# Package: circtree (via r-universe)

September 16, 2024

**Title** Regression Trees and Forests for Circular Responses

**Version** 0.1-0

**Description** Infrastructure for fitting distributional trees and forests based on maximum-likelihood estimation of parameters for a circular response, as well as regression methods for a circular response based on maximum-likelihood estimation are provided. For both approaches the von Mises distribution is employed as circular response distribution.

**Depends** R (>= 3.4.0), partykit (>= 1.2-5), disttree (>= 0.2-0)

**License** GPL-2 | GPL-3

**Imports** stats, circular, Formula, movMF, gridGraphics, grDevices, sandwich, scales, latex2exp

**Suggests** testthat, methods

**RoxygenNote** 6.1.1

**Repository** https://zeileis.r-universe.dev

**RemoteUrl** https://github.com/r-forge/partykit

**RemoteRef** HEAD

**RemoteSha** 4309c9bb2890ddf1e743e937b93a0281f28e2e17

## Contents

---

`circfit` *Maximum-Likelihood Fitting for a Circular Response*

---

## Description

The function `circfit` carries out maximum-likelihood estimation of parameters for a circular response employing the von Mises distribution. The parameters can be transformed through link functions but do not depend on further covariates (i.e., are constant across observations).

## Usage

```
circfit(y, weights = NULL, start = NULL, start.eta = NULL,
        response_range = NULL,
        vcov = TRUE, type.hessian =  c("checklist", "analytic", "numeric"),
        method = "L-BFGS-B", estfun = TRUE, optim.control = list(), ...)
```

## Arguments

| | |
|---|---|
| `y` | numeric vector of the response |
| `weights` | optional numeric vector of case weights. |
| `start` | starting values for the distribution parameters handed over to `optim` |
| `start.eta` | starting values for the distribution parameters on the link scale handed over to `optim`. |
| `response_range` | either a logical value indicating whether the response should be transformed to its original range (TRUE) or kept on the interval (-pi,pi] or a two-dimensional vector specifying a range of the circular response. |
| `vcov` | logical. Specifies whether or not a variance-covariance matrix should be calculated and returned. |
| `type.hessian` | Can either be 'checklist', 'analytic' or 'numeric' to decide how the hessian matrix should be calculated in the fitting process in `distfit`. For 'checklist' it is checked whether a function 'hdist' is given in the family list. If so, 'type.hessian' is set to 'analytic', otherwise to 'numeric'. |
| `method` | Optimization which should be applied in `optim` |
| `estfun` | logical. Should the matrix of observation-wise score contributions (or empirical estimating functions) be returned? |
| `optim.control` | A list with `optim` control parameters. |
| `...` | further arguments passed to `optim`. |

### Details

The function `circfit` fits the parameter of the von Mises distribution to a circular response variable by applying `distfit`.

### Value

An object of S3 class `circfit` inheriting from class `distfit`.

### See Also

`distfit`

### Examples

```
## example on parameter range:
sdat.par <- circtree_simulate(response_range = c(-pi, pi))
cf.par <- circfit(sdat.par$y)


## example on response range (0, 2pi):
sdat.rad <- circtree_simulate(response_range = c(0, 2*pi))
cf.rad <- circfit(sdat.rad$y)

## example on response range (0, 360):
sdat.deg <- circtree_simulate(response_range = c(0, 360))
cf.deg <- circfit(sdat.deg$y)
```

---

## circforest *Distributional Regression Forests for a Circular Response*

---

### Description

Distributional forests based on maximum-likelihood estimation of parameters for a circular response employing the von Mises distribution.

