Package 'threshr'

July 22, 2025

Title Threshold Selection and Uncertainty for Extreme Value Analysis

Version 1.0.6

Date 2024-07-17

Description Provides functions for the selection of thresholds for use in extreme value models, based mainly on the methodology in Northrop, Attalides and Jonathan (2017) <doi:10.1111/rssc.12159>. It also performs predictive inferences about future extreme values, based either on a single threshold or on a weighted average of inferences from multiple thresholds, using the 'revdbayes' package <https://cran.r-project.org/package=revdbayes>. At the moment only the case where the data can be treated as independent identically distributed observations is considered.

Imports graphics, methods, revdbayes (>= 1.3.4), rust (>= 1.2.2), stats

License GPL (>= 2)

LazyData TRUE

Encoding UTF-8

Depends R (>= 3.3.0)

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

URL https://paulnorthrop.github.io/threshr/, https://github.com/paulnorthrop/threshr

BugReports https://github.com/paulnorthrop/threshr/issues

Config/testthat/edition 3

NeedsCompilation no

Author Paul J. Northrop [aut, cre, cph], Nicolas Attalides [aut]

Maintainer Paul J. Northrop <p.northrop@ucl.ac.uk>

Repository CRAN

Date/Publication 2024-07-17 21:10:02 UTC

Contents

---

threshr-package   *threshr: Threshold Selection and Uncertainty for Extreme Value Analysis*

Description

Provides functions for the selection of extreme value threshold. At the moment only the simplest case, where the data can be treated as independent identically distributed observations, is considered. Future releases will tackle more general situations. See the 'threshr' website for more information, documentation and examples.

Details

The main function in the threshr package is `ithresh`, which uses leave-one-out cross-validation in a Bayesian setup to compare the predictive ability resulting from the use of each of a user-supplied set of thresholds.

See `vignette("threshr-vignette", package = "threshr")` for an overview of the package.

Author(s)

Maintainer: Paul J. Northrop <p.northrop@ucl.ac.uk> [copyright holder]

Authors:

- Nicolas Attalides

References

Northrop, P. J. (2017). revdbayes: Ratio-of-Uniforms Sampling for Bayesian Extreme Value Analysis. R package version 1.2.1. https://cran.r-project.org/package=revdbayes.

Northrop, P. J., Attalides, N. and Jonathan, P. (2017) Cross-validatory extreme value threshold selection and uncertainty with application to ocean storm severity. *Journal of the Royal Statistical Society Series C: Applied Statistics*, **66**(1), 93-120. doi:10.1111/rssc.12159

See Also

The packages revdbayes and rust.

ithresh for threshold selection in the i.i.d. case based on leave-one-out cross-validation.

---

gom   *Storm peak significant wave heights from the Gulf of Mexico*

Description

A numeric vector containing 315 hindcasts of storm peak significant wave heights, in metres, from 1900 to 2005 at an unnamed location in the Gulf of Mexico.

Usage

```
gom
```

Format

A vector containing 315 observations.

Source

Oceanweather Inc. (2005) GOMOS – Gulf of Mexico hindcast study.

References

Northrop, P. J., N. Attalides, and P. Jonathan. (2017). Cross-Validatory Extreme Value Threshold Selection and Uncertainty with Application to Ocean Storm Severity. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **66**(1), 93-120. doi:10.1111/rssc.12159.

---

ithresh   *Threshold selection in the i.i.d. case (peaks over threshold)*

Description

Produces a diagnostic plot to assist in the selection of an extreme value threshold in the case where the data can be treated as independent and identically distributed (i.i.d.) observations. For example, it could be that these observations are the cluster maxima resulting from the declustering of time series data. The predictive ability of models fitted using each of a user-supplied set of thresholds is assessed using leave-one-out cross-validation in a Bayesian setup. These models are based on a Generalized Pareto (GP) distribution for threshold excesses and a binomial model for the probability of threshold exceedance. See Northrop et al. (2017) for details.

Usage

```
ithresh(data, u_vec, ..., n_v = 1, npy = NULL, use_rcpp = TRUE)
```

Arguments

data
: A numeric vector of observations. Any missing values will be removed. The argument npy (see below) may be supplied as an attribute of data using attr(data, "npy") <- value, where value is the value of npy (see [attr](#)). If npy is supplied twice, as both attr(data, "npy")) and using the npy argument, then the former is used.

u_vec
: A numeric vector. A vector of *training* thresholds at which inferences are made from a binomial-GP model. These could be set at sample quantiles of data using [quantile](#). Any duplicated values will be removed.

