Package ‘pspline.inference’

October 14, 2022

Title Estimation of Characteristics of Seasonal and Sporadic Infectious Disease Outbreaks Using Generalized Additive Modeling with Penalized Basis Splines

Date 2021-01-18

Version 1.0.4

Description Inference of infectious disease outcomes using generalized additive (mixed) models with penalized basis splines (P-Splines). See <https://medrxiv.org/cgi/content/short/2020.07.14.20138180v1>.

Depends R (>= 3.4.0)

Imports stats, utils, mgcv, dplyr, magrittr, assertthat, plyr, reshape2, plotrix, rlang

Suggests import, ggplot2, data.table, ggstance, knitr, rmarkdown, roxygen2, kableExtra, doParallel, parallel, stringr, scales, testthat

License Apache License 2.0

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 7.1.0

URL https://github.com/weinbergerlab/pspline.inference

BugReports https://github.com/weinbergerlab/pspline.inference/issues

NeedsCompilation no

Author Ben Artin [aut, cre, cph]

Maintainer Ben Artin <ben@artins.org>

Repository CRAN

Date/Publication 2021-01-19 15:10:02 UTC

R topics documented:

`pspline.estimate.scalars` *Calculates confidence intervals for scalars estimated from generalized additive (mixed) model of an outbreak*

Description

This function performs Monte Carlo sampling of a GAM/GAMM outbreak model. For each sampled curve, it calls `outcomess` to calculate scalar outcomes It then calculates and returns the confidence interval of each scalar outcome

Usage

```
pspline.estimate.scalars(
  model,
  predictors,
  outcomes,
  samples = 100,
  level = 0.95
)
```

Arguments

`model`	model returned by `gam` or `gamm`
`predictors`	data.frame of predictor values at which the model will be evaluated
`outcomes`	function returning calculated scalar outcomes, as described above
`samples`	number of samples of outcomes to draw
`level`	confidence level for estimates

Details

The `outcomes` function must accept (`model`, `params`, `predictors`) and return a one-row data frame in which each column lists the value of a single scalar outcome calculated from the model estimates.

A typical implementation of the `outcomes` function would call `predict` on `model` and `predictors` to obtain model variable estimates at predictor values, then calculate the scalar outcomes of interest and return them in a data frame.

For example, to calculate the time of outbreak peak, you might use this function for `outcomes`:

`calc_peak = function(model, params, time) { incidence = predict(model, data.frame(time=time), type="response") data.frame(peak=time[which.max(incidence)]) }`

The data frame returned by `pspline.estimate.scalars` contains three columns for each outcome calculated by `outcomes`: for outcome `x` returned by `outcomes`, `pspline.estimate.scalars` returns columns `x.lower`, `x.median`, and `x.upper`, corresponding to lower confidence limit, median, and upper confidence limit of `x`.

Value

data frame of estimates, as described above

`pspline.estimate.timeseries` *Calculates confidence intervals for time series sampled from generalized additive (mixed) model of an outbreak*

Description

This function performs a series of Monte Carlo simulations of a GAM/GAMM outbreak model. For each simulated outbreak, it calls `outcome` to calculate a time series for the simulated outbreak (for example, the number of cumulative cases vs time). It then calculates and returns the confidence interval of the simulated time series at each time point across all simulations

Usage

```
pspline.estimate.timeseries(
  model,
  predictors,
  outcome,
  samples = 1000,
  level = 0.95
)
```

Arguments

`model`	model returned by `gam` or `gamm`, with a single parameter (time)
`predictors`	data frame of predictor values at which the model will be evaluated
`outcome`	function returning calculated outcome time series, as described above

`samples`	number of simulations to run
`level`	confidence level for returned estimates

Details

The `outcome` function must accept (`model`, `params`, `time`) and return a vector containing the outcome time series obtained by evaluating the model at the time points given in `time` and using the model parameters given in `params`.

A typical implementation of the `outcome` function would call `predict` on `model` and `time` to obtain the linear predictor matrix, and then post-multiply that matrix by `params`. Having thus obtained model prediction at every time point, it would calculate the desired time series outcome and return it in a vector.

