Package 'palm'

July 25, 2025

Type Package

Title Fitting Point Process Models via the Palm Likelihood

Version 1.1.6

Date 2025-07-25

Depends R (>= 3.0.0), Rcpp (>= 0.11.5)

Imports gsl, methods, minqa, mvtnorm, R6

LinkingTo Rcpp

Suggests testthat

Description Functions to fit point process models using the Palm likelihood. First proposed by Tanaka, Ogata, and Stoyan (2008) <DOI:10.1002/bimj.200610339>, maximisation of the Palm likelihood can provide computationally efficient parameter estimation for point process models in situations where the full likelihood is intractable. This package is chiefly focused on Neyman-Scott point processes, but can also fit the void processes proposed by Jones-Todd et al. (2019) <DOI:10.1002/sim.8046>. The development of this package was motivated by the analysis of capture-recapture surveys on which individuals cannot be identified---the data from which can conceptually be seen as a clustered point process (Stevenson, Borchers, and Fewster, 2019 <DOI:10.1111/biom.12983>). As such, some of the functions in this package are specifically for the estimation of cetacean density from two-camera aerial surveys.

License GPL

URL https://github.com/b-steve/palm

LazyData TRUE

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

Author Ben Stevenson [aut, cre]

Maintainer Ben Stevenson <ben.stevenson@auckland.ac.nz>

Repository CRAN

Date/Publication 2025-07-25 05:30:02 UTC

Contents

`boot.palm` *Bootstrapping for fitted models*

Description

Carries out a parametric bootstrap procedure for models fitted using the `palm` package.

Usage

```
boot.palm(fit, N, prog = TRUE)
```

Arguments

`fit`	A fitted object.
`N`	The number of bootstrap resamples.
`prog`	Logical, if `TRUE`, a progress bar is printed to the console.

Value

The original model object containing additional information from the bootstrap procedure. These are accessed by functions such as summary.palm and confint.palm. The bootstrap parameter estimates can be found in the `boots` component of the returned object.

Examples

```
## Fit model.
fit <- fit.ns(example.2D, lims = rbind(c(0, 1), c(0, 1)), R = 0.5)
## Carry out bootstrap.
fit <- boot.palm(fit, N = 100)
## Inspect standard errors and confidence intervals.
summary(fit)
confint(fit)
## Estimates are very imprecise---these data were only used as
## they can be fitted and bootstrapped quickly for example purposes.
```

`coef.palm` *Extract parameter estimates.*

Description

Extracts estimated parameters from an object returned by the fitting functions in this package, such as fit.ns, fit.void, and fit.twocamera.

Usage

```
## S3 method for class 'palm'
coef(object, se = FALSE, ...)

## S3 method for class 'palm_twocamerachild'
coef(object, se = FALSE, report.2D = TRUE, ...)
```

Arguments

`object`	A fitted model object.
`se`	Logical, if `TRUE` standard errors are presented (if available) instead of parameter estimates.
`...`	Other parameters (for S3 generic compatibility).
`report.2D`	Logical, for two-camera model fits only. If `TRUE`, two-dimensional animal density is reported.

`confint.palm` *Extracts Neyman-Scott point process parameter confidence intervals.*

Description

Extracts confidence intervals for estimated and derived parameters from a model fitted using fit.ns, fit.void, or fit.twocamera, then bootstrapped using boot.palm.

Usage

```
## S3 method for class 'palm'
confint(object, parm = NULL, level = 0.95, method = "percentile", ...)
```

Arguments

`object`	A fitted model returned by fit.ns, bootstrapped using boot.
`parm`	A vector of parameter names, specifying which parameters are to be given confidence intervals. Defaults to all parameters.
`level`	The confidence level required.
`method`	A character string specifying the method used to calculate confidence intervals. Choices are "normal", for a normal approximation, and "percentile", for the percentile method.
`...`	Other parameters (for S3 generic compatibility).

Details

Bootstrap parameter estimates can be found in the `boots` component of the model object, so alternative confidence interval methods can be calculated by hand.

