Package 'ppgam'

July 23, 2024

Type Package

Title Generalised Additive Point Process Models

Version 1.0.2

Date 2024-07-23

Author Ben Youngman [aut, cre], Theo Economou [aut]

Maintainer Ben Youngman <b.youngman@exeter.ac.uk>

Description Methods for fitting point processes with parameters of generalised additive
model (GAM) form are provided. For an introduction to point processes see Cox, D.R &
Isham, V. (Point Processes, 1980, CRC Press), GAMs see Wood, S.N. (2017)
<doi:10.1201/9781315370279>, and the fitting approach see Wood, S.N., Pya, N. &
Safken, B. (2016) <doi:10.1080/01621459.2016.1180986>.

License GPL-3

Depends R (>= 3.5.0)

Imports MASS, mgcv, evgam

RoxygenNote 7.3.1

NeedsCompilation no

Repository CRAN

Date/Publication 2024-07-23 15:10:01 UTC

Contents

---

coef.ppgam	*Extract Model Coefficients*

Description

Extract Model Coefficients

Usage

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

Arguments

object	a fitted ppgam object
...	not used

Value

Model coefficients extracted from the object 'object'.

Examples

```
# Times of landfalling US hurricanes
data(USlandfall)

# convert dates to years, as a continuous variable
year <- as.integer(format(USlandfall$date, "%Y"))
day <- as.integer(format(USlandfall$date, "%j"))
USlandfall$year <- year + pmin(day / 365, 1)
hits <- subset(USlandfall, landfall == 1)

m1 <- ppgam( ~ s(year), hits)
coef(m1)
```

---

fitted.ppgam	*Extract Model Fitted Values*

Description

Extract Model Fitted Values

Usage

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

Arguments

object	a fitted ppgam object
...	not used

Value

Fitted values extracted from the object 'object'.

Examples

```
# Times of landfalling US hurricanes
data(USlandfall)

# convert dates to years, as a continuous variable
year <- as.integer(format(USlandfall$date, "%Y"))
day <- as.integer(format(USlandfall$date, "%j"))
USlandfall$year <- year + pmin(day / 365, 1)
hits <- subset(USlandfall, landfall == 1)

m1 <- ppgam( ~ s(year), hits)
fitted(m1)
```

---

plot.ppgam *Plots smooths of a fitted* ppgam *object*

Description

Plots smooths of a fitted ppgam object

Usage

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

Arguments

x	a fitted ppgam object
...	arguments to be passed to mgcv::plot.gam

Value

Simulations of parameters

See Also

plot.gam

Examples

```
# Times of landfalling US hurricanes
data(USlandfall)

# convert dates to years, as a continuous variable
year <- as.integer(format(USlandfall$date, "%Y"))
day <- as.integer(format(USlandfall$date, "%j"))
USlandfall$year <- year + pmin(day / 365, 1)
hits <- subset(USlandfall, landfall == 1)

# this creates nodes in the default way
m1 <- ppgam( ~ s(year), hits)
plot(m1)
```

---

ppgam   *Fit a generalised additive point process model*

Description

Fit a generalised additive point process model

Usage

```
ppgam(
  formula,
  data,
  nodes = NULL,
  weights = 1,
  nquad,
  approx = c("midpoint", "exact"),
  knots = NULL,
  use.data = TRUE,
  trace = 0,
  weight.non.numeric = FALSE
)
```

Arguments

formula	a formula for a Poisson process log intensity function (compatible with gam)
data	a data frame
nodes	a list or data frame; see 'Details'
weights	a scalar, list or vector; see 'Details'
nquad	a scalar giving the number of quadrature nodes for each variable
approx	a length 2 character string; see 'Details'
knots	spline knots to pass to gam

use.data	should splines should be constructed from data (otherwise uses nodes)?
trace	integers controlling what's reported. Defaults to 0
weight.non.numeric	
	should nodes for non-numeric variables be weighted according how often they occur? Defaults to FALSE

Details

ppgam fits a Poisson process with intensity function $\lambda(\mathbf{x})$ for covariate $\mathbf{x} = (x_1, \ldots, x_d)$. The likelihood for this model with events occurring at $\mathbf{x}_i$, for $i = 1, \ldots, n$, is approximated by quadrature with

$$\exp\left[-\sum_{j=1}^{m} w_j \lambda(\mathbf{x}_j^*) \right] \prod_{i=1}^{n} \lambda(\mathbf{x}_i)$$

where $\mathbf{x}_j^*$ and w_j are quadrature nodes and weights, for $j = 1, \ldots, m$, defined with nodes and weights.