### Usage

```
circforest(formula, data, response_range = NULL, subset,
           na.action = na.pass, weights, offset, cluster, strata,
           control = disttree_control(teststat = "quad", testtype = "Univ",
           mincriterion = 0, saveinfo = FALSE, minsplit = 20, minbucket = 7,
           splittry = 2, ...), ntree = 500L, fit.par = FALSE,
           perturb = list(replace = FALSE, fraction = 0.632),
       mtry = ceiling(sqrt(nvar)), applyfun = NULL, cores = NULL, trace = FALSE, ...)
## S3 method for class 'circforest'
predict(object, newdata = NULL,
        type = c("parameter", "response", "weights", "node"),
        OOB = TRUE, scale = TRUE, response_range = FALSE, ...)
```

## Arguments

| | |
|---|---|
| formula | a symbolic description of the model to be fit. This should be of type y ~ x1 + x2 where y should be the response variable and x1 and x2 are used as partitioning variables. |
| data | an optional data frame containing the variables in the model. |
| response_range | either a logical value indicating whether the response should be transformed to its original range (TRUE) or kept on the interval (-pi,pi] or a two-dimensional vector specifying a range of the circular response. |
| subset | an optional vector specifying a subset of observations to be used in the fitting process. |
| na.action | a function which indicates what should happen when the data contain missing value. |
| weights | optional numeric vector of case weights. |
| offset | an optional vector of offset values. |
| cluster | an optional factor indicating independent clusters. Highly experimental, use at your own risk. |
| strata | an optional factor for stratified sampling. |
| control | a list with control parameters passed to extree_fit via disttree_control The default values that are not set within the call of distforest correspond to those of the default values used by disttree from the disttree package. saveinfo = FALSE leads to less memory hungry representations of trees. Note that arguments mtry, cores and applyfun in disttree_control are ignored for distforest, because they are already set. |
| ntree | number of trees to grow for the forest. |
| fit.par | logical. if TRUE, fitted and predicted values and predicted parameters are calculated for the learning data (together with loglikelihood) |
| perturb | a list with arguments replace and fraction determining which type of resampling with replace = TRUE referring to the n-out-of-n bootstrap and replace = FALSE to sample splitting. fraction is the number of observations to draw without replacement. |
| mtry | number of input variables randomly sampled as candidates at each node for random forest like algorithms. Bagging, as special case of a random forest without random input variable sampling, can be performed by setting mtry either equal to Inf or manually equal to the number of input variables. |
| applyfun | an optional lapply-style function with arguments function(X, FUN, ...). It is used for computing the variable selection criterion. The default is to use the basic lapply function unless the cores argument is specified (see below). |
| cores | numeric. If set to an integer the applyfun is set to mclapply with the desired number of cores. |
| trace | a logical indicating if a progress bar shall be printed while the forest grows. |
| object | an object as returned by circforest |
| newdata | an optional data frame containing test data. |

| | |
|---|---|
| type | a character string denoting the type of predicted value returned. For "parameter" the predicted distributional parameters are returned on the range of (-pi, pi] and for "response" the expectation on the range of the response is returned (response_range). "weights" returns an integer vector of prediction weights. For type = "node", a list of terminal node ids for each of the trees in the forest ist returned. |
| OOB | a logical defining out-of-bag predictions (only if newdata = NULL). |
| scale | a logical indicating scaling of the nearest neighbor weights by the sum of weights in the corresponding terminal node of each tree. In the simple regression forest, predicting the conditional mean by nearest neighbor weights will be equivalent to (but slower!) the aggregation of means. |
| ... | arguments to be used to form the default control argument if it is not supplied directly. |

## Details

Distributional regression forests for a circular response are an application of model-based recursive partitioning and unbiased recursive partitioning based on the implementation in distforest using the infrastructure of extree_fit.

## Value

An object of S3 class circforest inheriting from class distforest.

## See Also

distforest, disttree, distfit, extree_fit

## Examples

```
#sdat <- circtree_simulate()
#cf <- circforest(y ~ x1 + x2, data = sdat, ntree = 50)
```

---

# circmax *Circular Regression with Maximum Likelihood Estimation*

---

## Description

Fit a regression model for a circular response by maximum likelihood estimation employing the von Mises distribution.