For example, to calculate the time series of the first derivative of incidence, you might use this function for `outcome`:

```
 calc_deriv = function(model, params, time) { eps = 0.001 predictors = predict(model, data.frame(time=time
type="lpmatrix") fit = model$family$linkinv(predictors predictors_eps = predict(model,
data.frame(time=time + eps), type="lpmatrix") fit_eps = model$family$linkinv(predictors_eps
(fit_eps - fit) / eps }
```

The data frame returned by `pspline.estimate.timeseries` contains three columns and one row for each time point in `time`. The columns are `lower`, `median`, and `upper`, containing the median and the confidence interval for the computed outcome time series at each time point.

Value

data frame of estimates, as described above

`pspline.inference` *Inference using penalized basis splines (P-splines) in a generalized additive model (GAM), with applications in infectious disease outbreak modeling*

Description

This package lets you make point and interval estimates of outcomes modeled with a non-linear P-spline GAM.

Details

Applications in infectious disease outbreak modeling include estimating of outbreak onset, peak, or offset, as well as outbreak cumulative incidence over time.

The package can model two types of outcomes: scalar outcomes, which are single-value outcome measures (for example, timing of outbreak peak) and time series characteristics, which are functions of time (for example, infection incidence over time)

For each outcome measure, the package produces median and confidence interval estimates.

Typical use of this package begins by using the package `mgcv` to obtain a GAM/GAMM model of the process under investigation (such as an infectious disease outbreak), followed by calling either `pspline.estimate.scalars` or `pspline.estimate.timeseries` to obtain confidence intervals on the desired outcome measure

Both `pspline.estimate.scalars` and `pspline.estimate.timeseries` allow computation of arbitrary outcome measures, by passing a function that calculates the desired outcome measure into `pspline.estimate.scalars` or `pspline.estimate.timeseries`.

For convenience, this package also includes several utilities specifically aimed at modeling of infectious disease outbreaks, such as `pspline.outbreak.cases` and `pspline.outbreak.cumcases` (for estimation of incidence and cumulative incidence), and `pspline.outbreak.thresholds`, for estimation of outbreak onset and offset.

Author(s)

Ben Artin `<ben@artins.org>`

Examples

```
# Simulate an outbreak for analysis
cases = data.frame(
  time=seq(0, 51),
 cases=rpois(52, c(rep(1, 13), seq(1, 50, length.out=13), seq(50, 1, length.out=13), rep(1, 13)))
)

# Generate GAM model for outbreak; see mgcv for details
library(mgcv)
model = gam(cases ~ s(time, k=10, bs="cp", m=3), family=poisson, data=cases)

# Generate time series at which model will be evaluated for estimates
# Usually you want this to be the same as the time interval that your observations are in, except
# divided into small increments (here, eps). Using a smaller eps gives more accurate estimates,
# but takes longer to run. A value smaller than 0.5 would be better for final analysis
eps = 0.5
estTimes = data.frame(time=seq(min(cases$time) - 0.5, max(cases$time) + 0.5 - eps, by=eps))

# Estimate incidence
estCases = pspline.estimate.timeseries(
  model, estTimes,
  pspline.outbreak.cases,
  # Using a large number of samples makes the analysis more robust;
  # using only 15 samples makes this example run fast (default is 2000)
  samples=15,
  level=.95
)

# Estimate time when outbreak crosses 5\% and 95\% of cumulative case count
onsetThreshold = 0.025
offsetThreshold = 1 - onsetThreshold
thresholds = pspline.estimate.scalars(
  model, estTimes,
  pspline.outbreak.thresholds(onset=onsetThreshold, offset=offsetThreshold),
```

```
    # Using a large number of samples makes the analysis more robust;
    # using only 15 samples makes this example run fast (default is 2000)
    samples=15,
    level=.95
  )

  # Plot cumulative incidence estimates and threshold estimates
  library(ggplot2)
  ggplot() +
   geom_ribbon(data=estCases, aes(x=time, ymin=cases.lower, ymax=cases.upper), fill=grey(.75)) +
    geom_line(data=estCases, aes(x=time, y=cases.median)) +
    geom_point(data=cases, aes(x=time, y=cases)) +
    annotate("rect",
      xmin=thresholds$onset.lower,
      xmax=thresholds$onset.upper,
      ymin=-Inf, ymax=Inf, alpha=.25) +
    annotate("rect",
      xmin=thresholds$offset.lower,
      xmax=thresholds$offset.upper,
      ymin=-Inf, ymax=Inf, alpha=.25) +
   labs(x="Time", y="Incidence")
```

