Package 'foieGras'

July 22, 2020

Title Fit Continuous-Time State-Space and Latent Variable Models for Quality Control of Argos Satellite (and Other) Telemetry Data and for Estimating Movement Behaviour

Version 0.6-9

Date 2020-07-19

Author Ian Jonsen [aut, cre, cph], Toby Patterson [aut, ctb]

Maintainer Ian Jonsen <ian.jonsen@mq.edu.au>

Description Fits continuous-time random walk and correlated random walk state-space models for quality control animal tracking data ('Argos', processed light-level 'geolocation', 'GPS'). Template Model Builder ('TMB') is used for fast estimation. The 'Argos' data can be: (older) least squares-based locations; (newer) Kalman filter-based locations with error ellipse information; or a mixture of both. The models estimate two sets of location states corresponding to: 1) each observation, which are (usually) irregularly timed; and 2) user-specified time intervals (regular or irregular). Latent variable models are provided to estimate move persistence along tracks as an index of behaviour. 'Jonsen I', 'McMahon CR', 'Patterson TA', 'Auger-Méthé M', 'Harcourt R', 'Hindell MA', 'Bestley S' (2019) Movement responses to environment: fast inference of variation among southern elephant seals with a mixed effects model. Ecology 100:e02566 <doi:10.1002/ecy.2566>.

URL <https://cran.r-project.org/package=foieGras>

BugReports https://github.com/ianjonsen/foieGras/issues

License MIT + file LICENSE

LazyData true

Encoding UTF-8

RoxygenNote 7.1.1

LinkingTo TMB (>= 1.7.15), RcppEigen

Imports tibble (>= 2.1.3), ggplot2 (>= 3.0.0), lubridate, TMB (>= 1.7.15), sf (>= 0.9-4), stringr, tidyr, future (>= 1.13.0), furrr (>= 0.1.0), rworldmap, parallel, purrr, dplyr (>= 1.0.0), trip, assertthat, wesanderson, patchwork

Suggests testthat, covr, knitr, rmarkdown, rgeos

Depends R (>= 3.5.0)

SystemRequirements C++11, GDAL (>= 2.4.2), GEOS (>= 3.7.0), PROJ (>= 5.2.0)

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-07-22 21:50:02 UTC

R topics documented:

`foieGras-package` **foieGras**

Description

fit Continuous-Time Random Walk and Correlated Random Walk state-space models to filter Argos Least Squares or Kalman Filter location data

Author(s)

Ian Jonsen, Toby Patterson

References

Jonsen ID, Patterson TA, Costa DP, et al. (2020) A continuous-time state-space model for rapid quality-control of Argos locations from animal-borne tags. Movement Ecology 8:31 https://doi.org/10.1186/s40462-020-00217-7

Jonsen ID, McMahon CR, Patterson TA, et al. (2019) Movement responses to environment: fast inference of variation among southern elephant seals with a mixed effects model. Ecology. 100(1):e02566 https://doi.org/10.1002/ecy.2566

See Also

fit_ssm

---

dummy *Roxygen commands*

Description

Roxygen commands

Usage

```
dummy()
```

---

ellie *Southern elephant seal Argos satellite data (1 individual)*

Description

Example elephant seal Argos tracking data. Data were sourced from the Integrated Marine Observing System (IMOS) - IMOS is supported by the Australian Government through the National Collaborative Research Infrastructure Strategy and the Super Science Initiative.

Format

.RData

ellies *Southern elephant seal Argos satellite data (2 individuals)*

Description

Example elephant seal Argos tracking data. Data were sourced from the Integrated Marine Observing System (IMOS) - IMOS is supported by the Australian Government through the National Collaborative Research Infrastructure Strategy and the Super Science Initiative.

Format

.RData

emf *emf*

Description

emf

Usage

```
emf(
  gps = 0.1,
  emf.x = c(1, 1.54, 3.72, 13.51, 23.9, 44.22),
  emf.y = c(1, 1.29, 2.55, 14.99, 22, 32.53)
)
```

Arguments

gps	error multiplication factor(s) for GPS locations, can be a scalar (x = y) or vector of length 2 (x != y)
emf.x	error multiplication factors for Argos longitude classes 3, 2, 1, 0, A, B (Z assumed equal to B)
emf.y	error multiplication factors for Argos latitude classes 3, 2, 1, 0, A, B (Z assumed equal to B)

Details

Error Multiplication Factors for Argos (and GPS) locations. Default assumption is that GPS locations are 10x more accurate than Argos lc 3 in both x and y directions.