Examples

```
## Fitting model.
fit <- fit.ns(example.2D, lims = rbind(c(0, 1), c(0, 1)), R = 0.5)
## Carrying out bootstrap.
fit <- boot.palm(fit, N = 100)
## Calculating 95% confidence intervals.
confint(fit)
## Estimates are very imprecise---these data were only used as
## they can be fitted and bootstrapped quickly for example purposes.
```

example.1D *1-dimensional example data*

Description

Simulated data from a Neyman-Scott point process, with children points generated in the interval [0, 1]. The number of children spawned by each parent is from a Binomial(4, 0.5) distribution.

Usage

```
example.1D
```

Format

A matrix.

example.2D *2-dimensional example data*

Description

Simulated data from a Neyman-Scott point process, with children points generated on the unit square. The number of children spawned by each parent is from a Binomial(2, 0.5) distribution.

Usage

```
example.2D
```

Format

A matrix.

example.twocamera *Two-camera example data.*

Description

Simulated data from a two-camera aerial survey.

Usage

```
example.twocamera
```

Format

A list.

`fit.ns` *Fitting a Neyman-Scott point process model*

Description

Estimates parameters for a Neyman-Scott point process by maximising the Palm likelihood. This approach was first proposed by Tanaka et al. (2008) for two-dimensional Thomas processes. Further generalisations were made by Stevenson, Borchers, and Fewster (2019) and Jones-Todd et al. (2019).

Usage

```
fit.ns(
  points,
  lims,
  R,
  disp = "gaussian",
  child.dist = "pois",
  child.info = NULL,
  sibling.list = NULL,
  edge.correction = "pbc",
  start = NULL,
  bounds = NULL,
  use.bobyqa = FALSE,
  trace = FALSE
)
```

Arguments

`points`	A matrix or list of matrices containing locations of observed points, where each row corresponds to a point and each column corresponds to a dimension. If a list, then the patterns are assumed to be independent and a single process is fitted to all.
`lims`	A matrix or list of matrices with two columns, corresponding to the upper and lower limits of each dimension, respectively. If a list, then each matrix provides the limits for the corresponding pattern in `points`.
`R`	Truncation distance for the difference process.
`disp`	A character string indicating the distribution of children around their parents. Use `"gaussian"` for multivariate normal dispersion with standard deviation `sigma`, or `"uniform"` for uniform dispersion within distance `tau` of the parent.
`child.dist`	The distribution of the number of children generated by a randomly selected parent. For a Poisson distribution, use `"pois"`; for a binomial distribution, use `"binomx"`, where `"x"` is replaced by the fixed value of the number of independent trials (e.g., `"binom5"` for a Binomial(5, p) distribution, and `"binom50"` for a Binomial(50, p) distribution); and `"twocamera"` for a child distribution appropriate for a two-camera aerial survey.

`child.info` A list of further information that is required about the distribution for the number of children generated by parents. See 'Details'.

`sibling.list` An optional list that comprises (i) a component named `sibling.mat`, containing a matrix such that the jth entry in the ith row is `TRUE` if the ith and jth points are known siblings, `FALSE` if they are known nonsiblings, and `NA` if their sibling status is not known; (ii) alpha, providing the probability that a sibling is successfully identified as a sibling; and (iii) beta, providing the probability that a nonsibling is successfully identified as a nonsibling. For multi-pattern fitting, this object must be a list of such lists, one for each pattern.

`edge.correction` The method used for the correction of edge effects. Either `"pbc"` for periodic boundary conditions, or `"buffer"` for a buffer-zone correction.

`start` A named vector of starting values for the model parameters.

`bounds` A list with named components. Each component should be a vector of length two, giving the upper and lower bounds for the named parameter.

`use.bobyqa` Logical; if `TRUE` the bobyqa function is used for optimisation. Otherwise the nlminb function is used. Note that bobyqa seems to be less stable than nlminb, but does not require calculation of the Palm likelihood's partial derivatives.

