Package: conforest (via r-universe)

May 20, 2026

Type Package

Title Conformal Random Forests for Response Surface Emulation

Version 2.0.1

Description Fits emulators, also known as surrogates or response surfaces, using conformal inference with random forests. The conformal calibration is performed using out-of-bag samples from the forest, eliminating the need for a separate hold-out set. The method is based on Johansson et al. (2014 <doi:10.1007/s10994-014-5453-0>).

License BSD_3_clause + file LICENSE

Encoding UTF-8

Imports randomForest, RANN

RoxygenNote 7.3.2

NeedsCompilation no

Author Kellin Rumsey [aut, cre]

Maintainer Kellin Rumsey <knrumsey@lanl.gov>

Repository https://knrumsey-lanl.r-universe.dev

Date/Publication 2026-05-19 09:00:02 UTC

RemoteUrl https://github.com/cran/conforest

RemoteRef HEAD

RemoteSha 2b2e8c67666518ddc793bd5941374726d519d828

Contents

plot.rfok *Plot method for rfok*

Description

A conformal RF based on Johansson et al. (2014). Rather than using a traditional calibration set, conformal intervals are constructed using out-of-bag samples for each tree. K-nearest neighbors is used to calibrate the prediction intervals for each prediction location.

Usage

```
## S3 method for class 'rfok'
plot(x, conf = 0.95, ...)
```

Arguments

x	Object returned by rfok()
conf	Confidence level for plotting
...	Optional parameters to pass to plot()

Value

Invisibly returns x.

Examples

```
# Friedman function
f <-  function(x){
  10 * sin(pi * x[1] * x[2]) + 20 * (x[3] - 0.5)^2 + 10 * x[4] + 5 * x[5]
}
X <- matrix(runif(250), nrow=50, ncol=5)
y <- apply(X, 1, f)
fit <- rfok(X, y)

plot(fit)
```

predict.rfok *Predict method for rfok*

Description

A conformal RF based on Johansson et al. (2014). Rather than using a traditional calibration set, conformal intervals are constructed using out-of-bag samples for each tree. K-nearest neighbors is used to calibrate the prediction intervals for each prediction location.

Usage

```
## S3 method for class 'rfok'
predict(
  object,
  newdata = NULL,
  samples = 1000,
  conf = NULL,
  smoothing = TRUE,
  eps = 0,
  ...
)
```

Arguments

object	Object returned by rfok()
newdata	a data frame or matrix of new data
samples	Number of samples from the predictive distribution.
conf	a vector of levels for desired confidence intervals. Ignored unless samples is NULL.
smoothing	logical; should conformity scores be sampled or smoothed (using quantiles)
eps	Tolerance for nearest neighbor approach. When eps = 0, exact nearest neighbors are used.
...	Additional, ignored, parameters

Details

Prediction intervals are constructed by estimating a local scale for each prediction point using the k nearest training observations. The conformal margin of error is the requested quantile of the training non-conformity scores multiplied by this local scale.

If samples = NULL, the function returns point predictions and margins of error for the confidence levels supplied in conf. For example, conf = 0.95 uses the 0.95 quantile of the non-conformity scores.

If samples is a positive integer, the function returns simulated predictive samples. When smoothing = TRUE, signed non-conformity scores are generated using quantiles of the symmetrized empirical distribution. When smoothing = FALSE, non-conformity scores are sampled directly with replacement and assigned random signs.

Value

A matrix of predictions with 1 column per test point.

Examples

```
# Friedman function
f <-  function(x){
  10 * sin(pi * x[1] * x[2]) + 20 * (x[3] - 0.5)^2 + 10 * x[4] + 5 * x[5]
}
```

```
X <- matrix(runif(250), nrow=50, ncol=5)
y <- apply(X, 1, f)
fit <- rfok(X, y)

Xnew <- matrix(runif(250), nrow=50, ncol=5)
predict(fit, Xnew)
```

`print.rfok` *Print method for rfok objects*

Description

Print method for rfok objects

Usage

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

Arguments

`x`	Object returned by `rfok()`.
`...`	Additional arguments, currently ignored.

Value

Invisibly returns `x`.

`rfok` *Conformal Random Forest with Out-of-bag KNN*

Description

A conformal RF based on Johansson et al. (2014). Rather than using a traditional calibration set, conformal intervals are constructed using out-of-bag samples for each tree. K-nearest neighbors is used to calibrate the prediction intervals for each prediction location.

Usage

```
rfok(X, y, k = 5, beta = sd(y)/30, eps = 0, ...)
```

Arguments

X	a data frame or a matrix of predictors
y	a response vector
k	Number of nearest neighbors to use for out-of-bag error estimaters
beta	Sensitivity parameter
eps	Tolerance for nearest neighbor approach. When eps = 0, exact nearest neighbors are used.
...	additional parameters passed to randomForest

Details

This function fits a random forest and constructs conformal prediction intervals using out-of-bag (OOB) predictions. For each training observation, an OOB prediction is computed using only trees for which that observation was not included in the bootstrap sample. These OOB predictions are used to form residuals without requiring a separate calibration set.