## Usage

```
circmax(formula, data, subset, na.action,
  model = TRUE, y = TRUE, x = FALSE,
  control = circmax_control(...), ...)

circmax_fit(x, y, z = NULL, control)

circmax_control(maxit = 5000, start = NULL, method = "Nelder-Mead",
  solve_kappa = "Newton-Fourier",
  gradient = FALSE, hessian = TRUE, ...)
```

## Arguments

| | |
|---|---|
| formula | a formula expression of the form y ~ x \| z where y is the response and x and z are regressor variables for the location and the concentration of the von Mises distribution. |
| data | an optional data frame containing the variables occurring in the formulas; y has to be given in radians. |
| subset | an optional vector specifying a subset of observations to be used for fitting. |
| na.action | a function which indicates what should happen when the data contain NAs. |
| model | logical. If TRUE *model frame* is included as a component of the returned value. |
| x, y | for circmax: logical. If TRUE the model matrix and response vector used for fitting are returned as components of the returned value. For circmax_fit: x is a design matrix with regressors for the location and y is a vector of observations given in radians. |
| z | a design matrix with regressors for the concentration. |
| ... | arguments to be used to form the default control argument if it is not supplied directly. |
| control, maxit, start | a list of control parameters passed to optim. |
| method | The method to be used for optimization. |
| solve_kappa | Which kappa solver should be used for the starting values for kappa. By default a Newton Fourier is used ("Newton-Fourier"). Alternatively, a uniroot provides a safe option ("Uniroot") or code"Banerjee_et_al_2005" provides a quick approximation). |
| gradient | logical. Should gradients be used for optimization? If TRUE, the default method is "BFGS". Otherwise method = "Nelder-Mead" is used. |
| hessian | logical or character. Should a numeric approximation of the (negative) Hessian matrix by optim be computed? |

## Details

circmax fits a regression model for a circular response assuming a von Mises distribution.

circmax_fit is the lower level function where the parameters of the von Mises distribution are fitted by maximum likelihood estimation.

## Value

An object of class "circmax".

## Examples

```
## Example 1: Simulated Data:

sdat <- circmax_simulate(n = 1000, beta = c(3, 5, 2), gamma = c(3, 3))

(m1.circmax <- circmax(y ~ x1 + x2 | x3, data = sdat))

## Example 2: Periwinkle Dataset of Fisher and Lee, 1992:
require("circular")
distance <- c(107, 46, 33, 67, 122, 69, 43, 30, 12, 25, 37, 69, 5, 83,
  68, 38, 21, 1, 71, 60, 71, 71, 57, 53, 38, 70, 7, 48, 7, 21, 27)
directdeg <- c(67, 66, 74, 61, 58, 60, 100, 89, 171, 166, 98, 60, 197,
  98, 86, 123, 165, 133, 101, 105, 71, 84, 75, 98, 83, 71, 74, 91, 38, 200, 56)
cdirect <- circular(directdeg * 2 * pi/360)
plot(as.numeric(cdirect) ~ distance, ylim = c(0, 4*pi), pch = 20)
points(as.numeric(cdirect) + 2*pi ~ distance, pch = 20)

(m2.circ <- lm.circular(type = "c-l", y = cdirect, x = distance, init = 0.0))
(m2.circmax <- circmax(cdirect ~ distance, data = data.frame(cbind(distance, cdirect))))
```

---

# circmax_simulate *Simulated Data Set for* circmax

---

## Description

This function creates artifical data set for testing the regression models for a circular response by maximum likelihood estimation.

## Usage

```
circmax_simulate(n = 1000, beta = c(3, 5, 2), gamma = c(3, 3), seed = 111)
```

## Arguments

| | |
|---|---|
| n | The number of Observations. |
| beta | The coefficients for the intercept and the covariates of the location part. |
| gamma | The coefficients for the intercept and the covariates of the concentration part. |
| seed | Sets the 'seed' to a numeric value. |

## Value

Data frame with simualated covariates and respective response.

circtree *Distributional Regression Tree for a Circular Response*

## Description

Distributional trees based on maximum-likelihood estimation of parameters for a circular response employing the von Mises distribution.