`trace` Logical; if `TRUE`, parameter values are printed to the screen for each iteration of the optimisation procedure.

Details

The parameter `D` is the density of parent points, which is always estimated. Possible additional parameters are

- `lambda`, the expected number of children generated per parent (when `child.dist = "pois"`).
- `p`, the proportion of the `x` possible children that are generated (when `child.dist = "binomx"`).
- `kappa`, the average length of the surface phase of a diving cetacean (when `child.dist = "twocamera"`; see Stevenson, Borchers, and Fewster, 2019).
- `sigma`, the standard deviation of dispersion along each dimension (when `disp` = "gaussian").
- `tau`, the maximum distance a child can be from its parent (when `disp` = "uniform").

The `"child.info"` argument is required when `child.dist` is set to `"twocamera"`. It must be a list that comprises (i) a component named `w`, providing the halfwidth of the detection zone; (ii) a component named `b`, providing the halfwidth of the survey area; (iii) a component named `l`, providing the time lag between cameras (in seconds); and (iv) a component named `tau`, providing the mean dive-cycle duration. See Stevenson, Borchers, and Fewster (2019) for details.

Value

An R6 reference class object.

References

Jones-Todd, C. M., Caie, P., Illian, J. B., Stevenson, B. C., Savage, A., Harrison, D. J., and Bown, J. L. (in press). Identifying prognostic structural features in tissue sections of colon cancer patients using point pattern analysis. *Statistics in Medicine*, **38**: 1421–1441.

Stevenson, B. C., Borchers, D. L., and Fewster, R. M. (2019) Cluster capture-recapture to account for identification uncertainty on aerial surveys of animal populations. *Biometrics*, **75**: 326–336.

Tanaka, U., Ogata, Y., and Stoyan, D. (2008) Parameter estimation and model selection for Neyman-Scott point processes. *Biometrical Journal*, **50**: 43–57.

See Also

Use coef.palm to extract estimated parameters, and plot.palm to plot the estimated Palm intensity function. Use boot.palm to run a parametric bootstrap, allowing calculation of standard errors and confidence intervals.

See sim.ns to simulate from a Neyman-Scott point process.

Examples

```
## Fitting model to example data.
fit <- fit.ns(example.2D, lims = rbind(c(0, 1), c(0, 1)), R = 0.5)
## Printing estimates.
coef(fit)
## Plotting the estimated Palm intensity.
plot(fit)
## Not run:
## Simulating data and fitting additional models.
set.seed(1234)
## One-dimensional Thomas process.
data.thomas <- sim.ns(c(D = 10, lambda = 5, sigma = 0.025), lims = rbind(c(0, 1)))
## Fitting a model to these data.
fit.thomas <- fit.ns(data.thomas$points, lims = rbind(c(0, 1)), R = 0.5)
## Three-dimensional Matern process.
data.matern <- sim.ns(c(D = 10, lambda = 10, tau = 0.1), disp = "uniform",
                      lims = rbind(c(0, 1), c(0, 2), c(0, 3)))
## Fitting a model to these data.
fit.matern <- fit.ns(data.matern$points, lims = rbind(c(0, 1), c(0, 2), c(0, 3)),
                     R = 0.5, disp = "uniform")

## End(Not run)
```

---

fit.twocamera *Estimation of animal density from two-camera surveys.*

Description

Estimates animal density (amongst other parameters) from two-camera aerial surveys. This conceptualises sighting locations as a Neyman-Scott point pattern.