...
: Further (optional) arguments to be passed to the [revdbayes](#) function [rpost_rcpp](#) (or [rpost](#)), which use the generalized ratio-of-uniforms method to simulate from extreme value posterior distributions. In particular:

  - n. The size of the posterior sample used to perform predictive inference. Default: n = 1000.
  - prior. A prior for GP parameters to be passed to the **revdbayes** function [set_prior](#). Can be either a character scalar that chooses an in-built prior, or a user-supplied R function or pointer to a compiled C++ function. See the [set_prior](#) documentation for details of the in-built priors. See the **revdbayes** vignette [Faster simulation using revdbayes](#) for information about creating a pointer to a C++ function. See also the **Examples** section. If the user supplies an R function then [rpost](#) will be used for posterior simulation, rather than (the faster) [rpost_rcpp](#), regardless of the input value of use_rcpp.
    Default: prior = "mdi" with a = 0.6 and min_xi = -1. This particular prior is studied in Northrop et al. (2017).
  - h_prior. A *list* of further arguments (hyperparameters) for the GP prior specified in prior.
  - bin_prior. A prior for the threshold exceedance probability p to be passed to the **revdbayes** function [set_bin_prior](#). Can either be a character scalar that chooses an in-built prior, or a user_supplied R function.
    Default: prior = "jeffreys", i.e. Beta(1/2, 1/2).
  - h_bin_prior. A *list* of further arguments (hyperparameters) for the binomial prior specified in bin_prior. See the [set_bin_prior](#) documentation for details of the in-built priors.
  - trans. A character scalar: either "none" or "BC". See [rpost_rcpp](#) for details. The default is "none", which is usually faster than "BC". However, if there are very few threshold excesses then using trans = "BC" can make the optimizations involved in the generalized ratio-of-uniforms algorithm more stable. If using trans = "none" produces an error for a particular posterior simulation then trans = "BC" is used instead.

n_v
: A numeric scalar. Each of the n_v largest values in u_vec will be used (separately) as a *validation* threshold for the training thresholds in u_vec that lie at or below that validation threshold. If n_v = 1 then all the training thresholds are used with validation performed using the threshold u_vec[length(u_vec)]. If n_v = 2 then, in addition, the assessment is performed using u_vec[1], ..., u_vec[length(u_vec) - 1] with validation threshold u_vec[length(u_vec) - 1], and so on.

npy   A numeric scalar. The mean number of observations per year of data, after excluding any missing values, i.e. the number of non-missing observations divided by total number of years of non-missing data. May be supplied using as an attribute attr(data, "npy") of data instead.

     The value of npy does not affect any calculation in ithresh, it only affects subsequent extreme value inferences using predict.ithresh. However, setting npy in the call to rpost, or as an attribute of data avoids the need to supply npy when calling predict.ithresh.

use_rcpp A logical scalar. If TRUE (the default) the revdbayes function rpost_rcpp is used for posterior simulation. If FALSE, or if the user supplies an R function to set the prior for GP parameters, the (slower) function rpost is used.

Details

For a given threshold in u_vec:

- the number of values in data that exceed the threshold, and the amounts (the *threshold excesses*) by which these value exceed the threshold are calculated;
- rpost_rcpp (or rpost) is used to sample from the posterior distributions of the parameters of a GP model for the threshold excesses and a binomial model for the probability of threshold exceedance;
- the ability of this binomial-GP model to predict data thresholded at the validation threshold(s) specified by n_v is assessed using leave-one-out cross-validation (the measure of this is given in equation (7) of Northrop et al. (2017).

See Northrop et al. (2017) and the introductory threshr vignette for further details and examples.

Value

An object (list) of class "ithresh", containing the components

- pred_perf: A numeric matrix with length(u_vec) rows and n_v columns. Each column contains the values of the measure of predictive performance. Entries corresponding to cases where the training threshold is above the validation threshold will be NA.
- u_vec: The argument u_vec to ithresh.
- v_vec: A numeric vector. The validation thresholds implied by the argument n_v to ithresh.
- u_ps: A numeric vector. The approximate levels of the sample quantiles to which the values in u_vec correspond, i.e. the approximate percentage of the data the lie at or below each element in u_vec.
- v_ps: A numeric vector. The values in u_ps that correspond to the validation thresholds.
- sim_vals: A numeric matrix with 4 columns and n x length(u_vec) rows. The jth block of n rows contains in columns 1-3 the posterior samples of the threshold exceedance probability, the GP scale parameter and the GP shape parameter respectively, based on training threshold u_vec[i], and in column 4 the value of j.
- n: A numeric scalar. The value of n.
- npy: A numeric scalar. The value of npy.
- data: The argument data to ithresh detailed above, with any missing values removed.