formula gives the formula for the log intensity function of a Poisson process. It is passed to gam. If formula has no response, i.e. ~ s(...), then data is assumed to give the times at which events occur. Then nodes is used to control integration of the intensity function. If formula has a response, e.g. y ~ s(...), then y is assumed binary, comprising only zeros and ones. Then data is assumed to give the state space of the Poisson process, (e.g. daily time steps if occurrences of events are measured in days) and ones in y identify when events occur. Note that if formula has no response, data will have n rows, and m rows otherwise.

nodes is used to supply nodes for integrating the Poisson process intensity function by quadrature. It is supplied as a list or data frame.

If nodes is a list, its names must correspond to variables on the r.h.s. of formula. Elements of the list, x, say, can be a vector or 2-column matrix, where length(x) > 1 or nrow(x) > 1. If a matrix, its first and second columns are taken as integration nodes and weights, respectively. If a vector of length 2, it is assumed to give the range of the nquad midpoints used as integration nodes. If a longer vector, it is assumed to be the integration nodes, and nquad is ignored.

If nodes is a data frame, it is assumed to give the integration nodes.

nquad specifies the number of integration nodes per variable, unless nodes are specified in nodes. If a single integer and is.null(names(nquad)) it is used for all variables. Otherwise, names are matched to variables. An error is returned if any variables do not have values specified.

weights controls the quadrature weights. If nodes is a list, a scalar multiplies any weights calculated alongside nodes, i.e. node separations. If nodes is a data frame, weights can be a scalar that is repeated nrow(nodes), or a vector of length nrow(nodes) that gives the weights for each row of nodes.

approx controls quadrature details. Its first term controls the integration method, which uses either midpoint ("midpoint", default), Simpson's ("Simpson") or Gauss-Legendre ("Gauss") rules. The second term of approx controls the integration range, which is either the range of the variable ("exact"), or by calling pretty() ("pretty").

trace controls what is reported. Details of convergence are printed with trace = 1, of nodes with trace = 2, and trace = 3 prints both.

weight.non.numeric applied to any non-numeric variables, and gives non-equal quadrature weights to different nodes if TRUE. So nodes get weights according to their frequency of occurrence in data. If nquad is invoked, only a subset of the unique values of the non-numeric variable are used, which are the nquad with largest weights.

Value

An object of class gam, as returned by mgcv::gam, with parameters, covariance matrices and a few other things swapped

References

Wood, S. N., Pya, N., & Safken, B. (2016). Smoothing parameter and model selection for general smooth models. Journal of the American Statistical Association, 111(516), 1548-1563.

Youngman, B. D., & Economou, T. (2017). Generalised additive point process models for natural hazard occurrence. Environmetrics, 28(4), e2444.

Examples

```
# Times of landfalling US hurricanes
data(USlandfall)

# convert dates to years, as a continuous variable
year <- as.integer(format(USlandfall$date, "%Y"))
day <- as.integer(format(USlandfall$date, "%j"))
USlandfall$year <- year + pmin(day / 365, 1)
hits <- subset(USlandfall, landfall == 1)

# this creates nodes in the default way
m1 <- ppgam( ~ s(year), hits)

# some examples of providing nodes
nodes.year <- list(year=pretty(USlandfall$year, 20))
# as 2 is in trace, nodes and weights are printed
m2 <- ppgam( ~ s(year), hits, nodes = nodes.year, trace = 2)

# alternatively, we might just want to specify how many nodes to use
m3 <- ppgam( ~ s(year), hits, nquad = 30)



data(windstorm)
m4 <- ppgam(~ s(lon, lat, k=20), windstorm)

## Storm peak locations, given the North Atlantic Oscillation (NAO) index
# NAO values from https://crudata.uea.ac.uk/cru/data/nao/nao.dat
# NAO midpoints and weights based on `hist'

NAO.mids <- c(-2.75, -2.25, -1.75, -1.25, -0.75, -0.25, 0.25, 0.75, 1.25, 1.75, 2.25)
NAO.wts <- c(0.002, 0.014, 0.057, 0.145, 0.302, 0.427, 0.463, 0.364, 0.171, 0.047, 0.007)

m5 <- ppgam(~ te(lat, lon, NAO, d = 2:1, k = c(40, 8), bs = c("ts", "cr")), windstorm,
```

```
            nodes = list(NAO = cbind(NAO.mids, NAO.wts)))
```

---

predict.ppgam   *Predictions from a fitted* ppgam *object*

Description

Predictions from a fitted ppgam object

Usage

```
## S3 method for class 'ppgam'
predict(object, newdata, type = "link", se.fit = FALSE, ...)
```

Arguments

object	a fitted ppgam object
newdata	a data frame
type	a character string giving the type of prediction sought; see Details. Defaults to "link"
se.fit	a logical: should estimated standard errors be returned? Defaults to FALSE
...	passed to mgcv::predict()

Details

This calls predict.gam and gives predictions of the intensity function of the Poisson process on the original scale if type = "response", on log scale if type = "link" (default), and of the design matrix if type = "lpmatrix".