Usage

```
fit.twocamera(
  points,
  cameras = NULL,
  d,
  w,
  b,
  l,
  tau,
  R,
  edge.correction = "pbc",
  start = NULL,
  bounds = NULL,
  trace = FALSE
)
```

Arguments

points	A vector (or single-column matrix) containing the distance along the transect that each detection was made.
cameras	An optional vector containing the camera ID (either 1 or 2) that made the corresponding detection in points.
d	The length of the transect flown (in km).
w	The distance from the transect to which detection of individuals on the surface is certain. This is equivalent to the half-width of the detection zone.
b	The distance from the transect to the edge of the area of interest. Conceptually, the distance between the transect and the furthest distance a whale could be on the passing of the first camera and plausibly move into the detection zone by the passing of the second camera.
l	The lag between cameras (in seconds).
tau	Mean dive-cycle duration (in seconds).
R	Truncation distance (see fit.ns).
edge.correction	The method used for the correction of edge effects. Either "pbc" for periodic boundary conditions, or "buffer" for a buffer-zone correction.
start	A named vector of starting values for the model parameters.
bounds	A list with named components. Each component should be a vector of length two, giving the upper and lower bounds for the named parameter.
trace	Logical; if TRUE, parameter values are printed to the screen for each iteration of the optimisation procedure.

Details

This function is simply a wrapper for fit.ns, and facilitates the fitting of the model proposed by Stevenson, Borchers, and Fewster (2019). This function presents the parameter D.2D (two-dimensional cetacean density in cetaceans per square km) rather than D for enhanced interpretability.

For further details on the cluster capture-recapture estimation approach, see Fewster, Stevenson and Borchers (2016).

Value

An R6 reference class object.

References

Fewster, R. M., Stevenson, B. C., and Borchers, D. L. (2016) Trace-contrast methods for capture-recapture without capture histories. *Statistical Science*, **31**: 245–258.

Stevenson, B. C., Borchers, D. L., and Fewster, R. M. (2019) Cluster capture-recapture to account for identification uncertainty on aerial surveys of animal populations. *Biometrics*, **75**: 326–336.

See Also

Use coef.palm to extract estimated parameters, and plot.palm to plot the estimated Palm intensity function. Use boot.palm to run a parametric bootstrap, allowing calculation of standard errors and confidence intervals.

See sim.twocamera to simulate sightings from a two-camera aerial survey.

Examples

```
## Fitting model.
fit <- fit.twocamera(points = example.twocamera$points, cameras = example.twocamera$cameras,
                     d = 500, w = 0.175, b = 0.5, l = 20, tau = 110, R = 1)
## Printing estimates.
coef(fit)
## Plotting the estimated Palm intensity.
plot(fit)
```

---

fit.void *Fitting a model to a void point process*

Description

Estimates parameters for a void point process by maximising the Palm likelihood. This approach was first proposed by Tanaka et al. (2008) for two-dimensional Thomas processes. Generalisation to d-dimensional void processes was made by Jones-Todd et al. (in press).

Usage

```
fit.void(
  points,
  lims,
  R,
  edge.correction = "pbc",
```

```
  start = NULL,
  bounds = NULL,
  use.bobyqa = FALSE,
  trace = FALSE
)
```

Arguments

points	A matrix or list of matrices containing locations of observed points, where each row corresponds to a point and each column corresponds to a dimension. If a list, then the patterns are assumed to be independent and a single process is fitted to all.
lims	A matrix or list of matrices with two columns, corresponding to the upper and lower limits of each dimension, respectively. If a list, then each matrix provides the limits for the corresponding pattern in points.
R	Truncation distance for the difference process.
edge.correction	The method used for the correction of edge effects. Either "pbc" for periodic boundary conditions, or "buffer" for a buffer-zone correction.
start	A named vector of starting values for the model parameters.
bounds	A list with named components. Each component should be a vector of length two, giving the upper and lower bounds for the named parameter.
use.bobyqa	Logical; if TRUE the bobyqa function is used for optimisation. Otherwise the nlminb function is used. Note that bobyqa seems to be less stable than nlminb, but does not require calculation of the Palm likelihood's partial derivatives.
trace	Logical; if TRUE, parameter values are printed to the screen for each iteration of the optimisation procedure.

Details

Parameters to estimate are as follows:

- Dc, the baseline density of points prior to the deletion process.
- Dp, the density of unobserved parents that cause voids.
- tau, the radius of the deletion process centred at each parent.

Value

An R6 reference class object.

