero Marra <`giampiero.marra@ucl.ac.uk`>

References

Calabrese R., Marra G., Osmetti S.A. (2016), Bankruptcy Prediction of Small and Medium Enterprises Using a Flexible Binary Generalized Extreme Value Model. *Journal of the Operational Research Society*, 67(4), 604-615.

Gu C. (1992), Cross validating non-Gaussian data. *Journal of Computational and Graphical Statistics*, 1(2), 169-179.

Wood S.N. (2004), Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association*, 99(467), 673-686.

See Also

`plot.bgeva`, `bgeva-package`, `bgevaObject`, `summary.bgeva`

Examples

```
library(bgeva)

##########
## EXAMPLE
##########

set.seed(0)

n <- 1500

x1 <- round(runif(n))
x2 <- runif(n)
x3 <- runif(n)

f1 <- function(x) (cos(pi*2*x)) + sin(pi*x)
f2 <- function(x) (x+exp(-30*(x-0.5)^2))

y <- as.integer(rlogis(n, location = -6 + 2*x1 + f1(x2) + f2(x3), scale = 1) > 0)

dataSim <- data.frame(y,x1,x2,x3)

out <- bgeva(y ~ x1 + s(x2) + s(x3))
bg.checks(out)

summary(out)
plot(out,scale=0,pages=1,shade=TRUE)


#
#
```

`bgeva.gIMa` *Internal Function*

Description

It provides the log-likelihood, gradient and Hessian (or Fisher) information matrix for penalized or unpenalized maximum likelihood optimization.

Author(s)

Maintainer: Giampiero Marra <`giampiero.marra@ucl.ac.uk`>

`bgeva.gIMb` *Internal Function*

Description

It provides an alternative version of the log-likelihood, gradient and Hessian (or Fisher) information matrix for penalized or unpenalized maximum likelihood optimization.

Author(s)

Maintainer: Giampiero Marra <`giampiero.marra@ucl.ac.uk`>

`bgevaObject` *Fitted bgeva object*

Description

A fitted Binary Generalized Extreme Value Additive object returned by function `bgeva` and of class.

Value

`fit` A list of values and diagnostics extracted from the output of the algorithm. For instance, `fit$argument` and `fit$S.h` return the estimated parameters and overall penalty matrix scaled by its smoothing parameters, for the model. See the documentation of `trust` for diagnostics.

`coefficients` The coefficients of the fitted model provided as follows. Parametric and regression spline coefficients.

`gam.fit` A univariate logistic additive model object. See the documentation of `mgcv` for full details.

`sp` Estimated smoothing parameters of the smooth components for the fitted model.

fp If TRUE, then a fully parametric model was fitted.

iter.sp Number of iterations performed for the smoothing parameter estimation step.

iter.if Number of iterations performed in the initial step of the algorithm.

iter.inner Number of iterations performed inside smoothing parameter estimation step.

tau The tail parameter of the link function.

n Sample size.

X It returns the design matrix associated with the linear predictor.

Xr It returns the design matrix actually used in model fitting.

good It returns a vector indicating which observations have been discarded in the final iteration.

X.d2 Number of columns of the design matrix. This is used for internal calculations.

l.sp Number of smooth components.

He Penalized hessian.

HeSh Unpenalized hessian.

Vb Inverse of the penalized hessian. This corresponds to the Bayesian variance-covariance matrix used for 'confidence' interval calculations.

F This is given by Vb*HeSh.

t.edf Total degrees of freedom of the estimated model. It is calculated as sum(diag(F)).

bs.mgfit A list of values and diagnostics extracted from magic.

conv.sp If TRUE then the smoothing parameter selection algorithm converged.

wor.c It contains the working model quantities given by the square root of the weight matrix times the pseudo-data vector and design matrix, rW.Z and rW.X.

eta The estimated linear predictor.

logL It returns the value of the (unpenalized) log-likelihood evaluated at the (penalized or unpenalized) parameter estimates.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also

bgeva, plot.bgeva, summary.bgeva

plot.bgeva *bgeva plotting*

Description

It takes a fitted bgeva object produced by bgeva() and plots the component smooth functions that make it up on the scale of the linear predictor.

This function is based on plot.gam() in mgcv. Please see the documentation of plot.gam() for full details.

Usage

```
## S3 method for class 'bgeva'
plot(x, ...)
```

Arguments

x	A fitted bgeva object as produced by bgeva().
...	Other graphics parameters to pass on to plotting commands, as described for plot.gam in mgcv.