User-specified Error Multiplication Factors (emf). emf's must be provided as a data.frame with the following columns:

emf.x emf values for the x direction

emf.y emf values for y direction

lc location class designations

The location class designations can be the standard Argos lc values: 3, 2, 1, 0, A, B, Z or other values. The number of classes specified is flexible though may not be amenable to a large number of classes. Whatever class designations are chosen must also appear in the input data lc column. A GPS location class ("G") is provided by default and assumes that GPS locations are 10 x more precise than Argos lc 3 locations.

---

fit_mpm *fit a a Move Persistence Model (mpm)*

Description

fit a random walk with time-varying move persistence to location data (e.g., output from fit_ssm)

Usage

```
fit_mpm(
  x,
  model = c("mpm", "jmpm"),
  optim = "optim",
  optMeth = "L-BFGS-B",
  verbose = 1,
  control = NULL,
  inner.control = NULL
)
```

Arguments

x	a data frame of observations (see details)
model	mpm model to fit; either mpm with unpooled random walk variance parameters (sigma_(g,i)) or jmpm with a single, pooled random variance parameter (sigma_g)
optim	numerical optimizer
optMeth	optimization method to use (default is "L-BFGS-B"), ignored if optim = "nlminb" (see ?optim for details)
verbose	report progress during minimization
control	list of control parameters for the outer optimization (type ?nlminb or ?optim for details)
inner.control	list of control parameters for the inner optimization

Value

a list with components

fitted	a dataframe of fitted locations
par	model parameter summary
data	input dataframe
tmb	the tmb object
opt	the object returned by the optimizer

Examples

```
## fit jmpm to two southern elephant seals
data(xs)
dmp <- grab(xs, "predicted", as_sf=FALSE)
dmp <- dmp[, c("id", "date", "lon", "lat")]
fmpm <- fit_mpm(dmp, model = "jmpm")
```

---

fit_ssm *Fit a continuous-time state-space model to filter Argos satellite geolocation data*

Description

fits either a simple random walk or a correlated random walk (a random walk on velocity) in continuous time to filter Argos LS, and/or KF/KS location data, processed light-level geolocation data (GLS), and/or GPS data. Location data of different types can combined in a single data frame (see details). Predicts locations at user-specified time intervals (regular or irregular).

Usage

```
fit_ssm(
  d,
  vmax = 5,
  ang = c(15, 25),
  distlim = c(2500, 5000),
  spdf = TRUE,
  min.dt = 60,
  pf = FALSE,
  model = "crw",
  time.step = 6,
  scale = FALSE,
  emf = NULL,
  map = NULL,
  parameters = NULL,
```

```
  fit.to.subset = TRUE,
  optim = "optim",
  optMeth = "L-BFGS-B",
  verbose = 1,
  control = NULL,
  inner.control = NULL,
  lpsi = -Inf
)
```

Arguments

d	a data frame of observations including Argos KF error ellipse info (when present)
vmax	max travel rate (m/s) passed to sda to identify outlier locations
ang	angles (deg) of outlier location "spikes"
distlim	lengths (m) of outlier location "spikes"
spdf	(logical) turn trip::sda on (default; TRUE) or off
min.dt	minimum allowable time difference between observations; dt <= min.dt will be ignored by the SSM
pf	just pre-filter the data, do not fit the SSM (default is FALSE)
model	fit either a simple random walk ("rw") or correlated random walk ("crw") as a continuous-time process model
time.step	options: 1) the regular time interval, in hours, to predict to; 2) a vector of prediction times, possibly not regular, must be specified as a data.frame with id and POSIXt dates; 3) NA - turns off prediction and locations are only estimated at observation times.
scale	scale location data for more efficient optimization. This should rarely be needed (default = FALSE)
emf	optionally supplied data.frame of error multiplication factors for Argos location quality classes. Default behaviour is to use the factors supplied in foieGras::emf()
map	a named list of parameters as factors that are to be fixed during estimation, e.g., list(psi = factor(NA))
parameters	a list of initial values for all model parameters and unobserved states, default is to let sfilter specify these. Only play with this if you know what you are doing
fit.to.subset	fit the SSM to the data subset determined by prefilter (default is TRUE)
optim	numerical optimizer to be used ("nlminb" or "optim")
optMeth	optimization method to use (default is "L-BFGS-B"), ignored if optim = "nlminb" (see ?optim for details)
verbose	report progress during minimization; 0 for complete silence; 1 for parameter trace; 2 for optimizer trace
control	list of control settings for the outer optimizer (see nlminb or optim for details)
inner.control	list of control settings for the inner optimizer (see MakeADFun for additional details)
lpsi	lower bound for the psi parameter