References

Jones-Todd, C. M., Caie, P., Illian, J. B., Stevenson, B. C., Savage, A., Harrison, D. J., and Bown, J. L. (in press). Identifying prognostic structural features in tissue sections of colon cancer patients using point pattern analysis. *Statistics in Medicine*, **38**: 1421–1441.

Tanaka, U., Ogata, Y., and Stoyan, D. (2008) Parameter estimation and model selection for Neyman-Scott point processes. *Biometrical Journal*, **50**: 43–57.

See Also

Use coef.palm to extract estimated parameters, and plot.palm to plot the estimated Palm intensity function. Use boot.palm to run a parametric bootstrap, allowing calculation of standard errors and confidence intervals.

See sim.void to simulate from a void process.

Examples

```
## Not run:
set.seed(1234)
## Simulating a two-dimensional void process.
void.data <- sim.void(c(Dc = 1000, Dp = 10, tau = 0.05), rbind(c(0, 1), c(0, 1)))
## Fitting model.
fit <- fit.void(void.data$points, rbind(c(0, 1), c(0, 1)), R = 0.5)

## End(Not run)
```

`plot.palm` *Plotting an estimated Palm intensity function.*

Description

Plots a fitted Palm intensity function from an object returned by fit.ns.

Usage

```
## S3 method for class 'palm'
plot(x, xlim = NULL, ylim = NULL, show.empirical = TRUE, breaks = 50, ...)
```

Arguments

`x`	A fitted model from fit.ns.
`xlim`	Numeric vector giving the x-coordinate range.
`ylim`	Numeric vector giving the y-coordinate range.
`show.empirical`	Logical, if `TRUE` the empirical Palm intensity is also plotted.
`breaks`	The number of breakpoints between cells for the empirical Palm intensity.
`...`	Other parameters (for S3 generic compatibility).

Examples

```
## Fit model.
fit <- fit.ns(example.2D, lims = rbind(c(0, 1), c(0, 1)), R = 0.5)
## Plot fitted Palm intensity.
plot(fit)
```

porpoise.data *Two-camera porpoise data.*

Description

Synthetic data constructed from circle-back aerial survey data; see Stevenson, Borchers, and Fewster (2019).

Usage

```
porpoise.data
```

Format

A list.

References

Stevenson, B. C., Borchers, D. L., and Fewster, R. M. (2019) Cluster capture-recapture to account for identification uncertainty on aerial surveys of animal populations. *Biometrics*, **75**: 326–336.

sim.ns *Simulating points from a Neyman-Scott point process*

Description

Generates points from a Neyman-Scott point process using parameters provided by the user.

Usage

```
sim.ns(
  pars,
  lims,
  disp = "gaussian",
  child.dist = "pois",
  parents = NULL,
  child.info = NULL
)
```

Arguments

pars	A named vector containing the values of the parameters of the process that generates the points.
lims	A matrix or list of matrices with two columns, corresponding to the upper and lower limits of each dimension, respectively. If a list, then each matrix provides the limits for the corresponding pattern in points.
disp	A character string indicating the distribution of children around their parents. Use "gaussian" for multivariate normal dispersion with standard deviation sigma, or "uniform" for uniform dispersion within distance tau of the parent.
child.dist	The distribution of the number of children generated by a randomly selected parent. For a Poisson distribution, use "pois"; for a binomial distribution, use "binomx", where "x" is replaced by the fixed value of the number of independent trials (e.g., "binom5" for a Binomial(5, p) distribution, and "binom50" for a Binomial(50, p) distribution); and "twocamera" for a child distribution appropriate for a two-camera aerial survey.
parents	An optional matrix containing locations of parents. If this is provided, then the parameter D is not required in pars. If this is not provided, then parents are generated from a homogeneous Poisson point process with intensity D.
child.info	A list of further information that is required about the distribution for the number of children generated by parents. See 'Details'.

Details

For a list of possible parameter names, see fit.ns.