## Usage

```
circtree(formula, data, response_range = NULL, subset, na.action = na.pass,
         weights, offset, cluster, control = disttree_control(...),
         converged = NULL, scores = NULL, doFit = TRUE, ...)
```

## Arguments

| | |
|---|---|
| formula | a symbolic description of the model to be fit. This should be of type y ~ x1 + x2 where y should be the response variable and x1 and x2 are used as partitioning variables. |
| data | an optional data frame containing the variables in the model. |
| response_range | either a logical value indicating whether the response should be transformed to its original range (TRUE) or kept on the interval (-pi,pi] or a two-dimensional vector specifying a range of the circular response. |
| subset | an optional vector specifying a subset of observations to be used in the fitting process. |
| na.action | a function which indicates what should happen when the data contain missing value. |
| weights | optional numeric vector of case weights. |
| offset | an optional vector of offset values. |
| cluster | an optional factor indicating independent clusters. Highly experimental, use at your own risk. |
| control | control arguments passed to extree_fit via disttree_control. |
| converged | an optional function for checking user-defined criteria before splits are implemented. |
| scores | an optional named list of scores to be attached to ordered factors. |
| doFit | a logical indicating if the tree shall be grown (TRUE) or not FALSE |
| ... | arguments to be used to form the default control argument if it is not supplied directly. |

## Details

Distributional regression trees for a circular response are an application of model-based recursive partitioning and unbiased recursive partitioning based on the implementation in disttree using the infrastructure of extree_fit.

## Value

An object of S3 class `circtree` inheriting from class `disttree`.

## See Also

`disttree`, `distfit`, `extree_fit`

## Examples

```
## example on parameter range:
sdat.par <- circtree_simulate(response_range = c(-pi, pi))
ct.par <- circtree(y ~ x1 + x2, data = sdat.par)
plot(ct.par)

## example on response range (0, 2pi):
sdat.rad <- circtree_simulate(response_range = c(0, 2*pi))
ct.rad <- circtree(y ~ x1 + x2, data = sdat.rad)
## default: type = "response"
plot(ct.rad, tp_args = list(response_range = FALSE))
plot(ct.rad, tp_args = list(response_range = TRUE))
plot(ct.rad, tp_args = list(response_range = c(0, 24)))

## example on response range (0, 360):
sdat.deg <- circtree_simulate(response_range = c(0, 360))
ct.deg <- circtree(y ~ x1 + x2, data = sdat.deg)
plot(ct.deg, tp_args = list(response_range = FALSE))
plot(ct.deg, tp_args = list(response_range = TRUE))
plot(ct.deg, tp_args = list(template = "geographics"))

## example on response range (0, 24):
sdat.hour <- circtree_simulate(response_range = c(0, 24))
ct.hour <- circtree(y ~ x1 + x2, data = sdat.hour, response_range = c(0, 24))
plot(ct.hour, tp_args = list(response_range = FALSE))
plot(ct.hour, tp_args = list(template = "clock24"))
plot(ct.hour, tp_args = list(template = "clock24",
  circlab = c("no", "mo", "mi", "ev")))
```

---

`circtree_simulate` *Simulated Data Set for* `circtree`

---

## Description

This function creates artifical data set for testing the regression trees employing a von Mises distribution.

## Usage

```
circtree_simulate(n = 1000, mu = c(0, 2, 5), kappa = c(3, 3, 1),
response_range = c(0, 2 * pi), seed = 111)
```

### Arguments

| | |
|---|---|
| n | The number of Observations. |
| mu | The distribution parameters for the location part. **Currently exactly three parameters necessary.** |
| kappa | The distribution parameters for the concentration part. **Currently exactly three parameters necessary.** |
| response_range | Defines range of simulated response. |
| seed | Sets the 'seed' to a numeric value. |

### Value

Data frame with simualated covariates and respective response.

---

## coef.circmax *Methods for CIRCMAX Objects*

---

### Description

Methods for extracting information from fitted circmax objects.

### Usage

```
## S3 method for class 'circmax'
coef(object, model = c("full", "location", "concentration"), ...)
## S3 method for class 'circmax'
terms(x, model = c("location", "concentration", "full"), ...)
```

### Arguments

| | |
|---|---|
| object, x | an object of class "circmax". |
| model | model for which coefficients shall be returned. |
| ... | further arguments passed to or from other methods. |

### Details

In addition to the methods above, a set of standard extractor functions for "circmax" objects is available, including methods to the generic functions print, logLik, and model.frame. Additionally, estfun, vcov provide methods for 'robust' inference.