Local scale estimates are computed using the k nearest neighbors in the predictor space. For observation i, the local scale estimate μ_i is the average absolute OOB residual among its nearest neighbors. The non-conformity score is

$$\alpha_i = \frac{|y_i - \hat{y}_i^{OOB}|}{\mu_i + \beta}.$$

The parameter k controls the locality of the scale estimate. Smaller values make the intervals more locally adaptive, while larger values produce smoother, more stable local scale estimates.

The parameter beta is a non-negative stabilization parameter added to the local scale estimate. Smaller values make the intervals more sensitive to local changes in the OOB residuals. Larger values reduce this sensitivity and make the intervals behave more like globally scaled conformal intervals. The default, sd(y) / 30, is a scale-aware value intended to provide mild stabilization.

The parameter eps is passed to the nearest-neighbor search. Setting eps = 0 uses exact nearest neighbors. Larger values allow approximate nearest-neighbor search and may be faster for large data sets.

Value

An object with class "rfok"

References

Johansson, U., Boström, H., Löfström, T., & Linusson, H. (2014). Regression conformal prediction with random forests. Machine learning, 97, 155-176.

Examples

```
# Friedman function
f <-  function(x){
  10 * sin(pi * x[1] * x[2]) + 20 * (x[3] - 0.5)^2 + 10 * x[4] + 5 * x[5]
}
```

```
X <- matrix(runif(250), nrow=50, ncol=5)
y <- apply(X, 1, f)
fit <- rfok(X, y)

Xnew <- matrix(runif(250), nrow=50, ncol=5)
predict(fit, Xnew)
```

`rfok_cv_beta` *Cross-validate the rfok sensitivity parameter*

Description

Cross-validates the sensitivity parameter `beta` for an `rfok` model over a user-supplied or automatically generated grid.

Usage

```
rfok_cv_beta(
  X,
  y,
  beta = NULL,
  beta_mult = c(0, 1/100, 1/50, 1/30, 1/20, 1/10),
  k = 5,
  num_folds = 5,
  selection_rule = c("crps", "interval_score", "coverage_width"),
  conf = 0.95,
  samples = 1000,
  smoothing = TRUE,
  eps = 0,
  return_fit = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

`X`	a data frame, matrix, or vector of predictors. A vector is coerced to a one-column matrix.
`y`	a numeric response vector.
`beta`	optional numeric vector of candidate beta values. If `NULL`, candidates are generated as `sd(y) * beta_mult`.
`beta_mult`	numeric vector used to generate the beta grid when `beta = NULL`.
`k`	Number of nearest neighbors to use for out-of-bag error estimators.
`num_folds`	Number of cross-validation folds.
`selection_rule`	Criterion used to choose beta. Options are `"crps"`, `"interval_score"`, and `"coverage_width"`.

conf	Confidence level used for interval-based selection rules.
samples	Number of conformal samples used when selection_rule = "crps".
smoothing	logical; should conformity scores be sampled or smoothed when generating conformal samples?
eps	Tolerance for nearest neighbor search.
return_fit	logical; if TRUE, refit rfok() on the full data using the selected beta and include the fitted model in the result.
verbose	logical. should progress be printed?
...	additional parameters passed to rfok().

Details

The parameter beta controls the sensitivity of the conformal interval widths to the local out-of-bag residual scale estimate. Smaller values make intervals more locally adaptive, while larger values make intervals more stable and more similar to globally scaled conformal intervals.

This function evaluates candidate beta values using cross-validation. For selection_rule = "crps", predictive samples are generated for each held-out observation and the candidate with the smallest average continuous ranked probability score is selected.

For selection_rule = "interval_score", conformal intervals are constructed at level conf and the candidate with the smallest average interval score is selected. The interval score rewards narrow intervals but penalizes observations that fall outside the interval.

For selection_rule = "coverage_width", the selected candidate is the one with the smallest average interval width among candidates with empirical coverage at least conf. If no candidate reaches the desired coverage, the candidate with the largest empirical coverage is selected, with ties broken by average interval width.

Value

A data frame with one row per candidate beta. The selected beta is marked in the selected column. If return_fit = TRUE, a list is returned with elements results, best_beta, and fit.

---

summary.rfok *Summary method for rfok objects*

Description

Summary method for rfok objects

Usage

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

Arguments

`object`	Object returned by `rfok()`.
`...`	Additional arguments, currently ignored.

Value

An object of class `"summary.rfok"`.

Index