The "child.info" argument is required when child.dist is set to "twocamera". It must be a list that comprises (i) a component named w, providing the halfwidth of the detection zone; (ii) a component named b, providing the halfwidth of the survey area; (iii) a component named l, providing the time lag between cameras (in seconds); and (iv) a component named tau, providing the mean dive-cycle duration. See Stevenson, Borchers, and Fewster (2019) for details.

Value

A list. The first component gives the Cartesian coordinates of the generated points. The second component returns the parent locations. A third component may provide sibling information.

References

Stevenson, B. C., Borchers, D. L., and Fewster, R. M. (2019) Cluster capture-recapture to account for identification uncertainty on aerial surveys of animal populations. *Biometrics*, **75**: 326–336.

Examples

```
## Simulating from a one-dimensional Thomas process.
data.thomas <- sim.ns(c(D = 10, lambda = 5, sigma = 0.025), lims = rbind(c(0, 1)))
## Simulating from a three-dimensional Matern process.
data.matern <- sim.ns(c(D = 10, lambda = 10, tau = 0.1), disp = "uniform",
                      lims = rbind(c(0, 1), c(0, 2), c(0, 3)))
```

`sim.twocamera` *Simulating data from two-camera aerial surveys.*

Description

Simulating data from two-camera aerial surveys.

Usage

```
sim.twocamera(pars, d, w, b, l, tau, parents = NULL)
```

Arguments

`pars`	A vector containing elements named `D.2D`, `kappa`, and `sigma`, providing values of animal density (animals per square km), average duration of surface phase (s), and dispersion (km).
`d`	The length of the transect flown (in km).
`w`	The distance from the transect to which detection of individuals on the surface is certain. This is equivalent to the half-width of the detection zone.
`b`	The distance from the transect to the edge of the area of interest. Conceptually, the distance between the transect and the furthest distance a whale could be on the passing of the first camera and plausibly move into the detection zone by the passing of the second camera.
`l`	The lag between cameras (in seconds).
`tau`	Mean dive-cycle duration (in seconds).
`parents`	An optional vector containing the parent locations for all animals within the area of interest, given in distance along the transect (in km). If this is provided, then the parameter `D.2D` is not required in `pars`. If this is not provided, then parent locations are generated from a homogeneous Poisson point process with intensity `D.2D`.

Value

A list. The first component gives the distance along the transect of detected individuals. The second gives the parent locations. The third identifies which parent location generated each detected individual. The fourth gives the distance from the transect centre line of the detection location. The fifth provides observed sibling information.

Examples

```
twocamera.data <- sim.twocamera(c(D.2D = 1.3, kappa = 27, sigma = 0.02), d = 500,
                                w = 0.175, b = 0.5, l = 20, tau = 110)
```

`sim.void` *Simulating points from a void point process.*

Description

Generates points from a void point process using parameters provided by the user.

Usage

```
sim.void(pars, lims, parents = NULL)
```

Arguments

`pars`	A named vector containing the values of the parameters of the process that generates the points.
`lims`	A matrix or list of matrices with two columns, corresponding to the upper and lower limits of each dimension, respectively. If a list, then each matrix provides the limits for the corresponding pattern in `points`.
`parents`	An optional matrix containing locations of parents. If this is provided, then the parameter `D` is not required in `pars`. If this is not provided, then parents are generated from a homogeneous Poisson point process with intensity `Dp`.

Details

For a list of possible parameter names, see fit.ns.

Value

A list. The first component gives the Cartesian coordinates of the generated points. The second component returns the parent locations.

Examples

```
## Two-dimensional void process.
void.data <- sim.void(c(Dc = 1000, Dp = 10, tau = 0.05), rbind(c(0, 1), c(0, 1)))
## Plotting the data.
plot(void.data$points)
points(void.data$parents, pch = 16, col = "red")
```

`summary.palm` *Summarising palm model fits*

Description

Provides a useful summary of the model fit.

Usage

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

Arguments

`object`	A fitted model returned by fit.ns.
`...`	Other parameters (for S3 generic compatibility).

Index