Package 'foieGras'

October 8, 2019

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

Version 0.4.0

Date 2019-10-06

Author Ian Jonsen [aut, cre],
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 to filter 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-Methe 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 6.1.1

LinkingTo TMB (>= 1.7.15), RcppEigen

Imports dplyr (>= 0.8.0), tibble (>= 2.1.3), argosfilter, ggplot2 (>= 3.0.0), gridExtra, lubridate, TMB (>= 1.7.15), sf (>= 0.8-0), stringr, tidyr, future (>= 1.13.0), furrr (>= 0.1.0), rworldmap, parallel

Suggests testthat, covr, knitr, rmarkdown, units, rgeos

VignetteBuilder knitr

Depends R (>= 3.5.0)

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

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-10-07 22:10:03 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 I, McMahon CR, Patterson TA, Auger-Methe 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

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()
```

Details

Error Multiplication Factors for Argos (and GPS) locations. Called by `prefilter()`

Examples

```
emf()
```

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 = c("nlminb", "optim"),
  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
`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(fssm)
dmp <- grab(fssm, "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 KF and/or LS data, or processed light-level geolocation data. predicts locations at user-specified time intervals (regular or irregular).

Usage

```
fit_ssm(d, vmax = 50, ang = -1, distlim = c(2500, 5000),
  spdf = TRUE, min.dt = 60, pf = FALSE, model = "crw",
  time.step = 6, emf = NULL, map = NULL, parameters = NULL,
  fit.to.subset = TRUE, optim = "optim", verbose = 1,
  control = NULL, inner.control = NULL, lpsi = -Inf)
```

Arguments

d	a data frame of observations including Argos KF error ellipse info
vmax	max travel rate (m/s) passed to argosfilter::sdafilter to define outlier locations
ang	angles of outlier location "spikes" - see ?argosfilter::sdafilter for details
distlim	lengths of outlier location "spikes" - see ?argosfilter::sdafilter for details
spdf	(logical) turn argosfilter::sdafilter 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	the regular time interval, in hours, to predict to. Alternatively, a vector of prediction times, possibly not regular, must be specified as a data.frame with id and POSIXt dates.
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")
verbose	report progress during minimization; 0 for complete silence; 1 for progress bar only; 2 for minimizer trace but not progress bar
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 ?TMB::MakeADFUN for additional details)
lpsi	lower bound for the psi parameter

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

Examples

```
## fit rw model to one seal with Argos KF data
data(ellie)
fit <- fit_ssm(ellie, model = "rw", time.step = 24)

## time series plots of predicted value fits
plot(fit, what = "predicted", type = 1)

## fit crw model to both seals, with Argos KF & LS data
data(ellies)
fits <- fit_ssm(ellies, model = "crw", time.step = 24)

## track plots of fits for both seals
plot(fits, 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, what = c("fitted", "predicted"), obs = FALSE, crs = NULL,
  ext.rng = c(0.05, 0.05), size = 1)
```

Arguments

`x`	a foieGras fitted object
`what`	specify which location estimates to map: fitted or predicted
`obs`	include Argos observations on map (logical) (logical); ignored if ‘obs = FALSE‘
`crs`	‘proj4string‘ for reprojecting 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

fmpm *foieGras example mpm fit object*

Description

Example foieGras fit object, using ssm_fits example data and the following call: fmp <- fssm select(id,date,lon,lat) fit_mpm(., model="jmpm") 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

fssm *foieGras example fit object*

Description

Example foieGras fit object, using ellies example data and the following call: fssm <- fit_ssm(ellies, vmax=4, model="crw", time.step=36). 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

grab *grab tbl's by name from a foieGras model object*

Description

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

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 unprojected lonlat coordinates, otherwise return an sf tibble. Ignored if x is an mpm model object.

Value

a tbl with all individual tbl's appended

Examples

```
## load example foieGras fit object (to save time)
data(fssm)
## grab predicted values as an unprojected tibble
preds <- grab(fssm, 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, as_sf = TRUE)
```

Arguments

`ssm`	a `foieGras` ssm fitted model object
`mpm`	a `foieGras` mpm fitted model object
`as_sf`	logical; if FALSE then return a tibble with unprojected lonlat coordinates, otherwise return an sf tibble

Value

a single tbl with all individuals

Examples

```
## load example foieGras fit objects (to save time)
data(fssm)
data(fmpm)
## join predicted values as an unprojected tibble
fsmp <- join(fssm, fmpm, as_sf = FALSE)
fsmp
```

mpmf *fit the move persistence model to regularized location data*

Description

generates initial values for model parameters and unobserved gamma's; structures data and initial values for C++ TMB template; fits move persistence model; minimizes the joint log-likelihood via the selected optimizer (nlminb or optim); structures and passes output object to fit_mpm

Usage