- use_rcpp: A logical scalar indicating whether rpost_rcpp (use_rcpp = TRUE) or rpost (use_rcpp = FALSE) was used for posterior simulation.
- for_post: A list containing arguments with which rpost_rcpp (or rpost) was called, including any user-supplied arguments to these functions.
- call: The call to ithresh.

References

Northrop, P.J. and Attalides, N. (2016) Posterior propriety in Bayesian extreme value analyses using reference priors *Statistica Sinica*, **26**(2), 721–743 doi:10.5705/ss.2014.034.

Northrop, P. J., Attalides, N. and Jonathan, P. (2017) Cross-validatory extreme value threshold selection and uncertainty with application to ocean storm severity. *Journal of the Royal Statistical Society Series C: Applied Statistics*, **66**(1), 93-120. doi:10.1111/rssc.12159

Jonathan, P. and Ewans, K. (2013) Statistical modelling of extreme ocean environments for marine design : a review. *Ocean Engineering*, **62**, 91-109. doi:10.1016/j.oceaneng.2013.01.004

See Also

plot.ithresh for the S3 plot method for objects of class ithresh.

summary.ithresh Summarizing measures of threshold predictive performance.

predict.ithresh for predictive inference for the largest value observed in N years.

rpost in the revdbayes package for details of the arguments that can be passed to rpost_rcpp/rpost.

set_prior and set_bin_prior in the revdbayes package for details of how to set a prior distributions for GP parameters and for the exceedance probability p.

quantile.

Examples

```
# Note:
# 1. Smoother plots result from making n larger than the default n = 1000.
# 2. In some examples below validation thresholds rather higher than is
#    advisable have been used, with far fewer excesses than the minimum of
#    50 suggested by Jonathan and Ewans (2013).

## North Sea significant wave heights, default prior ----------------------
#' # A plot akin to the top left of Figure 7 in Northrop et al. (2017)
#' # ... but with fewer training thresholds

u_vec_ns <- quantile(ns, probs = seq(0.1, 0.9, by = 0.1))
ns_cv <- ithresh(data = ns, u_vec = u_vec_ns, n_v = 2)
plot(ns_cv, lwd = 2, add_legend = TRUE, legend_pos = "topright")
mtext("significant wave height / m", side = 3, line = 2.5)

## Gulf of Mexico significant wave heights, default prior ------------------

u_vec_gom <- quantile(gom, probs = seq(0.2, 0.9, by = 0.1))
# Setting a prior using its name and parameter value(s) -------------------
# This example gives the same prior as the default
```

```
gom_cv <- ithresh(data = gom, u_vec = u_vec_gom, n_v = 2, prior = "mdi",
                  h_prior = list(a = 0.6))

## Setting a user-defined (log-)prior R function --------------------------
# This example also gives the same prior as the default
# (It will take longer to run than the example above because ithresh detects
#  that the prior is an R function and sets use_rcpp to FALSE.)

user_prior <- function(pars, a, min_xi = -1) {
  if (pars[1] <= 0 | pars[2] < min_xi) {
    return(-Inf)
  }
  return(-log(pars[1]) - a * pars[2])
}
user_bin_prior <- function(p, ab) {
  return(stats::dbeta(p, shape1 = ab[1], shape2 = ab[2], log = TRUE))
}
gom_cv <- ithresh(data = gom, u_vec = u_vec_gom, n_v = 2, prior = user_prior,
                  h_prior = list(a = 0.6), bin_prior = user_bin_prior,
                  h_bin_prior = list(ab = c(1 / 2, 1 / 2)))

## Setting a user-defined (log-)prior (pointer to a) C++ function ----------
# We make use of a C++ function and function create_prior_xptr() to create
# the required pointer from the revdbayes package

prior_ptr <- revdbayes::create_prior_xptr("gp_flat")
gom_cv <- ithresh(data = gom, u_vec = u_vec_gom, n_v = 2, prior = prior_ptr,
                  h_prior = list(min_xi = -1))
```

---

ns   *Storm peak significant wave heights from the North Sea*

Description

A numeric vector containing 628 hindcasts of storm peak significant wave heights, in metres, from 1964 to 1995 at an unnamed location in the North Sea.