Details

d is a data.frame, tibble, or sf-tibble with 5, 7 or 8 columns, depending on the tracking data type. Argos Least-Squares and GPS data should have 5 columns in the following order: "id", "date", "lc", "lon", "lat". Where "date" can be a POSIX object or text string in YYYY-MM-DD HH:MM:SS format. If a text string is supplied then the time zone is assumed to be "GMT". lc (location class) can include the following values: 3, 2, 1, 0, A, B, Z, G, or GL. The latter two are for GPS and GLS locations, respectively. Class Z values are assumed to have the same error variances as class B. By default, class G (GPS) locations are assumed to have error variances 10x smaller than Argos class 3 variances, but unlike Argos error variances the GPS variances are the same for longitude and latitude.

See emf for details on how to modify these assumptions.

Argos Kalman Filter (or Kalman Smoother) data should have 8 columns, including the above 5 plus "smaj", "smin", "eor" that contain Argos error ellipse variables (in m for "smaj", "smin" and deg for "eor").

Light-level geolocation (GLS) locations can be modelled provided each longitude and latitude has a corresponding standard error. These data should have 7 columns, including the above 5 plus "lonerr", "laterr" (in degrees). In this case, all lc values should be set to "GL".

Multiple location data types can be combined in a single data frame (see the vignette for examples).

When data are provided as an sf-tibble, the user-specified projection is respected. Otherwise, longlat data are re-projected internally to a global Mercator grid and provided as the default output. An un-projected tibble of lon,lat and x,y location estimates can be obtained by using grab with the argument as_sf = FALSE.

Value

a list with components

call	the matched call
predicted	an sf tbl of predicted location states
fitted	an sf tbl of fitted locations
par	model parameter summary
data	an augmented sf tbl of the input data
inits	a list of initial values
pm	the process model fit, either "rw" or "crw"
ts	time time.step in h used
opt	the object returned by the optimizer
tmb	the TMB object
rep	TMB sdreport
aic	the calculated Akaike Information Criterion
time	the processing time for sfilter

References

Jonsen ID, Patterson TA, Costa DP, et al. (2020) A continuous-time state-space model for rapid quality-control of Argos locations from animal-borne tags. Movement Ecology 8:31 https://doi.org/10.1186/s40462-020-00217-7

Jonsen ID, McMahon CR, Patterson TA, et al. (2019) Movement responses to environment: fast inference of variation among southern elephant seals with a mixed effects model. Ecology. 100(1):e02566 https://doi.org/10.1002/ecy.2566

See Also

sfilter

Examples

```
## fit crw model to two seals with Argos LS data
data(ellies)
fit <- fit_ssm(ellies, vmax = 5, model = "crw", time.step = 48)

## time series plots of fitted value fits to both seals
plot(fit, what = "fitted", type = 1)

## track plots of predicted value fits for both seals
plot(fit, what = "predicted", type = 2)
```

`fmap` *fmap*

Description

map foieGras fitted or predicted locations, with or without Argos observations, optionally apply a different projection

Usage

```
fmap(
  x,
  y = NULL,
  what = c("fitted", "predicted"),
  conf = TRUE,
  obs = FALSE,
  by.date = FALSE,
  crs = NULL,
  ext.rng = c(0.05, 0.05),
  size = 0.25,
  col = "black"
)
```