```
mpmf(x, model = c("mpm", "jmpm"), optim = c("nlminb", "optim"),
  verbose = FALSE, control = NULL, inner.control = NULL)
```

Arguments

x	temporally regularized location data, eg. output from fit_ssm
model	specify whether MPM is to be fit with unpooled ("mpm") or pooled ("jmpm") RW variance(s).
optim	numerical optimizer to be used ("nlminb" or "optim")
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 settings for the inner optimization (see ?TMB::MakeADFUN for additional details)

Details

called by fit_mpm, see ?fit_mpm.

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(fssm)
fres <- osar(fssm)
plot(fres)
```

---

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(fssm)
fres <- osar(fssm)
plot(fres, "qq")
plot(fres, "hist")
```

---

plot.fG_ssm *plot*

Description

visualize multiple fits from an fG compound tibble

Usage

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

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)
ncol	number of columns to use for faceting. Default is ncol = 1 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(fssm)
plot(fssm, what = "f", type = 1)
plot(fssm, what = "p", type = 2)
```

`prefilter` *Prepare Argos data for fitting state-space model*

Description

`prefilter` (1) determines Argos data type (LS or KF); (2) converts dates to POSIXt & identifies observations with duplicate dates; (3) orders observations in time; (4) removes duplicate observations; (5) removes observations occurring within 60 s of one another (keeps first); (6) shifts longitudes that straddle -180,180 to 0,360 and vice-versa; (7) projects lonlat coords to mercator x,y coords (in km); (8) adds location error multiplication factors based on Argos location class (for type LS); (9) uses a `argosfilter::sdafilter` to identify potential outlier locations (by distance only) to be ignored when fitting the `ctrw` model

Usage

```
prefilter(data, vmax = 50, ang = -1, distlim = c(2500, 5000),
  spdf = TRUE, min.dt = 60, emf = NULL)
```

Arguments

`data`	input data - must have 5 (LS), or 8 (KF) columns (see details)
`vmax`	max travel rate (m/s) - see `?argosfilter::sdafilter` for details
`ang`	angles of outlier location "spikes" - see `?argosfilter::sdafilter` for details
`distlim`	lengths of outlier location "spikes" - see `?argosfilter::sdafilter` for details
`spdf`	turn speed filter on/off (logical; default is TRUE)
`min.dt`	minimum allowable time difference between observations; `dt < min.dt` will be ignored by the SSM
`emf`	optionally supplied data.frame of error multiplication factors for Argos location quality classes. see Details

Details

called by `fit_ssm`.

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.

Value

an sf object with all observations passed from `data` and the following appended columns

`keep`	logical indicating whether observation should be ignored by `sfilter` (FALSE)
`obs.type`	flag indicating whether KF or LS measurement model applies
`emf_x`	error multiplication factors for `x` direction
`emf_y`	error multiplication factors for `y` direction
`geometry`	sf POINT object giving `x,y` coordinates in km

Examples

```
data(ellie)
pf <- prefilter(ellie, vmax=4, ang=c(15,25), min.dt=120)
pf
```

---

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

Description

print `ssm` 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(fssm)
fssm$ssm[[1]]
```

sfilter *fit the state-space model to* prefilter*-ed data*

Description

generates initial values for model parameters and unobserved states; structures data and initial values for C++ TMB template; fits state-space model; minimises the joint log-likelihood via the selected optimizer (nlminb or optim); structures and passes output object to fit_ssm

Usage

```
sfilter(x, model = c("rw", "crw"), time.step = 6, parameters = NULL,
  map = NULL, fit.to.subset = TRUE, optim = c("nlminb", "optim"),
  verbose = FALSE, control = NULL, inner.control = NULL,
  lpsi = -10)
```

Arguments

x	Argos data passed through prefilter()
model	specify which SSM is to be fit: "rw" or "crw"
time.step	the regular time interval, in hours, to predict to. Alternatively, a vector of prediction times, possibly not regular, must be specified as a data.frame with id and POSIXt dates.
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...
map	a named list of parameters as factors that are to be fixed during estimation, e.g., list(psi = factor(NA))
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")
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 settings for the inner optimization (see ?TMB::MakeADFUN for additional details)
lpsi	lower bound for the psi parameter

Details

called by fit_ssm. sfilter can only fit to an individual track, use fit_ssm to fit to multiple tracks (see ?fit_ssm).

Examples

```
data(ellie)
pf <- prefilter(ellie, vmax=4, ang=c(15,25), min.dt=120)
out <- sfilter(pf, model="rw", time.step=24)
```

---

wrap_lon *wrap_lon*

Description

wrap_lon

Usage

```
wrap_lon(lon, lon_min = -180)
```

Arguments

lon	a vector of longitudes
lon_min	the minimum longitude value to wrap appropriately, eg. 0 to wrap -180, 180 on to 0, 360 and -180 to wrap 0,360 on to -180,180

Details

wrap longitudes from an arbitrary minimum

Examples

```
lon <- seq(-180,180)
lon1 <- wrap_lon(lon, 0)
range(lon)
range(lon1)
```

Index