Usage

```
ns
```

Format

A vector containing 628 observations.

Source

Oceanweather Inc. (1995) NEXT – North Sea hindcast study.

References

Northrop, P. J., N. Attalides, and P. Jonathan. (2017). Cross-Validatory Extreme Value Threshold Selection and Uncertainty with Application to Ocean Storm Severity. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **66**(1), 93-120. doi:10.1111/rssc.12159.

---

plot.ithresh	*Plot diagnostics an ithresh object*

Description

plot method for class "ithresh". Produces an extreme value threshold diagnostic plot based on an analysis performed by ithresh. Can also be used to produce a plot of the posterior sample generated by ithresh for a particular training threshold.

Usage

```
## S3 method for class 'ithresh'
plot(
  x,
  y,
  ...,
  which_v = NULL,
  prob = TRUE,
  top_scale = TRUE,
  add_legend = FALSE,
  legend_pos = "topleft",
  which_u = NULL
)
```

Arguments

x	an object of class "ithresh", a result of a call to ithresh.
y	Not used.
...	Additional arguments passed on to matplot and/or legend and/or axis. If which_u is supplied then these arguments are passed to plot.evpost.
which_v	A numeric scalar or vector.
	If which_u is not supplied (a threshold diagnostic plot is required) which_v specifies the validation thresholds, that is, the components of x$v_vec, to include in the plot.
	If which_u is supplied (a plot of a posterior sample for a given threshold is required) then which_v is a numeric scalar that indicates which element of object$v_vec is used in selecting a single threshold (if which_u = "best"). Note: the default, which_v = 1 gives the *lowest* of the validation thresholds in object$v_vec.

prob	A logical scalar. If `TRUE` then the levels of thresholds are represented by the proportion of observations that lie below a threshold. If `prob = FALSE` then the values of the thresholds are used.
top_scale	A logical scalar indicating Whether or not to add a scale to the top horizontal axis. If this is added it gives the threshold on the scale not chosen by `prob`.
add_legend	A logical scalar indicating whether or not to add a legend to the plot. If `method = "cv"` then the legend gives the levels of the validation thresholds.
legend_pos	The position of the legend (if required) specified using the argument x in [legend](#).
which_u	Either a character scalar or a numeric scalar. If `which_u` is supplied then [plot.evpost](#) is used to produce a plot of the posterior sample generated using a particular training threshold. By default a scatter plot of the posterior sample of Generalized Pareto parameters is produced.
	If `which_u = "best"` then the training threshold achieving the largest measure of predictive performance in object$pred_perf, based on the validation threshold selected using which_v, is used. See [summary.ithresh](#) to print the best thresholds for each validation threshold.
	Otherwise, `which_u` is a numeric scalar that selects training threshold x$u_vec[which_u]. Therefore, `which_u` must be an integer in 1, ..., length(x$u_vec).

Details

Produces plots of the *threshold weights*, defined in equation (14) of Northrop et al. (2017) against training threshold. A line is produced for each of the validation thresholds chosen in which_v. The result is a plot like those in the top row of Figure 7 in Northrop et al. (2017).

It is possible that a curve on the plot may be incomplete. This indicates that, for a particular threshold level, a measure of predictive performance is `-Inf`. This occurs when an observation in the data lies above the estimated upper end point of the predictive distribution produced when this observation is removed.

Value

If `which_u` is supplied then the object with which [plot.evpost](#) was called is returned (invisibly). Otherwise, a list is returned (again invisibly) with two components. x is a vector containing the co-ordinates plotted on the (lower) horizontal axis. y is an length(u_vec) by n_v matrix of *threshold weights* obtained by normalising the columns of the matrix pred_perf returned by [ithresh](#). See equation (14) of Northrop et al. (2017).

See Also

[ithresh](#) for threshold selection in the i.i.d. case based on leave-one-out cross-validation.

[summary.ithresh](#) Summarizing measures of threshold predictive performance.

[print.ithresh](#) Prints the threshold weights.

[predict.ithresh](#) for predictive inference for the largest value observed in N years.