Arguments

x	a foieGras ssm fit object with class ‘fG_ssm‘
y	optionally, a foieGras mpm fit object with class ‘fG_mpm‘; default is NULL
what	specify which location estimates to map: fitted or predicted
conf	include confidence regions around estimated location (logigal; default = TRUE, unless y is an mpm fit object then conf is FALSE)
obs	include Argos observations on map (logical; default = FALSE)
by.date	when mapping single tracks, should locations be coloured by date (logical; default = FALSE)
crs	‘proj4string‘ for re-projecting locations, if NULL the default projection ("+proj=merc") for the fitting the SSM will be used
ext.rng	factors to extend the plot range in x and y dimensions (can exceed 1)
size	size of estimated location points; optionally a vector of length 2, with size of observed locations given by 2nd value (ignored if obs = FALSE)
col	colour of observed locations (ignored if obs = FALSE)

---

grab *grab tibble’s by name from a foieGras model object*

Description

‘grab()‘ lets you obtain ‘fitted‘, ‘predicted‘, or ‘data‘ tibble’s from a compound tibble created when fitting to multiple individual data sets. The specified tibble’s are appended to a single output tibble.

Usage

```
grab(x, what = "fitted", as_sf = TRUE)
```

Arguments

x	a foieGras ssm or mpm model object
what	the tibble to be grabbed; either ‘fitted‘, ‘predicted‘ (ssm only), or ‘data‘ (single letters can be used)
as_sf	logical; if FALSE then return a tibble with un-projected lonlat coordinates, otherwise return an sf tibble. Ignored if x is an mpm model object.

Value

a tibble with all individual tibble’s appended

Examples

```
## load example foieGras fit object (to save time)
data(xs)
## grab predicted values as an un-projected tibble
preds <- grab(xs, what = "predicted", as_sf = FALSE)
```

join *join an mpm-estimated behavioural index to ssm-predicted locations*

Description

'join()' joins ssm-predicted locations and mpm-estimated behavioural index into a single tibble. If the ssm-predicted tibble is a projected sf object then the output of join will also be an sf object (default). This can be avoided by using 'as_sf = FALSE'.

Usage

```
join(ssm, mpm, what.ssm = "predicted", as_sf = TRUE)
```

Arguments

ssm	a foieGras ssm fitted model object
mpm	a foieGras mpm fitted model object
what.ssm	specifies whether ssm 'predicted' or 'fitted' values are to be extracted
as_sf	logical; if FALSE then return a tibble with un-projected lonlat coordinates, otherwise return an sf tibble

Value

a single tbl with all individuals

Examples

```
## load example foieGras fit objects (to save time)
data(xs)
data(xm)
## join predicted values as an un-projected tibble
xsm <- join(xs, xm, as_sf = FALSE)
xsm
```

osar *calculate one-step-ahead (prediction) residuals from a* foieGras *fit*

Description

calculate one-step-ahead (prediction) residuals from a foieGras fit

Usage

```
osar(x, method = "fullGaussian", ...)
```

Arguments

x	a compound fG tbl fit object
method	method to calculate prediction residuals (default is "oneStepGaussianOffMode"; see '?TMB::oneStepPrediction' for details)
...	other arguments to TMB::oneStepPrediction

Details

One-step-ahead residuals are useful for assessing goodness-of-fit in latent variable models. This is a wrapper function for TMB::oneStepPredict (beta version). osar tries the "fullGaussian" (fastest) method first and falls back to the "oneStepGaussianOffMode" (slower) method for any failures. Subsequent failures are dropped from the output and a warning message is given. Note, OSA residuals can take a considerable time to calculate if there are many individual fits and/or deployments are long. The method is automatically parallelized across 2 x the number of individual fits, up to the number of processor cores available.

References

Thygesen, U. H., C. M. Albertsen, C. W. Berg, K. Kristensen, and A. Neilsen. 2017. Validation of ecological state space models using the Laplace approximation. Environmental and Ecological Statistics 24:317–339.