### See Also

circmax

crps_vonmises *Circular CRPS*

## Description

Continuous Ranked Probability Score (CRPS) for a circular response following the von Mises distribution.

## Usage

```
crps_vonmises(y, mu, kappa, sum = FALSE, na.rm = FALSE)
```

## Arguments

| | |
|---|---|
| y | numeric. Circular response. |
| mu | numeric. Location parameter of the von Mises distribution. |
| kappa | numeric. Concentration parameter of the von Mises distribution. |
| sum | logical. Should the sum of the CRPS-values over all response values be returned. |
| na.rm | logical. Should missing values (including 'NaN') in case of 'sum = TRUE' be removed? |

## Value

For sum=TRUE the sum of the CRPS-values of all response values is returned. Otherwise a vector of the same length as y with the observation-wise CRPS-values is returned.

## See Also

circtree

## Examples

```
set.seed(123)
sdat <- circtree_simulate(n = 100)
m1.circtree <- circtree(y ~ x1 + x2, data = sdat)
foo <- function(x, deg = FALSE){
  if(deg) x <- x *  pi / 180
  tmp <- x
  tmp <- ifelse(tmp > pi, -(pi - (tmp - pi)), tmp)
  if(deg) tmp <- tmp *  180 / pi
  tmp
}

testcrps <- crps_vonmises(foo(sdat$y), mu = predict(m1.circtree)$mu, predict(m1.circtree)$kappa)
```

## dist_vonmises *Von Mises Family 'Dist-List' for* disttree.

### Description

Exported Von Mises Family for implementation in disttree.

### Usage

```
dist_vonmises(useC = FALSE, ncores = 1)
```

### Arguments

| | |
|---|---|
| useC | logical; if TRUE C routines are used. |
| ncores | Number of cores for parallelization with openMP (No big improvements in terms of running time). |

## dvonmises *Von Mises Density*

### Description

Density function for the von Mises distribution with location parameter mu and concentration parameter kappa.

### Usage

```
dvonmises(y, mu, kappa, log = FALSE)
```

### Arguments

| | |
|---|---|
| y | vector of observations. |
| mu | vector of location parameters. |
| kappa | vector of concentration parameters. |
| log | logical; if TRUE, probabilities p are given as log(p) |

### Value

Von Mises Density

`plot.circtree` *Plotting a Regression Tree with a Circular Response (under development).*

## Description

This function plots regression trees with a circular response based on `plot.constparty`.

## Usage

```
  ## S3 method for class 'circtree'
plot(x, terminal_panel = node_circular,
    tp_args = list(), tnex = NULL, drop_terminal = NULL, ...)
```

## Arguments

| | |
|---|---|
| `x` | Object of class `circtree`. |
| `terminal_panel` | Do not change. |
| `tp_args` | Do not change. |
| `tnex` | Do not change. |
| `drop_terminal` | Do not change. |
| `...` | Do not change. |

`predict.circmax` *Predicted/Fitted Values for CIRCMAX Fits*

## Description

Obtains various types of predictions for `circmax` models.

## Usage

```
## S3 method for class 'circmax'
predict(object, newdata = NULL, type = c("location", "concentration",
  "parameter"),
  na.action = na.pass, ...)
```

### Arguments

| | |
|---|---|
| object | an object of class "circmax". |
| newdata | an optional data frame in which to look for variables which to predict. |
| type | type of prediction: "location" returns the location of the predicted distribution. "scale" returns the scale of the predicted distribution. "parameter" returns a data frame with predicted location and scale parameters. |
| na.action | a function which indicates what should happen when the data contain NAs. Default is na.pass |
| ... | further arguments passed to or from other methods. |

### Value

For type "location", or "scale" a vector with either the location or the scale of the predicted distribution.

### See Also

circmax

---

## vonmises_bamlss *Von Mises Family for* bamlss.

---

### Description

Exported Von Mises Family for implementation in bamlss.

### Usage

```
vonmises_bamlss(...)
```

### Arguments

| | |
|---|---|
| ... | Not used. |

# Index