Examples

```
# [Smoother plots result from making n larger than the default n = 1000.]

# Threshold diagnostic plot
u_vec_gom <- quantile(gom, probs = seq(0, 0.9, by = 0.05))
gom_cv <- ithresh(data = gom, u_vec = u_vec_gom, n_v = 3)
plot(gom_cv, lwd = 2, add_legend = TRUE, legend_pos = "topleft")
mtext("significant wave height / m", side = 3, line = 2.5)

# Plot of Generalized Pareto posterior sample at the best threshold
# (based on the lowest validation threshold)
plot(gom_cv, which_u = "best")
# See which threshold was used
summary(gom_cv)

# Plot of Generalized Pareto posterior sample at the highest threshold
n_u <- length(u_vec_gom)
plot(gom_cv, which_u = n_u, points_par = list(pch = 20, col = "grey"))
```

---

plot.ithreshpred   *Plot diagnostics an ithreshpred object*

Description

plot method for class "ithreshpred". Produces plots to summarise the predictive inferences made by predict.ithresh.

Usage

```
## S3 method for class 'ithreshpred'
plot(x, ..., ave_only = FALSE, add_best = FALSE)
```

Arguments

x	an object of class "ithreshpred", a result of a call to ithresh.
...	Additional arguments passed on to plot.evpred.
ave_only	A logical scalar. Only relevant if predict.ithresh was called with which_u = "all". If TRUE then plot only a curve for the weighted average over multiple training thresholds. If FALSE then also plot a curve for each training threshold.
add_best	A logical scalar. If TRUE then the best threshold, as judged using the validation threshold selected using the argument which_v supplied to predict.ithresh, is highlighted by plotting it with a different line style.

Details

Single threshold case, where `predict.ithresh` was called with numeric scalar `which_u` or `which_u` = `"best"`. `plot.evpred` is called to produce the plot.

Multiple threshold case, where `predict.ithresh` was called with `which_u` = `"all"`. Again, `plot.evpred` is called but now the estimated predictive distribution function (type = `"p"` used in the call to `predict.ithresh`) or density function (type = `"d"`) is plotted for each of the training thresholds (grey lines) as is the result of the weighted average over the different training thresholds (black line). If graphical parameters, such as `lty`, `lwd` or `col` are passed via ... then the first element relates to the weighted average and the remaining `length(x$u_vec)` elements to the respective training thresholds in `u_vec`.

Value

A list containing the graphical parameters using in producing the plot including any arguments supplied via ... is returned (invisibly).

See Also

`ithresh` for threshold selection in the i.i.d. case based on leave-one-out cross-validation.

`predict.ithresh` for predictive inference for the largest value observed in N years.

`plot.ithresh` for the S3 plot method for objects of class `ithresh`.

`summary.ithresh` Summarizing measures of threshold predictive performance.

Examples

```
u_vec_gom <- quantile(gom, probs = seq(0, 0.9, by = 0.05))
gom_cv <- ithresh(data = gom, u_vec = u_vec_gom, n_v = 3)

# Note: gom_cv$npy contains the correct value of npy (it was set in the
#       call to ithresh, via attr(gom, "npy").
#       If object$npy doesn't exist then the argument npy must be supplied
#       in the call to predict().

### Best training threshold based on the lowest validation threshold

# Predictive distribution function
npy_gom <- length(gom)/105
best_p <- predict(gom_cv, n_years = c(100, 1000))
plot(best_p)

# Predictive density function
best_d <- predict(gom_cv, type = "d", n_years = c(100, 1000))
plot(best_d)

### All thresholds plus weighted average of inferences over all thresholds

# Predictive distribution function
all_p <- predict(gom_cv, which_u = "all")
plot(all_p)
```

```
# Predictive density function
all_d <- predict(gom_cv, which_u = "all", type = "d")
plot(all_d)

### ... and highlight the best threshold

plot(all_p, add_best = TRUE)
plot(all_d, add_best = TRUE)
```

---

plot.stability   *Plot diagnostics for a stability object*

Description

plot method for objects of class "stability" returned from [stability](#)

Usage

```
## S3 method for class 'stability'
plot(
  x,
  y,
  ...,
  prob = TRUE,
  top_scale = c("none", "excesses", "opposite"),
  vertical = TRUE
)
```

Arguments

x	an object of class "stability", a result of a call to [stability](#).
y	Not used.
...	Additional arguments passed on to [matplot](#), [axis](#) and/or [segments](#).
prob	A logical scalar. If TRUE then the levels of thresholds on the lower horizontal axis are represented by the proportion of observations that lie below a threshold. If prob = FALSE then the values of the thresholds are used.
top_scale	A character scalar. If top_scale = "none" then no axis labels appear on the upper horizontal axis. If top_scale = "excesses" then the number of threshold excesses at each threshold are indicated. If top_scale = "opposite" then the type of threshold level *not* chosen using prob is indicated.
vertical	A logical scalar. Should the confidence intervals be depicted using a vertical line for each threshold (TRUE) or by joining up confidence limits across thresholds (FALSE)?