Details

This function produces plot showing the smooth terms of a fitted semiparametric bivariate probit model. For plots of 1-D smooths, the x axis of each plot is labelled using the name of the regressor, while the y axis is labelled as s(regr,edf) where regr is the regressor name, and edf the estimated degrees of freedom of the smooth. As for 2-D smooths, perspective plots are produced with the x-axes labelled with the first and second variable names and the y axis is labelled as s(var1,var2,edf), which indicates the variables of which the term is a function and the edf for the term.

If seWithMean=TRUE, then the confidence intervals include the uncertainty about the overall mean. That is, although each smooth is shown centred, the confidence intervals are obtained as if every other term in the model was constrained to have average 0 (average taken over the covariate values) except for the smooth being plotted. The theoretical arguments and simulation study of Marra and Wood (2012) suggests that seWithMean=TRUE results in intervals with close to nominal frequentist coverage probabilities. This option should not be used when fitting a random effect model.

Value

The function generates plots.

WARNING

The function can not deal with smooths of more than 2 variables.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

References

Marra G. and Wood S.N. (2012), Coverage Properties of Confidence Intervals for Generalized Additive Model Components. *Scandinavian Journal of Statistics*, 39(1), 53-74.

See Also

bgeva, summary.bgeva

Examples

```
## see examples for bgeva
```

---

print.bgeva *Print a bgeva object*

Description

The print method for a bgeva object.

Usage

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

Arguments

x	A bgeva object produced by bgeva().
...	Other arguments.

Details

print.bgeva prints out the family, model equation, total number of observations, chosen tail parameter and estimated total effective degrees of freedom for the penalized or unpenalized model.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also

bgeva

print.summary.bgeva *Print a summary.bgeva object*

Description

The print method for a summary.bgeva object.

Usage

```
## S3 method for class 'summary.bgeva'
print(x,digits = max(3, getOption("digits") - 3),
           signif.stars = getOption("show.signif.stars"),...)
```

Arguments

x	A summary.bgeva object produced by summary.bgeva().
digits	Number of digits printed in output.
signif.stars	By default significance stars are printed alongside output.
...	Other arguments.

Details

print.summary.bgeva prints model term summaries.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also

summary.bgeva

S.m *Internal Function*

Description

It provides penalty matrices in a format suitable for the automatic smoothness selection procedure.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

`summary.bgeva` *bgeva summary*

Description

It takes a fitted `bgeva` object produced by `bgeva()` and produces some summaries from it.

Usage

```
## S3 method for class 'bgeva'
summary(object,s.meth=”svd”,sig.lev=0.05,...)
```

Arguments

`object` A fitted `bgeva` object as produced by `bgeva()`.

`s.meth` Matrix decomposition used to determine the matrix root of the covariance matrix. See the documentation of `mvtnorm` for further details.

`sig.lev` Significance level used for intervals obtained via posterior simulation.

`...` Other arguments.

Details

As in the package `mgcv`, based on the results of Wood (2013), ‘Bayesian p-values’ are returned for the smooth terms. These have better frequentist performance than their frequentist counterpart. Let $\hat{\mathbf{f}}$ and $\mathbf{V}_f$ denote the vector of values of a smooth term evaluated at the original covariate values and the corresponding Bayesian covariance matrix, and let $\mathbf{V}_f^{r-}$ denote the rank r pseudoinverse of $\mathbf{V}_f$. The statistic used is $T = \hat{\mathbf{f}}'\mathbf{V}_f^{r-}\hat{\mathbf{f}}$. This is compared to a chi-squared distribution with degrees of freedom given by r, which is obtained by biased rounding of the estimated degrees of freedom. See Wood (2013) for further details.

Note that covariate selection can also be achieved using a single penalty shrinkage approach as shown in Marra and Wood (2011).

Consider also using the version of the model implemented in the `gamlss()` function of the `SemiParBIVProbit` package, where p-value calculations are more rigorous.

Value

`tableP` It returns a table containing parametric estimates, their standard errors, z-values and p-values.

`tableNP` It returns a table of nonparametric summaries for each smooth component including estimated degrees of freedom, estimated rank, approximate Wald statistic for testing the null hypothesis that the smooth term is zero, and p-value.

`n` Sample size.

`tau` Tail parameter of the link function.

`formula` The original GAM formula used.

`l.sp` Number of smooth components.

`t.edf` Total degrees of freedom of the estimated model.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

References

Marra G. and Wood S.N. (2011), Practical Variable Selection for Generalized Additive Models. *Computational Statistics and Data Analysis*, 55(7), 2372-2387.

Wood, S.N. (2013). On p-values for smooth components of an extended generalized additive model. *Biometrika*, 100(1), 221-228.

See Also

bgevaObject, plot.bgeva

Examples

```
## see examples for bgeva
```

working.comp *Internal Function*

Description

It efficiently calculates the working model quantities needed to implement the automatic multiple smoothing parameter procedure by exploiting the band structure of the weight matrix.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

Index