`pspline.outbreak.calc.cumcases` *Calculate cumulative incidence time series from incidence time series*

Description

Correctly handles accumulating over time intervals different from 1

Usage

```
pspline.outbreak.calc.cumcases(time, cases)
```

Arguments

`time`	vector of times
`cases`	vector of corresponding incidences

Value

vector of corresponding cumulative incidences

`pspline.outbreak.cases`
Calculate cumulative incidence for an outbreak

Description

This is useful as `outcome` for `pspline.estimate.timeseries`.

Usage

```
pspline.outbreak.cases(model, data)
```

Arguments

`model`	model returned by `gam` or `gamm`, with a single parameter (time)
`data`	data frame of predictor values at which the model will be evaluated

Value

data frame of predictor values with corresponding cumulative incidence estimates in `$cumcases`

`pspline.outbreak.cumcases`
Calculate cumulative incidence for an outbreak

Description

This is useful as `outcome` for `pspline.estimate.timeseries`.

Usage

```
pspline.outbreak.cumcases(model, data)
```

Arguments

`model`	model returned by `gam` or `gamm`, with a single parameter (time)
`data`	data frame of predictor values at which the model will be evaluated

Value

data frame of predictor values with corresponding cumulative incidence estimates in `$cumcases`

`pspline.outbreak.cumcases.relative` *Calculate relative incidence for an outbreak*

Description

This is useful as `outcome` for `pspline.estimate.timeseries`.

Usage

```
pspline.outbreak.cumcases.relative(model, data)
```

Arguments

`model`	model returned by `gam` or `gamm`, with a single parameter (time)
`data`	data frame of predictor values at which the model will be evaluated

Value

data frame of predictor values with corresponding relative cumulative incidence estimates in `$cumcases.relative`

`pspline.outbreak.thresholds` *Calculate outbreak thresholds for an outbreak*

Description

The result of calling this is useful as `outcomes` for `pspline.estimate.scalars`.

Usage

```
pspline.outbreak.thresholds(onset = NA, offset = NA)
```

Arguments

`onset`	onset threshold (as fraction of total outbreak case count)
`offset`	offset threshold (as fraction of total outbreak case count)

Value

function suitable as outcome estimator parameter of `pspline.estimate.scalars`

---

pspline.validate.scalars

Run a simulation study to validate a scalar estimator

Description

Run a simulation study to validate a scalar estimator

Usage

```
pspline.validate.scalars(
  fun.truth,
  n.truths,
  fun.observations,
  n.observations,
  fun.model,
  fun.outcomes,
  n.samples,
  level
)
```

Arguments

`fun.truth`	function that generates a true state of the system. Takes no arguments, returns data frame of true values for model variables
`n.truths`	number of different truths to generate for simulation study
`fun.observations`	function that generates a set of observations from truth. Takes one argument (truth data frame) and returns data frame of observations
`n.observations`	number of sets of observations to generate for each truth in the simulation study
`fun.model`	function that returns a model to be used for estimation. Takes one argument (observations data frame) and returns the model
`fun.outcomes`	function that calculates the outcomes of interest. Same as outcomes function in `pspline.estimate.scalars`.
`n.samples`	number of samples to use for estimation. See `pspline.estimate.scalars`.
`level`	confidence level to use for estimation. See `pspline.estimate.scalars`.

Value

list of summary (which is a data frame specifying the fraction of true values that were contained in their estimated confidence interval) and results (which is a data frame specifying the quantile of the true value in the estimated sampled distribution for each simulation)

Index