Details

Produces a simple threshold diagnostic plot based on the object returned from `stability`. The MLEs of the GP shape parameter ξ and approximate `conf`% confidence intervals for ξ are plotted against the threshold used to fit the GP model. This plot is used to choose a threshold above which the underlying GP shape parameter may be approximately constant. See Chapter 4 of Coles (2001). See also the vignette "Introducing threshr". as described in . See also the vignette "Introducing threshr".

Value

In addition to producing the plot a list of the arguments used by `matplot`, `axis` is returned (invisibly).

See Also

`stability`.

Examples

```
u_vec_gom <- quantile(gom, probs = seq(0, 0.9, by = 0.05))
gom_stab <- stability(data = gom, u_vec = u_vec_gom)
plot(gom_stab)
```

---

predict.ithresh   *Predictive inference for the largest value observed in N years.*

Description

`predict` method for class `"ithresh"`. Predictive inferences can either be based on a *single training threshold* or using a weighted average of inferences over *multiple training thresholds*. A single threshold may chosen to be the best performing threshold, as judged by the measure of predictive performance calculated by `ithresh` or chosen by the user. The weights used in the latter case are based on the measures of predictive performance and prior probabilities assigned to the training thresholds. By default all thresholds are given the same prior probability but the user can specify their own prior.

Usage

```
## S3 method for class 'ithresh'
predict(
  object,
  npy = NULL,
  n_years = 100,
  which_u = c("best", "all"),
  which_v = 1L,
  u_prior = rep(1, length(object$u_vec)),
  type = c("p", "d", "q", "i", "r"),
```

```
    hpd = FALSE,
    x = NULL,
    ...
)
```

Arguments

object	An object of class "ithresh", a result of a call to [ithresh](#).
npy	A numeric scalar. The mean number of observations per year of data, after excluding any missing values, i.e. the number of non-missing observations divided by total number of years of non-missing data.
n_years	A numeric vector. Value(s) of N. If which_u = "all" then n_years must have length one.
which_u	Either a character scalar or a numeric scalar. If which_u is a character scalar it must be either "best" or "all".
	If which_u = "best" then the single threshold achieving the largest measure of predictive performance in object$pred_perf, based on the validation threshold selected using which_v, is used to perform prediction. See [summary.ithresh](#) to print the best thresholds for each validation threshold.
	If which_u = "all" then *all* the thresholds are used to perform prediction. The inferences from each threshold are weighted according to the *posterior threshold weights* given in equation (15) of Northrop et al. (2017) based on the prior probabilities of thresholds in u_prior and column which_v of the measures of predictive performance in object$pred_perf.
	Otherwise, which_u is a numeric scalar that indicates which element of object$u_vec the user wishes to select as a single threshold on which to base prediction, that is, which_u must be an integer in 1, ..., length(object$u_vec).
which_v	A numeric scalar. Indicates which element of object$v_vec is used in selecting a single threshold (if which_u = "best") or weighting the inferences from all thresholds (if which_u = "all"). Note: the default, which_v = 1 gives the *lowest* of the validation thresholds in object$v_vec.
u_prior	A numeric vector. Prior probabilities for the training thresholds in object$u_vec. Only used if which_u = "all".
	Only the first length(object$u_vec) - length(object$v_vec) + which_v elements of u_prior are used. This is because only training thresholds up to and including object$v_vec[which_v] are relevant. u_prior must have length length(object$u_vec) or length(object$u_vec) - length(object$v_vec) + which_v.
	If u_prior is not supplied then all (relevant) training thresholds are given equal prior probability. u_prior is normalized to have sum equal to 1 inside predict.ithresh.
type	A character vector. Passed to [predict.evpost](#). Indicates which type of inference is required:

- "p" for the predictive distribution function,
- "d" for the predictive density function,
- "q" for the predictive quantile function,

- • "i" for predictive intervals (see ... to set level),
- • "r" for random generation from the predictive distribution.

        If which_u = "all" then only type = "p" or type = "d" are allowed.

hpd   A logical scalar. The argument hpd of `predict.evpost`. Only relevant if type = "i".

x   A numeric vector. The argument x of `predict.evpost`. In the current context this must be a vector (not a matrix, as suggested by the documentation of `predict.evpost`). If x is not supplied then a default value is set within `predict.evpost`.