Examples

```
## see summary fit output
## load example foieGras fit object (to save time)
data(xs)
d <- xs[1, ] ## just use the first seal to save time
dres <- osar(d)
```

---

plot.fG_mpm *plot*

Description

visualize fits from an fG_mpm object

Usage

```
## S3 method for class 'fG_mpm'
plot(x, y = NULL, pages = 1, asp = 0, ncol = 1, ...)
```

Arguments

x	a foieGras mpm fit object with class fG_mpm
y	optional ssm fit object with class fG_ssm corresponding to x. If absent, 1-d plots of gamma_t time series are rendered otherwise, 2-d track plots with locations coloured by gamma_t are rendered.
pages	plots of all individuals on a single page (pages = 1; default) or each individual on a separate page (pages = 0)
asp	used a fixed 1:1 aspect ratio for 2-d track plots (asp = 1), or allow aspect ratio to vary between plots (asp = 0; default). Ignored if y is NULL and/or pages = 0
ncol	number of columns to use for faceting. Default is ncol = 1 but this may be increased for multi-individual objects. Ignored if pages = 0
...	additional arguments to be ignored

Value

a ggplot object with either: 1-d time series of gamma_t estimates (if y not provided), with estimation uncertainty ribbons (95 or 2-d track plots (if y provided) coloured by gamma_t, with smaller points having greater uncertainty (size is proportional to SE^-2). Plots can be rendered all on a single page (pages = 1) or on separate pages.

Examples

```
# plot mpm fit object
# 1-d time-series plots
plot(xm)
# 2-d track plots by adding ssm fit object
plot(xm, xs)
```

---

plot.fG_osar *plot*

Description

plot One-Step-Ahead (prediction) residuals from a foieGras osar object

Usage

```
## S3 method for class 'fG_osar'
plot(x, type = c("qqnorm", "histogram"), bw = 0.5, ...)
```

Arguments

x	a foieGras osar object with class ‘fG_osar‘
type	type of residual plot to generate; either qqnorm (default) or histogram
bw	binwidth for histogram plots (see ggplot2::geom_histogram for details), ignored if type = "qqnorm"
...	additional arguments to be ignored

Examples

```
## load example osar output (to save time)
data(xs)
dres <- osar(xs[1, ]) # only use first seal to save time
plot(dres, type = "qq")
```

`plot.fG_ssm` *plot*

Description

visualize fits from an fG_ssm object

Usage

```
## S3 method for class 'fG_ssm'
plot(
  x,
  what = c("fitted", "predicted"),
  type = 1,
  outlier = TRUE,
  pages = 1,
  ncol = 2,
  ...
)
```

Arguments

`x`	a `foieGras` ssm fit object with class 'fG_ssm'
`what`	specify which location estimates to display on time-series plots: fitted or predicted
`type`	of plot to generate: 1-d time series for lon and lat separately (type = 1, default) or 2-d track plot (type = 2)
`outlier`	include outlier locations dropped by prefilter (outlier = TRUE, default)
`pages`	plots of all individuals on a single page (pages = 1; default) or each individual on a separate page (pages = 0)
`ncol`	number of columns to use for faceting. Default is ncol = 2 but this may be increased for multi-individual fit objects
`...`	additional arguments to be ignored

Value

a ggplot object with either: (type = 1) 1-d time series of fits to data, separated into x and y components (units = km) with prediction uncertainty ribbons (2 x SE); or (type = 2) 2-d fits to data (units = km)

Examples

```
## load example foieGras fit object (to save time)
data(xs)
plot(xs, what = "f", type = 1)
plot(xs, what = "p", type = 2)
```

---

print.ssm *print* foieGras *fit object summary information*

Description

print foieGras fit object summary information

Usage

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

Arguments

x	a foieGras ssm fit object
...	unused. For compatibility with the generic method.

Examples

```
## see summary fit output
## load example foieGras ssm fit object (to save time)
data(xs)
xs$ssm[[1]]
```

---

xm *foieGras example mpm fit object*

Description

Example foieGras mpm fit object. This example fit is included purely to speed up examples where a fit object is required but fitting to data is not the focus of the example.

Format

.RData

xs *foieGras example fit object*

Description

Example foieGras fit object. This example fit is included purely to speed up examples where a fit object is required but fitting to data is not the focus of the example.

Format

.RData

Index