Value

A data frame or list of predictions

References

Youngman, B. D., & Economou, T. (2017). Generalised additive point process models for natural hazard occurrence. Environmetrics, 28(4), e2444.

See Also

predict.gam

Examples

```
# Times of landfalling US hurricanes
data(USlandfall)

# convert dates to years, as a continuous variable
year <- as.integer(format(USlandfall$date, "%Y"))
day <- as.integer(format(USlandfall$date, "%j"))
USlandfall$year <- year + pmin(day / 365, 1)
hits <- subset(USlandfall, landfall == 1)

# this creates nodes in the default way
m1 <- ppgam( ~ s(year), hits)
predict(m1)
predict(m1, type = "response")
predict(m1, type = "lpmatrix")
predict(m1, newdata = data.frame(year = c(2000, 2001)))
predict(m1, se.fit = TRUE)
```

---

print.ppgam   *Print a fitted* ppgam *object*

Description

Print a fitted ppgam object

Usage

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

Arguments

x	a fitted ppgam object
...	other arguments passed to print.gam

Details

Calls print.gam.

Value

Prints a details of a fitted ppgam object

See Also

print.gam

Examples

```
# Times of landfalling US hurricanes
data(USlandfall)

# convert dates to years, as a continuous variable
year <- as.integer(format(USlandfall$date, "%Y"))
day <- as.integer(format(USlandfall$date, "%j"))
USlandfall$year <- year + pmin(day / 365, 1)
hits <- subset(USlandfall, landfall == 1)

# this creates nodes in the default way
m1 <- ppgam( ~ s(year), hits)
print(m1)
```

---

simulate.ppgam *Simulations from a fitted* ppgam *object*

Description

Simulations from a fitted ppgam object

Usage

```
## S3 method for class 'ppgam'
simulate(object, nsim = 1, seed = NULL, newdata, type = "link", ...)
```

Arguments

object	a fitted ppgam object
nsim	an integer giving the number of simulations
seed	an integer giving the seed for simulations
newdata	a data frame
type	a character string, as in `predict.ppgam`
...	arguments to be passed to `predict.ppgam`

Value

Simulations of parameters

See Also

predict.ppgam

Examples

```
# Times of landfalling US hurricanes
data(USlandfall)

# convert dates to years, as a continuous variable
year <- as.integer(format(USlandfall$date, "%Y"))
day <- as.integer(format(USlandfall$date, "%j"))
USlandfall$year <- year + pmin(day / 365, 1)
hits <- subset(USlandfall, landfall == 1)

# this creates nodes in the default way
m1 <- ppgam( ~ s(year), hits)
simulate(m1)
simulate(m1, type = "response")
simulate(m1, newdata = data.frame(year = c(2000, 2001)))
```

USlandfall   *Times of landfalling US hurricanes*

Description

A data frame:

Usage

```
data(USlandfall)
```

Format

A data frame with 61129 rows and 2 variables

The variables are as follows:

date date of landfall, as class "Date"

landfall an integer: did a hurricane make landfall on this day?

References

https://www.nhc.noaa.gov/data/

Examples

```
data(USlandfall)
plot(USlandfall, type="h")
```

---

windstorm   *Locations of windstorm peaks and tracks over the North Atlantic*

Description

A dataset in windstorm peaks between 1st January 1979 and 31st December 2014 occurring in [-50, 33] longitude and [36, 77] latitude.

Usage

```
data(windstorm)
```

Format

A data frame with 3133 rows and 4 variables

The variables are as follows:

date date of peak, as class "Date"

lon longitude, in degrees

lat latitude, in degrees

NAO North Atlantic Oscillation index

References

Youngman, B. D., & Economou, T. (2017). Generalised additive point process models for natural hazard occurrence. Environmetrics, 28(4), e2444.

Examples

```
data(windstorm)
plot(windstorm[,c("lon", "lat")])
```

Index