...   Additional arguments to be passed to `predict.evpost`. In particular: level, which can be used to set (multiple) levels for predictive intervals when type = "i"; lower_tail (relevant when type = "p" or "q") and log (relevant when type = "d").

Details

The function `predict.evpost` is used to perform predictive based on the binomial-GP posterior sample generated using a given training threshold. For mathematical details of the single threshold and multiple threshold cases see Sections 2.3 and 3 of Northrop et al. (2017) respectively.

Value

An list object of class "ithreshpred" with a similar structure to an object of class "evpred" returned from `predict.evpost` is returned *invisibly*. In addition, the object contains u_vec = object$u_vec and v_vec = object$v_vec, which_v and the index best_u in u_vec = object$u_vec of the best training threshold based on the value of which_v. It also contains the value of the Box-Cox transformation parameter lambda. This will always be equal to 1 if object was returned from ithresh.

If which_u = "all" then

- • the list also contains the *posterior threshold weights* in component post_thresh_wts

- • the component y is a matrix with length{x} rows and 1 + length(object$u_vec) − length(object$v_vec) + which_v columns. Column 1 contains the estimated predictive distribution function (type = "p") or density function (type = "d") obtained using a weighted average of the inferences over different training thresholds. The other columns contain the estimated functions for each of the training thresholds in u_vec.

References

Northrop, P. J., Attalides, N. and Jonathan, P. (2017) Cross-validatory extreme value threshold selection and uncertainty with application to ocean storm severity. *Journal of the Royal Statistical Society Series C: Applied Statistics*, **66**(1), 93-120. doi:10.1111/rssc.12159

See Also

`ithresh` for threshold selection in the i.i.d. case based on leave-one-out cross-validation.

`plot.ithreshpred` for the S3 plot method for objects of class ithreshpred.

Examples

```
# Note:
#'  In the examples below validation thresholds rather higher than is
#   advisable have been used, with far fewer excesses than the minimum of
#   50 suggested by Jonathan and Ewans (2013).

# Gulf of Mexico significant wave heights, default priors.
u_vec_gom <- quantile(gom, probs = seq(0, 0.9, by = 0.05))
gom_cv <- ithresh(data = gom, u_vec = u_vec_gom, n_v = 3)

# Note: gom_cv$npy contains the correct value of npy (it was set in the
#       call to ithresh, via attr(gom, "npy").
#       If object$npy doesn't exist then the argument npy must be supplied
#       in the call to predict().

### Best training threshold based on the lowest validation threshold

# Predictive distribution function
best_p <- predict(gom_cv, n_years = c(100, 1000))
plot(best_p)

# Predictive density function
best_d <- predict(gom_cv, type = "d", n_years = c(100, 1000))
plot(best_d)

# Predictive intervals
best_i <- predict(gom_cv, n_years = c(100, 1000), type = "i", hpd = TRUE,
                  level = c(95, 99))
plot(best_i, which_int = "both")

# See which threshold was used
summary(gom_cv)

### All thresholds plus weighted average of inferences over all thresholds

# Predictive distribution function
all_p <- predict(gom_cv, which_u = "all")
plot(all_p)

# Predictive density function
all_d <- predict(gom_cv, which_u = "all", type = "d")
plot(all_d)
```

---

  print.ithresh   *Print method for objects of class* "ithresh"

Description

print method for class "ithresh".

Usage

```
## S3 method for class 'ithresh'
print(x, digits = 2, ...)
```

Arguments

x	an object inheriting from class "ithresh", a result of a call to ithresh.
digits	An integer. Used for number formatting with format and signif.
...	Additional optional arguments. At present no optional arguments are used.

Details

Prints a matrix of the estimated threshold weights. Each row gives the weights for each training threshold for a given validation threshold. The row and column names are the approximate quantile levels of the thresholds.

Value

The argument x, invisibly, as for all print methods.

See Also

ithresh for threshold selection in the i.i.d. case based on leave-one-out cross-validation.

summary.ithresh Summarizing measures of threshold predictive performance.

plot.ithresh for the S3 plot method for objects of class ithresh.

predict.ithresh for predictive inference for the largest value observed in N years.

---

stability	*Generalized Pareto parameter estimate stability*

Description

Uses maximum likelihood estimation to fit a Generalized Pareto (GP) model to threshold excesses over a range of thresholds. The threshold excesses are treated as independent and identically distributed (i.i.d.) observations. The resulting estimates and confidence intervals can be plotted, using plot.stability, to produce a crude graphical diagnostic for threshold choice.

Usage

```
stability(
  data,
  u_vec,
  prof = FALSE,
  conf = 95,
  mult = 1:2,
  plot_prof = FALSE,
  ...
)
```

Arguments

data	A numeric vector of observations.
u_vec	A numeric vector of thresholds to be applied to the data. Any duplicated values will be removed. These could be set at sample quantiles of data using [quantile]().
prof	A logical scalar. Whether to calculate confidence intervals for the GP shape parameter ξ based on the profile-likelihood for ξ or using the MLE plus or minus a multiple of the estimated standard error (SE) of the MLE. The intervals produced by the former may be better but they take longer to calculate. Default: FALSE.
conf	A numeric scalar in (0, 100). Confidence level for the confidence intervals. Default: 95%.
mult	A numeric vector of length 2. The range of values over which the profile log-likelihood for ξ is calculated is (MLE - mult[1] c SE, MLE + mult[2] c SE), where MLE and SE are the MLE and estimated standard error of ξ and c is the constant for which this interval gives an approximate 100conf% level confidence interval for ξ when mult = c(1, 1). The default, mult = c(1, 2), works well in most cases. If the routine fails because the range of ξ is not sufficiently wide then the relevant components of mult should be increased.
plot_prof	A logical scalar. Only relevant if prof = TRUE. If plot_prof = TRUE then the profile log-likelihood is plotted for each threshold. If FALSE then nothing is plotted.
...	Further (optional) arguments to be passed to the [optim]() function for the optimizations on which the profile-likelihood for xi is based.

Details

For each threshold in u_vec a GP model is fitted by maximum likelihood estimation to the threshold excesses, i.e. the amounts by which the data exceed that threshold. The MLEs of the GP shape parameter ξ and approximate conf% confidence intervals for ξ are stored for plotting (by [plot.stability]()) to produce a simple graphical diagnostic to inform threshold selection. This plot is used to choose a threshold above which the underlying GP shape parameter may be approximately constant. See Chapter 4 of Coles (2001). See also the vignette "Introducing threshr".

Value

An object (list) of class "stability" with components:

ests	MLEs of the GP shape parameter ξ.
ses	Estimated SEs of the MLEs of ξ.
lower	Lower limit of 100conf% confidence intervals for ξ.
upper	Upper limit of 100conf% confidence intervals for ξ.
nexc	The number of threshold excesses.
u_vec	The thresholds supplied by the user.
u_ps	The approximate sample quantiles to which the thresholds in u_vec correspond.

data   The input data.

conf   The input conf.

Each of these components is a numeric vector of length `length(u_vec)`.

References

Coles, S. G. (2001) *An Introduction to Statistical Modeling of Extreme Values*, Springer-Verlag, London. doi:10.1007/9781447136750_3

See Also

`ithresh` for threshold selection in the i.i.d. case based on leave-one-out cross-validation.

`plot.stability` for the S3 plot method for objects of class `stability`.

`quantile`.

Examples

```
# Set a vector of thresholds
u_vec_gom <- quantile(gom, probs = seq(0, 0.9, by = 0.05))

# Symmetric confidence intervals
gom_stab <- stability(data = gom, u_vec = u_vec_gom)
plot(gom_stab)

# Profile-likelihood-based confidence intervals
gom_stab <- stability(data = gom, u_vec = u_vec_gom, prof = TRUE)
plot(gom_stab)
```

---

summary.ithresh   *Summarizing measures of threshold predictive performance*

Description

summary method for class `"ithresh"`

Usage

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

Arguments

object   an object of class `"ithresh"`, a result of a call to `ithresh`.

...   Additional optional arguments. At present no optional arguments are used.

Value

Returns a numeric matrix with 5 columns and n_v rows, where n_v is an argument to ithresh that determines how many of the largest training thresholds are used a validation thresholds. The columns contain:

- column 1: the validation threshold v
- column 2: the sample quantile to which the validation threshold corresponds
- column 3: the best training threshold u judged using the validation threshold v
- column 4: the sample quantile to which the best training threshold corresponds
- column 5: the index of the vector u_vec of training thresholds to which the threshold in column2 corresponds

See Also

ithresh for threshold selection in the i.i.d. case based on leave-one-out cross-validation.

plot.ithresh for the S3 plot method for objects of class ithresh.

print.ithresh Prints the threshold weights.

Examples

```
u_vec_gom <- quantile(gom, probs = seq(0, 0.9, by = 0.05))
gom_cv <- ithresh(data = gom, u_vec = u_vec_gom, n_v = 3)
summary(gom_cv)
```

Index